Distributed Optimization Algorithms for Nonlinear Programming in Power Systems

Zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

von der KIT-Fakultät für
Elektrotechnik und Informationstechnik
des Karlsruher Instituts für Technologie (KIT)

genehmigte

DISSERTATION

von

M.Sc. Xinliang Dai
geb. in Fujian, China

Tag der mündlichen Prüfung:	28.02.2025
Hauptreferent:	Prof. Dr. Veit Hagenmeyer
Korreferent:	Prof. Dr.-Ing. Thomas Leibfried
	Prof. Dr. Moritz Diehl

Acknowledgment

Nearing the end of my nearly four-year Ph.D. journey, I am filled with gratitude as I reflect on the invaluable experiences and cherished memories I've gained. I sincerely thank everyone who directly or indirectly supported and helped me along the way. Without your encouragement and guidance, this dissertation would not have been possible.

First and foremost, I extend my deepest gratitude to my supervisor, Veit Hagenmeyer, for his guidance and unwavering support throughout this journey. Our numerous discussions, whether in the office or walking along the road, were invaluable to my growth. Beyond academia, his optimism, approach to life, and personal philosophy have profoundly influenced me. He is a true role model to me. I am also deeply grateful to Moritz Diehl for his warm hospitality during my visit to the University of Freiburg. The discussions, insights, and advice during that time were truly inspiring and significantly shaped my research. His warm-hearted nature made my stay in his group an enjoyable and enriching experience. Additionally, I sincerely thank Thomas Leibfried for serving as a reviewer of my dissertation and providing invaluable feedback. A special acknowledgment goes to Marc Hiller, Giovanni De Carne, and Sander Wahls, members of my defense committee, for their time, expertise, and encouragement during this critical phase.

Special thanks go to Yuning Jiang for supporting me through discussions and exploring optimization algorithms, and to Frederik Zahn, Arne Groß, and François Pacaud for their help in proofreading my thesis and providing valuable suggestions and feedback, which helped improve the quality of the dissertation. I am also grateful to Alexander Engelmann and Tillmann Mühlpfordt for their mentorship, when I joined the Institute for Automation and Applied Informatics (IAI) as a research assistant in the summer of 2019. Their solid work and dedication set a strong foundation for my Ph.D. journey.

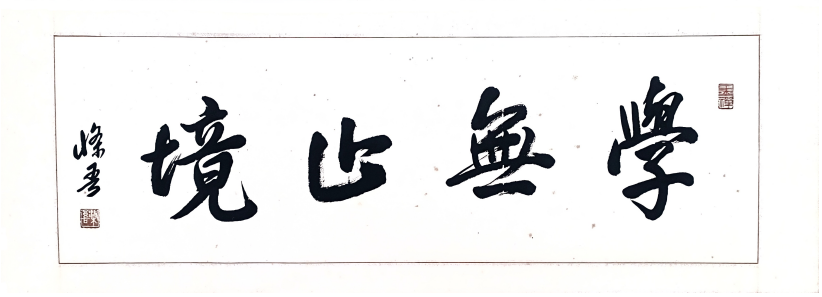
I extend my gratitude to my fellow group members at the time (in alphabetical order): Benedikt Heidrich, Dorina Werling, Janik Pinter, Rebecca Bauer, and Yanlin Jiang, as well as other colleagues at IAI: Alexander Kocher, Bernadette Lehmann, Burak Dindar, Hüseyin Kemal Cakmak, Jan Lukas Wachter, Jianlei Liu, Jovana Kovacevic, Katja Nau, Ömer Ekin, Philipp Schmurr, and Richard Jumar. My appreciation also goes to external collaborators and friends: Andrea Gehezzi, Armin Nurkanović, Colin Jones, David Kiessling, Fabian Ney, Gabriela Hug, Jakob Harzer, Jicheng Shi, Jonathan Frey, Junyi Zhai, Linbing Huang, Rachel Leuthold, Renzi Wang,

Rudolf Reiter, Shiying Dong, Wenliang Zhang, Xiao-Ping Zhang, Yi Guo, Ying Xue, Yingzhao Lian and Yunfan Gao.

I am deeply grateful to my dear parents for their support and encouragement in pursuing what I'm really interested in. My heartfelt thanks also go to my grandparents, my cousin, and other family members for their love and care, which have been a constant source of strength.

Last but not least, my heartfelt gratitude goes to my beloved fiancée, Yuhan, for her unconditional love and support throughout this journey. No matter we were close together or separated by thousands of kilometers, your presence has been my greatest motivation. I know that I could not have achieved this without you. As the saying goes,
所爱隔山海，山海亦可平
(If true love is separated by mountains and seas, then let mountains bow and seas yield—for I shall cross them all to be with you)

I would like to end my Ph.D. journey by sharing a piece of Chinese calligraphy my grandfather created and gave it to me as a prophetic saying when I was born:



(The pursuit of knowledge has no end)

Abstract

Modern power systems are extensive and increasingly complex, spanning vast geographical areas and integrating numerous distributed energy resources (DERs). Unlike traditional power systems with unidirectional flows and fixed loads, the rapid integration of DERs introduces variability in both demand and supply, alongside distributed generation. This shift complicates energy management and raises concerns about data privacy and communication infrastructure demands. Distributed optimization offers a viable solution by allowing local agents to solve smaller, decoupled problems with minimal data sharing, enhancing scalability, data privacy, resilience to single-point failures, and adaptability to network changes.

A major challenge in applying distributed optimization to power systems is the nonconvex nature of these systems. While existing distributed algorithms like alternating direction method of multipliers (ADMM) handle convex problems well, they struggle to achieve reliable convergence in nonconvex cases, such as AC power flow (PF) and optimal power flow (OPF) problems. Advancements are needed to improve convergence speed, scalability, and communication efficiency in these scenarios while maintaining convergence guarantees. This dissertation addresses these challenges by advancing distributed algorithms for generic nonlinear program (NLP), with a focus on AC PF/OPF problems.

The dissertation begins with enhancements to the augmented Lagrangian based alternating direction inexact Newton (ALADIN) algorithm, which offers theoretical local and global convergence guarantees for generic NLP. These improvements further enhance its computational efficiency and global convergence properties. Theoretical analysis confirms that these upgrades preserve convergence guarantees, and simulations demonstrate robust scalability in cases without inequality constraints. However, scalability remains challenging when inequality constraints are present. Further testing conducted within geographically distributed environments reveals communication as a critical bottleneck towards real-world application.

To solve these issues, the second part proposes a novel distributed optimization framework for large-scale NLP. This approach achieves rapid convergence and scalability by smoothing the problem into equality-constrained subproblems by employing the barrier method, solving these subproblems with an ALADIN-type algorithm, and using the Schur complement to reduce communication efforts. A graph-based decomposition strategy, paired with a single program

multiple data (SPMD) paradigm, enables efficient parallel processing. Extensive simulations on large-scale AC OPF benchmarks confirm that this framework outperforms the state-of-the-art centralized solver IPOPT on a modest workstation.

Further research addresses model simplification and managing variability introduced by DERs. We present a real-time coordination framework for integrated transmission-distribution (ITD) systems. For transmission systems, a distributed nonlinear model predictive control (NMPC) approach is proposed to coordinate the economic dispatch of multiple transmission system operators (TSOs), considering the aggregated DERs flexibility in distribution systems rather than considering the entire distribution system models. The framework's effectiveness and applicability in real-world scenarios are validated through simulated operational scenarios on a summer day in Germany, highlighting its robustness in the face of significant prediction mismatches due to severe weather conditions.

Kurzfassung

Moderne Energiesysteme sind zunehmend komplex und erstrecken sich über große geografische Gebiete mit zahlreichen dezentralen Energieressourcen (DERs). Anders als traditionelle Energiesysteme mit unidirektionalen Flüssen und festen Lasten führt die schnelle Integration von DERs zu einer erhöhten Variabilität in Nachfrage und Angebot sowie zur dezentralen Stromerzeugung. Diese Veränderung erschwert das Energiemanagement und stellt neue Anforderungen an den Datenschutz und die Kommunikationsinfrastruktur. Verteilte Optimierung bietet eine praktische Lösung, indem sie lokalen Agenten ermöglicht, kleinere, entkoppelte Probleme mit minimalem Datenaustausch zu lösen. Dies verbessert die Skalierbarkeit, den Datenschutz, die Widerstandsfähigkeit gegenüber Fehlern und die Anpassungsfähigkeit an Netzwerkänderungen.

Eine große Herausforderung bei der Anwendung verteilter Optimierung in Energiesystemen ist die nichtkonvexe Natur dieser Probleme. Während bestehende verteilte Algorithmen, wie die Alternating Direction Method of Multipliers (ADMM), konvexe Probleme gut lösen, haben sie Schwierigkeiten, in nichtkonvexen Fällen wie den AC-Lastfluss- (PF) und den Optimalen Lastfluss- (OPF) Problemen, zuverlässige Konvergenz zu erreichen. Es bedarf weiterer Fortschritte, um die Konvergenzgeschwindigkeit, Skalierbarkeit und Kommunikationseffizienz in diesen Szenarien zu verbessern und gleichzeitig Konvergenzgarantien aufrechtzuerhalten. Diese Dissertation widmet sich diesen Herausforderungen durch die Weiterentwicklung verteilter Algorithmen für allgemeine nichtlineare Programmierung (NLP) mit einem Fokus auf AC-PF/OPF-Probleme.

Diese Dissertation führt zunächst Verbesserungen am ALADIN-Algorithmus ein, welcher theoretische lokale und globale Konvergenzgarantien für allgemeine NLP bietet. Diese Verbesserungen steigern die rechnerische Effizienz und die globale Konvergenz des Algorithmus weiter. Theoretische Analysen bestätigen, dass diese Optimierungsrechnungen die Konvergenzgarantien beibehalten, und Simulationen zeigen eine robuste Skalierbarkeit in Fällen ohne Ungleichheitsbeschränkungen. Allerdings bleibt die Skalierbarkeit eine Herausforderung, wenn Ungleichheitsbeschränkungen vorliegen. Weiterführende Tests in geografisch verteilten Umgebungen identifizieren die Kommunikation als kritisches Hindernis für eine praktische Anwendung.

Um diese Herausforderungen zu bewältigen, wird im zweiten Teil ein neuartiges verteiltes Optimierungsframework für großskalige NLP vorgestellt. Dieser Ansatz erreicht eine schnelle

Konvergenz und Skalierbarkeit, indem das Problem mithilfe der Barrierenmethode in gleichheitsbeschränkte Teilprobleme geglättet wird. Diese Teilprobleme werden mit einem ALADIN-ähnlichen Algorithmus gelöst, und der Schur-Komplement-Ansatz wird zur Reduzierung des Kommunikationsaufwands eingesetzt. Eine graphenbasierte Dekompositionsstrategie, gepaart mit dem Single-Program-Multiple-Data (SPMD)-Paradigma, ermöglicht eine effiziente Parallelverarbeitung. Umfangreiche Simulationen auf großskaligen AC-OPF-Benchmarks zeigen, dass dieser Ansatz den zentralisierten Stand-der-Technik-Solver IPOPT auf einem einfachen Arbeitsplatzrechner übertrifft.

Weiterführende Forschungen befassen sich mit der Vereinfachung von Modellen und dem Management der durch DERs eingeführten Variabilität. Wir stellen ein Echtzeit-Koordinationsframework für integrierte Übertragungs- und Verteilersysteme (ITD) vor. Für Übertragungssysteme wird ein verteilter nichtlinearer modellprädiktiver Steuerungsansatz (NMPC) vorgeschlagen, um den wirtschaftlichen Einsatz mehrerer Übertragungsnetzbetreiber (TSOs) zu koordinieren, wobei die aggregierte Flexibilität der DERs in Verteilersystemen berücksichtigt wird, ohne die vollständigen Modelle der Verteilersysteme einzubeziehen. Die Effektivität und Anwendbarkeit des Frameworks werden durch simulierte Betriebsszenarien an einem Sommertag in Deutschland validiert, die seine Robustheit angesichts erheblicher Vorhersageabweichungen bei extremen Wetterbedingungen hervorheben.

Contents

Notation

$\mathbb{R}$	real numbers
$\mathbb{R}^n$	n-dimensional Euclidean space
$\mathbb{1}$	ones vector of given dimension
I_n	identity matrix
$\left\|\cdot\right\|_p$	p-norm
$\mathfrak{Re}(\cdot)$	real component of a complex quantity
$\mathfrak{Im}(\cdot)$	imaginary component of a complex quantity

Optimization

x, z	primal variables
λ, γ, κ	dual variables
ρ	penalty parameter
μ	barrier parameter
f	objective function
c	constraint function
$\mathcal{L}$	Lagrangian function
$\mathcal{A}$	active set
g	gradient
J	Jacobian
H	Hessian of Lagrangian

Distributed Formulation

$\mathcal{R}$	set of subsystems
N^{reg}	number of subsystems
ℓ	region $\ell \in \mathcal{R}$
x_ℓ	subvector of x with respect to the region ℓ
A_ℓ	submatrix of A with respect to the region ℓ
blkdiag(A_ℓ)	block diagonal matrix with A_ℓ for all $\ell \in \mathcal{R}$ along its diagonal
vertcat(A_ℓ)	vertically concatenating A_ℓ for all $\ell \in \mathcal{R}$
horzcat(A_ℓ)	horizontally concatenating A_ℓ for all $\ell \in \mathcal{R}$

Power Systems

$\mathcal{N}$	set of buses
$\mathcal{L}$	set of (from-side) branches
$\mathcal{L}^R$	set of (to-side) branches
$\mathcal{G}$	set of generators
s, p, q	apparent/active/reactive power
θ, v	voltage angle/magnitude
u, w	real/imaginary parts of complex voltage
$(\cdot)_i$	quantities related bus i
$(\cdot)_{ij}$	quantities related to the branch from bus i to bus j
$(\cdot)^g$	generation
$(\cdot)^d$	demand
$(\cdot)^s$	storage

Acronyms

AD	automatic differentiation
SPMD	single program multiple data
VPN	virtual private network

Optimization

LP	linear programming
MILP	mixed-integer linear programming
NLP	nonlinear program
LICQ	linear independence constraint qualification
SOSC	second order sufficient condition
SCC	strict complementarity condition
KKT	Karush–Kuhn–Tucker
QP	quadratic program
SQP	sequential quadratic programming
IPM	interior point method
ALM	augmented Lagrangian method
MPC	model predictive control
NMPC	nonlinear model predictive control
ADMM	alternating direction method of multipliers
ALADIN	augmented Lagrangian based alternating direction inexact Newton
BALADIN	Barrier ALADIN

Power Systems

AC	alternating current
PF	power flow
OPF	optimal power flow
MPOPF	multiperiod AC optimal power flow
BIM	bus injection model
BFM	branch flow model
ITD	integrated transmission-distribution
TSO	transmission system operator
DSO	distribution system operator
PCC	point of common coupling
DERs	distributed energy resources
SOC	state-of-charge
std	standard
sad	small angle difference
api	active power increased
p.u.	per-unit system

1 Introduction

1.1 Motivation

Modern power systems are extensive and geographically distributed, often spanning entire countries or continents, and are closely connected with local utilities. Optimizing these systems is challenging due to the high number of variables, complex constraints, and nonlinear physical properties involved.

Traditionally, electricity flows unidirectionally—from large, centralized fossil-fuel power plants through transmission systems down to end-users in distribution systems. Although transmission and distribution systems are physically connected, they are typically operated separately: a transmission system operator (TSO) manages the dispatch of power centrally, either directly or through a centralized marketplace [1, 2], while a distribution system operator (DSO) focuses on operating the distribution network, making investments, and performing maintenance [3].

In recent decades, the power system has been transitioning from relying on large fossil-fuel plants with inflexible loads to incorporating numerous distributed energy resources (DERs) such as rooftop solar panels, battery storage, electric vehicles with vehicle-to-grid capabilities, and controllable loads for demand response. Their decentralized nature introduces bidirectional power flows, and their weather-dependent characteristics cause rapid fluctuations in both supply and demand. This significantly complicates energy management and underscores the need for efficient coordination and advanced management solutions. Additionally, the increase in controllable devices adds model complexity, raising privacy concerns and intensifies communication infrastructure demands.

In the context of centralized coordination between transmission and distribution systems, DSOs generally do not validate or dispatch DERs, which risks inefficiencies in utilizing DERs services. Without direct or indirect control over DSOs assets, TSOs may struggle to effectively orchestrate DERs contributions, reducing their potential benefits. Fully centralized management becomes increasingly impractical as DER penetration rises due to substantial computational and modeling challenges and the significant volumes of real-time data required to transfer between

DSOs and TSOs. Additionally, TSOs, lacking specific expertise in distribution network requirements, may face difficulties in accurately interpreting these needs, further complicating system coordination [1].

In response to these challenges, distributed optimization provides an efficient alternative for coordinating geographically distributed systems. It enables independent operation and effective collaboration through limited information sharing. In distributed optimization, agents in multi-agent systems work together to minimize a global function composed of individual local objective functions based on its own data and the limited information exchanged via a communication network, allowing the overall optimization problem to be solved in a distributed manner.

Potential advantages of distributed optimization in power systems include [4, 5, 6]:

- **Scalability**: Distributed algorithms decompose large, complex problems into smaller subproblems. This enables fast computations and makes the approach scalable.
- **Privacy and/or Sovereignty Preservation**: With distributed optimization, data privacy is better preserved because detailed information, such as grid configurations or customer usage behaviors, does not need to be shared. Each entity can maintain the confidentiality of its own data, which is crucial in collaborations where different entities own and operate separate parts of the system.
- **Robustness**: Distributed approaches enhance system reliability and resiliency by mitigating the risk of single-point cyber failures that centralized systems are prone to.
- **Adaptability**: These algorithms adapt more quickly to network topology and infrastructure changes without requiring a complete system overview. This enables flexible reconfiguration of the system when new components are added, or existing ones are modified.

While these advantages are promising, implementing distributed optimization in power systems is challenging, and several issues must be solved before these benefits can be fully realized.

1.2 Challenges

Distributed optimization has been studied for many years, tracing back to foundational work in parallel and distributed computation [7, 8, 9]. In the past decade, interest in this area has grown due to its applications in power systems, communication networks, machine learning, and sensor networks. Numerous distributed algorithms have been developed, and recent progress is documented in several surveys [10, 4, 11, 12].

In power system operations, key application areas for distributed optimization are steady-state problems, including distributed power flow (PF) [13, 14], distributed optimal power flow (OPF) [15, 16, 17, 18], distributed economic dispatch [19, 20], distributed coordinated restoration [21, 22], etc. For more comprehensive overviews of distributed optimization in power system applications, we refer to [4, 5, 23].

While distributed algorithms often perform well for convex problems, their convergence becomes more challenging in nonconvex cases like AC PF/OPF problems. As one of the most practically important optimization problems in electric power systems engineering, the AC PF/OPF problems are generally NP-hard, even for radial power grids [24, 25]. The nonconvexity lies in the nonlinear AC PF equations. State-of-the-art distributed optimization algorithms often lack guaranteed convergence and scalability for solving AC OPF problems, such as Optimality Condition Decomposition (OCD) [26], Auxiliary Problem Principle (APP) [27], Diagonal Quadratic Approximation (DQA) [17], and alternating direction method of multipliers (ADMM) [28, 29, 30, 31]. Exceptions include a two-level variant of ADMM [32] and a heterogeneous decomposition algorithm [16]. Nonetheless, both are first-order algorithms and exhibit slow numerical convergence with modest accuracy.

In contrast to the aforementioned distributed approaches, the augmented Lagrangian based alternating direction inexact Newton (ALADIN) method is designed for generic nonconvex distributed problems [33]. By adapting the sequential quadratic programming (SQP) methodology, ALADIN can be viewed as a second-order upgrade of the ADMM algorithm and provides several advantages over the aforementioned distributed approaches. It provides a local convergence guarantee with a locally quadratic convergence rate for generic distributed nonconvex optimization problems when suitable Hessian approximations are employed. Additionally, ALADIN guarantees global convergence by implementing a globalization strategy [33, Alg. 3]. Recently, ALADIN has been successfully applied to solve AC OPF of transmission systems [34] and of hybrid AC/DC systems [35], demonstrating fast convergence and potential for handling generic distributed problems with heterogeneous models. However, these studies have been limited by grid sizes (less than 300 buses) and the number of subgrids (less than half a dozen).

Critical Questions of distributed optimization have to be solved before its true potential can be fully realized. Challenges for implementing distributed optimization include:

- **Convergence Speed**: Algorithms must be fast enough to handle rapid changes in grid conditions, but many current methods require thousands of iterations for acceptable accuracy, indicating the need for improved convergence rates.
- **Convergence Guarantees**: There is a need for theoretical frameworks that ensure convergence within reasonable time frames, providing mathematical rigor to the solutions.

- **Scalability**: It is essential to demonstrate scalability with increasingly large-scale power system benchmarks under diverse but realistic operational conditions.
- **Communication Efficiency**: Algorithms should require simple and limited communication that can be implemented using existing technologies and infrastructures.

1.3 Outline and Contributions

With these theoretical and practical considerations in mind, this dissertation investigates and advances distributed optimization algorithms for application in power systems, particularly focusing on AC PF/OPF problems. It combines theoretical insights with practical implementation to address critical challenges in scalability and efficiency.

This dissertation begins by establishing the foundations of nonlinear programming (NLP) in Chapter 2 and provides a detailed overview of the standard ALADIN algorithm in Chapter 3. This chapter highlights the potential limitations of the standard ALADIN, particularly regarding its scalability, computational efficiency, and communication overhead.

The primary contributions of this dissertation are threefold. First, Chapter 4 enhances the standard ALADIN framework to improve both local and global performance for distributed optimization problems in power systems. This chapter also explores the impact of communication overhead when ALADIN is implemented in a geographically distributed computing environment, indicating that the communication overhead is the key limiting factor in efficiently solving NLPs in such distributed settings.

Second, Chapter 5 introduces a two-level distributed optimization algorithm designed for large-scale NLPs. By integrating smoothing techniques with the Schur complement, the proposed approach addresses the scalability challenges and reduces communication overhead, a key limitation of the standard ALADIN as discussed in previous chapters. Under the single program multiple data (SPMD) paradigm, the computational tasks are distributed and processed in parallel with distributed memory. Extensive simulations on large-scale benchmarks across various operating scenarios account for synchronization overhead and show that the proposed approach outperforms the state-of-the-art centralized nonlinear solver IPOPT on modest hardware.

Finally, Chapter 6 shifts the focus from algorithmic improvements to the development of a scalable coordination framework for integrated transmission and distribution (ITD) systems. This framework integrates distributed optimization with system model simplifications for low-level agents and employs distributed nonlinear model predictive control (NMPC) to enable real-time distributed coordination. The approach aggregates the flexibility of DERs in distribution systems

and is validated through simulations conducted over a summer day in Germany with significant prediction mismatches due to severe weather conditons.

The following provides detailed outlines and the individual contributions of the chapters.

Chapter 2 - Foundations of Nonlinear Programming

This chapter introduces the basic concepts and algorithms of nonlinear program (NLP) that permeate the entire dissertation. It begins with the terminology and notation used throughout the dissertation, followed by an introduction to widely-used centralized optimization methods for solving NLPs, including sequential quadratic programming (SQP) and interior point methods (IPM). Techniques for solving quadratic subproblems are also discussed, along with techniques to solve the quadratic subproblems. Additionally, the chapter covers foundational approaches for distributed optimization, such as dual decomposition, augmented Lagrangian methods (ALM), and alternating direction method of multipliers (ADMM).

Chapter 3 - ALADIN: A Distributed SQP Approach

This dissertation focuses on solving nonconvex NLP problems in a distributed manner. The augmented Lagrangian based alternating direction inexact Newton (ALADIN) is introduced as an effective alternative to the widely-used ADMM. This chapter begins with a comparative analysis of ADMM and ALADIN, followed by a detailed overview of the standard ALADIN algorithm, including its framework, convergence properties, as well as key limitations related to scalability and efficiency. This analysis provides the foundation for the algorithmic advancements and applications to power systems optimization presented in later chapters.

The specific contributions are as follows:

- In Section 3.1, we provide a comparison between ADMM and ALADIN. This analysis, based on simulation and discussion in [6], provides motivation to adopt ALADIN over ADMM for NLPs in power systems optimization.
- In Section 3.4, we summarize the main challenges applying ALADIN to generic NLPs, including combinatorial difficulty posed by active-set methods as explored in [36], inefficient globalization strategy as highlighted in [37], and communication overhead and potential security concerns discussed in [38].

Chapter 4 - Advancing ALADIN in Power System Applications

This chapter adapts the ALADIN algorithm for power system applications, focusing on AC power flow (PF) and AC optimal power flow (OPF) problems. With the ALADIN framework, this chapter improves the algorithm performance for solving AC PF problems by using the Gauss-Newton methods based on [39] and introduces an improved globalization strategy to balance global reliability and efficient computation based on [37].

Previously, ALADIN research has focused on single-machine implementations where subproblems are solved sequentially, leaving the impact of communication overhead in distributed environments unexplored. This chapter addresses this research gap by implementing the standard ALADIN algorithm to solve AC OPF in a geographically distributed computing environment as demonstrated in [40]. Results indicate that communication overhead is the primary limiting factor for efficiently solving NLP problems in such settings, particularly for second-order algorithms like ALADIN.

The contributions of this chapter include the following:

- Section 4.2 introduces the distributed reformulation of power system optimization problems by sharing components between neighboring regions, as proposed in [6]. Different from cutting tie lines and introducing auxiliary variables in [34], this approach avoids additional modeling assumptions.
- Section 4.3.1 reformulates the AC power flow problem as a zero-residual least-squares problem and applies the Gauss-Newton method to approximate the Hessian within the ALADIN framework, as detailed in Algorithm 6 in Section 4.3.2. Theorem 4.1 proves that this combination ensures a local quadratic convergence rate.
- Simulation results in Section 4.3.4 highlight the scalability and computational efficiency of the proposed Algorithm 6. Despite executing all decoupled subproblems sequentially, the algorithm achieves computing times and a solution quality comparable to the centralized approach by the Matpower toolbox.
- Section 4.4.1 introduces a global ALADIN variant in Algorithm 7, incorporating a new globalization strategy to improve the computational efficiency. Theorem 4.3 in Section 4.4.2 provides a detailed global convergence analysis.
- Benchmark results in Section 4.4.3 demonstrate how Algorithm 7 maintains robustness starting from an arbitrary initial guess, in contrast to the traditional full-step ALADIN, which often diverges when the initial dual guess is far from optimal, as discussed in Section 3.4.

- Section 4.5 proposes a novel distributed approach for solving AC OPF problems using the standard ALADIN algorithm within a geographically distributed computing environment. The method enables iterative communication between local clients and a coordinator, protecting confidential grid information and preserving customer privacy. Simulations reveal that the communication overhead is the key limiting factor in efficiently solving NLPs in such distributed settings.

Chapter 5 - A Scalable Distributed IPM Approach

As discussed in the previous chapter, the primary challenges of applying the ALADIN algorithm to real-world applications are the scalability and the communication requirements. Scalability is hindered by the combinatorial complexity that arises when handling inequality constraints, while communication issues stem from the need to transfer the full approximated model to a central coordinator. This not only results in significant communication overhead but also raises potential security concerns due to the exposure of sensitive data from local agents.

This chapter integrates ALADIN with the interior point method (IPM) to develop a scalable distributed optimization algorithm for generic NLP. Addressing the challenges above, the approach achieves improved scalability and communication efficiency with convergence guarantees. Case studies conducted on large-scale benchmarks under various operational scenarios demonstrate the enhanced performance of the proposed algorithms, surpassing the state-of-the-art centralized solvers IPOPT on modest hardware.

This chapter is based on [38], and the contributions are as follows:

- In Section 5.2.1, we introduce a two-level distributed optimization algorithm (Algorithm 10) for generic NLP. The upper level employs barrier methods to handle inequality constraints, while the lower level uses ALADIN to solve the resulting smoothed equality-constrained problems. This improves scalability by avoiding combinatorial challenges identified as a limitation of the standard ALADIN in Section 3.4.
- To further mitigate communication issues of the standard ALADIN, as discussed in Section 3.4, a condensing step is incorporated into Algorithm 10 in Section 5.2.1 to condense the derivatives in the dual space. This step reduces overall communication overhead while also mitigating the risk of exposing network topology information contained in the sparsity pattern of derivatives. Moreover, an analysis of the communication effort is provided later in Section 5.6, considering both theoretical worst-case and practical scenarios.

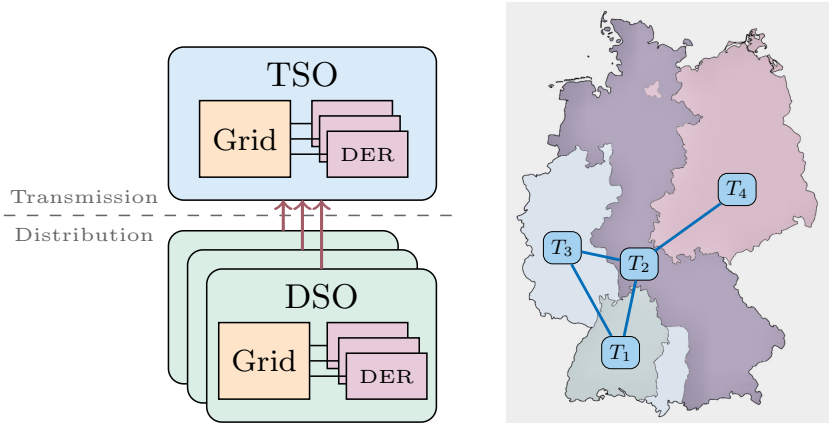
- Section 5.2.2 proposes a distributed inertia correction method to control inertia in the dual space as part of the coordination step within Algorithm 10. This addresses negative curvature along feasible directions, improving both scalability and robustness.
- In Section 5.3, the global convergence of the proposed Algorithm 10 is analyzed in Theorem 5.2. It integrates the convergence results from the standard ALADIN (Theorem 3.2) within convergence properties of IPM (Remark 2.5).
- Section 5.4 covers practical considerations for implementing the Algorithm 10, including criteria for optimal network decomposition, distributed automatic differentiation (AD), and strategies to ensure resilience against single-point failures. A trade-off in achieving optimal network decomposition is identified and further analyzed later in Section 5.5.2.
- In Section 5.5, we demonstrate the scalability of this algorithm through extensive large-scale simulations with different operation scenarios. Under the single program multiple data (SPMD) paradigm, the computational tasks are distributed and processed in parallel with distributed memory. These tests account for synchronization overhead and show that the proposed approach outperforms the state-of-the-art centralized nonlinear solver IPOPT on modest hardware.

Chapter 6 - Distributed Coordination Framework for Integrated Transmission and Distribution Systems

Unlike previous chapters, which focused solely on iterative distributed optimization algorithms, this chapter presents a scalable coordination framework by integrating distributed optimization with system model simplifications for low-level agents. The aggregated flexibility of controllable devices in distribution networks is encapsulated, re-calculated, and communicated through the power-energy envelopes, facilitating a reduction in computational complexity and eliminating redundant information exchanges between TSOs and DSOs, thereby enhancing privacy and security. Unlike previous chapters, which use methods such as the active-set method by ALADIN as in Section 4.5 or the smoothing technique by BALADIN to solve coupled equality-constrained QP subproblems as in Chapter 5, the proposed algorithm solves coupled inequality-constrained QP subproblems, enhancing scalability at the cost of computation effort, as discussed in Section 2.3.1.

The scheme of the proposed operational architecture is shown in Fig. 1.1, as inspired by the actual situation in Germany. The simulations rely on real-world data sourced from the ENTSO-E Transparency Platform* [41]. The framework's effectiveness and real-world applicability are

*The data used in this chapter is publicly available at the ENTSO-E Transparency Platform: https://transparency.entsoe.eu



(a) Aggregated flexibility of DSOs

(b) TSOs Coordination

Figure 1.1: Proposed real-time coordination of integrated transmission-distribution

validated through operational simulations conducted for a summer day in Germany, accounting for substantial prediction mismatches caused by varying weather conditions. These results highlight the framework's practical utility in real-time power system operations.

This chapter is based on [42], and the contributions are as follow:

- Section 6.2 proposes a flexibility aggregation method to compute the feasible dispatch region of all controllable devices leveraging the LinDistFlow model [43] and employs an inner hyperbox approximation.
- In Section 6.2.1, we demonstrate that all state variables in the LinDistFlow model, including squared voltage magnitudes, can be explicitly expressed in terms of controllable power injections in radial networks. Compared with prior analyses in [44], Proposition 6.1 does not rely on the positive definiteness of resistance and reactance for all branches, a condition not universally met in practical power system datasets as discussed in [45].
- In Section 6.4.1, we propose a spatial & temporal decomposition approach to decouple the multiperiod AC OPF problems across system operators and time periods. Each resulting subproblem corresponds to s a standard AC OPF problem of a specific TSO.
- In Section 6.4.2, we propose a novel distributed real-time framework for coordination of the economic dispatch problem of ITD systems. This framework provides optimal dispatch solutions by using distributed NMPC to solve the multiperiod AC OPF with spatial & temporal decomposition at the transmission level while accounting for the aggregated flexibility of distribution systems.

- Finally, Section 6.5 provides a comprehensive simulation using real-world measurement data, including load profiles and solar and wind generation outputs from a summer day in Germany with significant forecast errors caused by heavy rainfall. This simulation, involving over 100,000 state variables divided into 400 subproblems at the transmission level, underscores the proposed approach's efficiency, scalability, and practical applicability for TSO-DSO coordination.

Chapter 7 - Conclusion

This section concludes the dissertation by summarizing its key contributions. It discusses the main advantages and limitations of the developed methods and theory, highlighting their impact on power system optimization. Additionally, several ideas for improving the current methods and suggestions for future research directions are proposed.

2 Foundations of Nonlinear Programming

The primary focus of this dissertation is to develop efficient distributed approaches for solving nonlinear optimization problems in electric power systems. While centralized approaches have been extensively studied and widely applied, distributed optimization methods, despite recent progress, remain impractical for many real-world power system applications. Key challenges include the absence of robust convergence guarantees, slow convergence requiring numerous iterative data exchanges, and difficulties in achieving high solution accuracy.

This chapter reviews some basic concepts and algorithms in nonlinear programming to establish the terminology and notation used throughout this dissertation, as well as laying the foundations for addressing the challenges of distributed optimization in later chapters. The mathematical formulation largely follows the conventions outlined by [46, 47] for centralized optimization. For distributed optimization, the principles align more closely with those presented by [10], though these two sources use different but mathematically equivalent notation.

Outline. Section 2.1 introduces the essential notation, terminology, and methodologies employed in this work. Section 2.2 outlines the optimality conditions central to nonlinear optimization. Section 2.3 discusses foundational centralized optimization methods, including sequential quadratic programming (SQP), interior-point methods (IPM), and techniques for solving quadratic programming (QP) problems. Section 2.4 introduces dual decomposition and augmented Lagrangian methods (ALM) as foundational concepts leading to the alternating direction method of multipliers (ADMM), a powerful algorithm well-suited for distributed convex optimization. Section 2.5 concludes the chapter and summarizes the challenges of distributed optimization.

2.1 Problem Statement

Depending on the type of engineering application, numerical optimization problems can be formulated in various forms. Most of the quantities, such as voltage, current, and power in

power system operation, are *continuous*, although some cases involve *discrete* decisions, such as switching a specific device on or off or choosing between control strategies. This dissertation focuses on optimization problems with continuous variables.

The relationships between quantities in engineering applications are represented by equations, which are often nonlinear. These quantities often need to be kept within certain intervals to ensure physical meaningfulness, such as positive semidefinite current magnitude, or to meet operational constraints, such as the thermal limit of transmission lines.

Therefore, the resulting optimization problems are often *nonconvex* and *nonlinear*, and are referred as nonlinear program (NLP). This dissertation only considers *smooth* optimization problems. These problems involve minimizing or maximizing a nonlinear objective function while keeping the decision variables within a feasible set defined by nonlinear constraints, which could be either equality or inequality constraints.

A generic formulation of an NLP can be written as

$$\min_{x \in \mathbb{R}^{n_x}} \quad f(x) \tag{2.1a}$$
$$\text{s.t.} \quad c^{\mathrm{E}}(x) = 0 \tag{2.1b}$$
$$c^{\mathrm{I}}(x) \leq 0 \tag{2.1c}$$

where the objective function $f : \mathbb{R}^{N^x} \to \mathbb{R}$, the equality constraint function $c^{\mathrm{E}} : \mathbb{R}^{N^x} \to \mathbb{R}^{N^{\mathrm{E}}}$ with $N^x \geq N^{\mathrm{E}}$, and the inequality constraint function $c^{\mathrm{I}} : \mathbb{R}^{N^x} \to \mathbb{R}^{N^{\mathrm{I}}}$ are assumed to be sufficiently smooth and continuously differentiable on the *feasible set* defined as

$$\mathcal{X} = \{x \in \mathbb{R}^{N^x} \mid c^{\mathrm{E}}(x) = 0,\ c^{\mathrm{I}}(x) \leq 0\},$$

and Problem (2.1) is said to be *feasible* if $\mathcal{X}$ is non-empty. Set $\mathcal{C}$ denotes the set of indices for all the constraints

$$c(x) = \begin{bmatrix} c^{\mathrm{E}}(x) \\ c^{\mathrm{I}}(x) \end{bmatrix}.$$

Following this, we introduce an important item of terminology that recurs throughout this dissertation:

Definition 2.1 (Active Set). *The active set $\mathcal{A}(x)$ at any feasible $x \in \mathcal{X}$ is the set of the indices of the equality and inequality constraints for which $c_i(x) = 0$; that is*

$$\mathcal{A}(x) = \{i \in \mathcal{C} \mid c_i(x) = 0\}$$

denotes the set of active constraints at x.

With the definition of a neighborhood

$$\mathcal{B}_\varepsilon(\bar{x}) = \{x \mid \|x - \bar{x}\| \leq \varepsilon\},$$

minimizers to a *feasible* Problem (2.1) are defined as follows:

Definition 2.2 (Global & Local Minimizers [47]). *A point* $x^* \in \mathcal{X}$ *is said to be a local minimizer of* f *on* $\mathcal{X}$ *if*

$$\exists \varepsilon > 0 \text{ such that } f(x) \geq f(x^*),\ \forall x \in \mathcal{B}_\varepsilon(x^*) \cap \mathcal{X},$$

and is said to be a global minimizer of f *on* $\mathcal{X}$ *if and only if*

$$f(x) \geq f(x^*),\ \forall x \in \mathcal{X}.$$

Remark 2.1. *A minimizer* x^* *is considered a strict minimizer if the above inequalities hold strictly for all* $x \in \mathcal{X} \backslash \{x^*\}$.

2.2 Optimality Conditions

This section states optimality conditions required for x^* to be a local minimizer.

As a preliminary step to the optimality conditions, we define the *Lagrangian function* for Problem (2.1) as

$$\mathcal{L}(x,\ \gamma,\ \kappa) = f(x) + \gamma^\top c^{\mathrm{E}}(x) + \kappa^\top c^{\mathrm{I}}(x). \tag{2.2}$$

with *Lagrangian multipliers* $\gamma \in \mathbb{R}^{N^{\mathrm{E}}}$ and $\kappa \in \mathbb{R}^{N^{\mathrm{I}}}$ corresponding to the equality constraints (2.1b) and the inequality constraints (2.1c), respectively. The corresponding *dual problem* is given by

$$\max_{\gamma,\kappa} d(x, \gamma, \kappa),\ \text{s.t. } \kappa \geq 0,$$

where the Lagrangian dual function d is defined as infimum of the Lagrangian over x:

$$d(x, \gamma, \kappa) = \inf_x \mathcal{L}(x,\ \gamma,\ \kappa).$$

Remark 2.2 ([47]). *Suppose an inequality constraint $c_j^I(x) \leq 0$ in (2.1c) is replaced by $c_j^I(x) \leq \epsilon$, and $x_\epsilon^\star$ is a solution to this new NLP. If c_j^I is active at $x_\epsilon^\star$ and ϵ is sufficiently small, then*

$$f(x_\epsilon^\star) = f(x_0^\star) - \kappa_j \, \epsilon + \mathcal{O}(\epsilon^2)$$

Namely, the Lagrange multiplier can be interpreted as the sensitivity differential $\kappa_j = -\frac{\partial f}{\partial \epsilon}(x_0^\star)$, indicating how the objective value change with respect to a slight deviation ϵ of the corresponding inequality constraints. This interpretation can also be easily adapted for an equality constraint c^E and its corresponding Lagrange multiplier γ_j.

We also introduce a strong constraint qualification*:

Definition 2.3 (Linear Independence Constraint Qualification). *Given a feasible point x to Problem (2.1) and its active set $\mathcal{A}(x)$ defined in definition 2.1. We say that the linear independence constraint qualification (LICQ) holds at x if $\nabla c_i(x)$ are linearly independent for all $i \in \mathcal{A}(x)$.*

With this constraint qualification, we are able to introduce the *first-order necessary optimality conditions*.

Theorem 2.1 (First-order Necessary Optimality Conditions†). *Let x^* be a local minimizer of Problem (2.1), and assume that f, c^E and c^I are continuously differentiable functions and LICQ hold at x^*. Then, there exists unique Lagrangian multiplier vectors γ^* and κ^* with $\kappa^* \geq 0$ such that the following conditions are satisfied at $(x^*, \gamma^*, \kappa^*)$*

$$\nabla_x \mathcal{L}(x^*, \gamma^*, \kappa^*) = 0, \tag{2.3a}$$
$$c^E(x^*) = 0, \tag{2.3b}$$
$$c^I(x^*) \leq 0, \tag{2.3c}$$
$$\kappa^* \geq 0 \tag{2.3d}$$
$$\kappa_i^* \, c_i^I(x^*) = 0, \quad \forall i \in \mathcal{C}^I, \tag{2.3e}$$

The condition 2.3 is also known as the *Karush–Kuhn–Tucker (KKT) conditions*, and the point $(x^*, \gamma^*, \kappa^*)$ is called a *KKT point*. Condition (2.3e) is known as *complementarity slackness*.

*Mangasarian-Fromovitz constraint qualification (MFCQ) [48] represents a less restrictive condition than the LICQ, with even weaker constraint qualifications existing in optimization theory. This dissertation employs only LICQ because the Lagrangian multipliers can be guaranteed to be unique if LICQ holds [47], as stated in Theorem 2.1. For detailed description of MFCQ, refer to [46, Sec 12.6]

†The necessity of LICQ is well-illustrated in Example 1.43 in [47], where the KKT conditions do not hold at the minimizer due to the lack of constraint qualification.

Definition 2.4 (Strict Complementarity Condition). *The strict complementarity condition (SCC) hold if a KKT point* $(x^*, \gamma^*, \kappa^*)$ *of Problem* (2.1) *satisfy that* $\kappa_i^* > 0$ *for each* $i \in \mathcal{C}^I \cap \mathcal{A}(x^*)$.

Consider moving away from x^* along the *first-order feasible direction*

$$\mathcal{D}(x^*) = \left\{ w \in \mathcal{R}^{N^x} \mid \nabla c_i(x^*) w = 0, \quad \forall i \in \mathcal{A}(x^*) \right\}$$

within a small neighborhood $\mathcal{B}_\varepsilon(x^*)$. Using only first-order information alone with the strict complementarity condition (SCC)[‡], defined below, is insufficient to predict whether such movement will increase or decrease the objective function f.

The second order sufficient condition (SOSC) condition, presented below, examines the curvature of the Lagrangian function $\mathcal{L}$ along the first-order feasible direction $\mathcal{D}(x^*)$.

Since second derivatives are involved, stronger smoothness assumptions are necessary compared to the KKT conditions (Theorem 2.1). In the following, f and c are assumed to be twice continuously differentiable.

Theorem 2.2 (Second-Order Sufficient Condition). *Let* $(x^*, \gamma^*, \kappa^*)$ *be a KKT point of Problem* (2.1) *where LICQ and SCC hold. If*

$$w^\top \nabla^2_{xx} \mathcal{L}(x^*, \gamma^*, \kappa^*) w > 0, \quad \forall w \in \mathcal{D}(x^*),\ w \neq 0 \tag{2.4}$$

holds, then, x^* *is a strict local minimizer of Problem* (2.1).

Here, we introduce a common regularity assumption on KKT points used throughout this dissertation.

Definition 2.5 (Regular KKT Point). *A KKT point for the problem is considered regular if the linear independence constraint qualification (LICQ), the strict complementarity condition (SCC), and the second order sufficient condition (SOSC) are satisfied.*

[‡]If the SCC is not satisfied, the *first-order feasible direction* should be replaced by the *critical cone*. We refer to [49, Sec. 12.5] for more details

2.3 Centralized Newton-Type Methods for Nonlinear Optimization

Newton-type methods are root-finding algorithms that linearize nonlinear functions to converge towardssolutions iteratively. For a constrained nonlinear program (NLP) (2.1), which is a nonlinear and possibly nonconvex optimization problem, two widely used methods are sequential quadratic programming (SQP) and interior point method (IPM) [50]. SQP operates by solving a sequence of QP subproblems, where each subproblem minimizes a quadratic approximation of the objective function while satisfying the linearized constraints. On the other hand, IPM approximates a continuous path leading to the solution of Problem (2.1). In its simplest form, this path is defined by a positive scalar parameter μ, which can be seen as a perturbation of the optimality conditions for Problem (2.1).

In this section, we introduce the SQP framework for optimization problems with equality constraints and discuss the challenges encountered when extending it to inequality constraints in Section 2.3.1. The IPM is presented in Section 2.3.2. Finally, QP problems and relevant techniques are discussed in Section 2.3.3.

2.3.1 Sequential Quadratic Programming

SQP methods were first introduced in 1963 [51], and have garnered significant research interest in the decades since. Despite the numerous variations of SQP-type algorithms available today [52, 53, 54, 55, 56, 57], their fundamental concept remains similar. For simplicity, consider an NLP without inequality constraints

$$\min_{x \in \mathbb{R}^{n_x}} \quad f(x) \tag{2.5a}$$

$$\text{s.t.} \quad c(x) = 0. \tag{2.5b}$$

The KKT conditions defining the optimal solution in this scenario are

$$\nabla_x \mathcal{L}(x^*,\, \gamma^*) = 0 \tag{2.6a}$$

$$c(x^*) = 0 \tag{2.6b}$$

with the Lagrangian function

$$\mathcal{L}(x,\, \gamma) = f(x) + \gamma^\top c(x).$$

We define

$$\mathcal{F}(x,\gamma) = \begin{pmatrix} \nabla_x \mathcal{L}(x,\gamma) \\ c(x) \end{pmatrix} = \begin{pmatrix} \nabla f(x) + \nabla c(x)^\top \gamma \\ c(x) \end{pmatrix}$$

Then, we can summarize the KKT conditions (2.6) as

$$\mathcal{F}(x^*,\gamma^*) = 0. \tag{2.7}$$

To approach the KKT point (x^*,γ^*) for Problem (2.5), where KKT conditions hold, the SQP algorithms apply a Newton method to solve the nonlinear system (2.6) iteratively. With an initial guess on the primal-dual point $(x_0,\ \gamma_0)$, SQP algorithms generate a primal-dual sequence by

$$\begin{bmatrix} x^{k+1} \\ \gamma^{k+1} \end{bmatrix} = \begin{bmatrix} x^k \\ \gamma^k \end{bmatrix} + \begin{bmatrix} \Delta x^k \\ \Delta \gamma^k \end{bmatrix} \tag{2.8}$$

where the step $(\Delta x^k, \Delta\gamma^k)$ can be obtained with the assistance of Taylor expansion

$$\mathcal{F}(x^k,\gamma^k) + \nabla \mathcal{F}(x^k,\gamma^k)^\top \begin{pmatrix} \Delta x^k \\ \Delta \gamma^k \end{pmatrix} = 0 \tag{2.9}$$

For simplification, we ignore k and define $c := c(x^k)$, $J := \nabla c(x^k)^\top$, $g := \nabla f(x^k)$ and $H := \nabla^2_{xx}\mathcal{L}(x^k,\gamma^k)$. Then, the linearized KKT condition (2.9) can be written as

$$\begin{bmatrix} H & J^\top \\ J & \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \gamma \end{bmatrix} = - \begin{bmatrix} g + (J)^\top \gamma \\ c \end{bmatrix} \tag{2.10}$$

The linear system (2.10) in each SQP iteration can alternatively be viewed as a standard quadratic program (QP)[§]

$$\min_{\Delta x \in \mathbb{R}^{n_x}} \quad \frac{1}{2} \Delta x^\top\ H\ \Delta x + (g)^\top\ \Delta x \tag{2.11a}$$

$$\text{s.t.} \quad J\ \Delta x\ +\ c\ =\ 0 \tag{2.11b}$$

with corresponding Lagrangian multipliers $\gamma^+ = \gamma + \Delta\gamma$. Therefore, if H is positive definite on the null space of J, i.e., satisfying

$$w^\top H w > 0, \quad \text{for all } w \in \mathcal{D}(x) \text{ with } w \neq 0,$$

[§]There are some quasi-Newton approximation methods to reduce computation burden in SQP algorithm, along with correction to an exact gradient, as shown in [58].

then a solution Δx to the linear system (2.10) is indeed the minimizer of the QP subproblem (2.11). In summary, obtaining the step $(\Delta x, \Delta\gamma)$ from the linear system (2.10) is equivalent to solving the QP subproblem (2.11).

Local Convergence

There exist various local convergence analyses of SQP methods. For example, [59, Thm 8.1] analyze the local convergence of inexact SQP type methods, as well as the local convergence rate depending on the choices of the Hessian matrix. A similar analysis is provided in [59, Thm 18.4]. We now provide an analysis of the local convergence for SQP methods, combining insights from both [46] and [59].

To proceed, we first introduce some mild assumption:

Assumption 2.1. *The primal-dual point (x^*, γ^*) is a local solution of Problem (2.1) at which the following conditions holds*

(a) The functions f and c are twice differentiable in the neighbourhood of x^* with Lipschitz continuous second derivatives.

(b) The linear independence constraint qualification (Definition 2.3) holds at x^*.

(c) The strict complementarity condition (Definition 2.4) holds at (x^*, γ^*).

(d) The second order sufficient condition (Theorem 2.2) holds at (x^*, γ^*).

The first condition ensures that $\mathcal{F}(x, \gamma)$ is Lipschitz continuous, while the remaining conditions imply that (x, γ) is a regular KKT point. For simplicity, we denote $\xi = (x, \gamma)$ to represent primal-dual points and use $\mathcal{F}_k$ to refer to $\mathcal{F}(\xi^k)$ throughout the proof.

We consider an approximation $B_k \approx \nabla\mathcal{F}(\xi^k)^\top$ and define the corresponding inexact Newton step as:

$$\Delta\xi^k = -B_k^{-1}\mathcal{F}_k. \tag{2.12}$$

To proceed with the convergence analysis, we introduce an additional assumption for the approximation:

Assumption 2.2 (Compatibility Condition [59, Ch 8.1]). *The matrix B_k is nonsingular and bounded, such that $\|B_k^{-1}\| \leq \beta < \infty$. Moreover, there exists a constant $\varrho < 1$ such that for all ξ^k, the following holds:*

$$\|B_k - \nabla\mathcal{F}_k\| \leq \varrho_k \leq \varrho. \tag{2.13}$$

Theorem 2.3 (Local convergence of inexact SQP [59, Thm 8.1]). *Suppose that Assumptions 2.1 and Assumptions 2.2 hold. Then, if (x^0, γ^0) is sufficiently close to (x^*, γ^*), sequence (x^k, γ^k) generated by (2.12) converges to (x^*, γ^*)*

Proof. Since $\xi^* = (x^*, \gamma^*)$ is a local solution of Problem (2.1), i.e., $\mathcal{F}(q^*) = \mathcal{F}_* = 0$, we have

$$
\begin{aligned}
\xi^{k+1} - \xi^* =& \xi^k + \Delta\xi^k - \xi^* \\
& \Downarrow \text{ Newton Step (2.12)} \\
=& \xi^k - \xi^* - B_k^{-1}\,\mathcal{F}_k \\
=& \xi^k - \xi^* - B_k^{-1}(\mathcal{F}_k - \mathcal{F}_*) \\
& \Downarrow \text{ Mean Value Theorem (Theorem A.2)} \\
=& \xi^k - \xi^* - B_k^{-1}\int_0^1 \nabla\mathcal{F}(\xi^k + \tau(\xi^* - \xi^k))^\top(\xi^k - \xi^*)d\tau \\
=& B_k^{-1}(B_k - \nabla\mathcal{F}_k^\top)(\xi^k - \xi^*) - B_k^{-1}\int_0^1 \left\{\nabla\mathcal{F}_k - \nabla\mathcal{F}(\xi^k + \tau(\xi^* - \xi^k))\right\}^\top(\xi^k - \xi^*)d\tau
\end{aligned}
$$

Taking the norm of both sides:

$$
\begin{aligned}
\left\|\xi^{k+1} - \xi^*\right\| \leq& \left\|B_k^{-1}\right\|\left\|B_k - \nabla\mathcal{F}_k^\top\right\|\left\|\xi^k - \xi^*\right\| \\
&+ \left\|B_k^{-1}\right\|\int_0^1 \left\|\nabla\mathcal{F}_k - \nabla\mathcal{F}(\xi^k + \tau(\xi^* - \xi^k))\right\|\left\|(\xi^k - \xi^*)\right\| d\tau \\
& \Downarrow \text{ Compatibility Condition (Assumption 2.2) \& Lipschitz Continuous (Definition A.1)} \\
\leq& \beta\varrho_k\left\|\xi^k - \xi^*\right\| + \beta\int_0^1 L\,\tau\left\|\xi^k - \xi^*\right\|^2 d\tau \\
=& \beta\left(\varrho_k + \frac{L}{2}\left\|\xi^k - \xi^*\right\|\right)\left\|\xi^k - \xi^*\right\| \\
\leq& \beta\left(\varrho + \frac{L}{2}\left\|\xi^k - \xi^*\right\|\right)\left\|\xi^k - \xi^*\right\| \qquad (2.14)
\end{aligned}
$$

where L is Lipschitz constant of $\nabla\mathcal{F}$ near ξ^*. Choosing ξ_0 in the neighborhood of

$$
\mathcal{B}(\xi^*) = \left\{\xi \mid \left\|\xi^0 - \xi^*\right\| < \frac{2(1 - \varrho\beta)}{L}\right\}
$$

the sequence ξ^k converges to ξ^*. □

The convergence rate may be faster, depending on the values of ϱ_k. For a more detailed local convergence analysis for different Hessian approximations in the context of SQP methods, we refer to [46, Ch 18.7].

Remark 2.3. *The condition* (2.14) *can be rewritten as*

$$\left\|\xi^{k+1} - \xi^*\right\| \leq \|B_k - \nabla \mathcal{F}_k\| \cdot \mathcal{O}\left(\left\|\xi^k - \xi^*\right\|\right) + \mathcal{O}(\left\|\xi^k - \xi^*\right\|^2),$$

indicating that a quadratic convergence rate can be achieved if the Hessian approximation is accurate enough so that

$$B_k - \nabla \mathcal{F}_k = \mathcal{O}\left(\left\|\xi^k - \xi^*\right\|\right) \tag{2.15}$$

Combinatorial Difficulty for Inequality Constraints

A significant challenge in solving generic NLP with inequality constraints involved is managing the inequality constraints effectively. Conventionally, there are two main approaches to extending the standard SQP for generic NLP (2.1) with inequality constraints involved. The first approach linearizes the inequality constraints and iteratively solves inequality-constrained QP subproblems

$$\min_{\Delta x \in \mathbb{R}^{n_x}} \quad \frac{1}{2} \Delta x^\top \, H \, \Delta x + g^\top \, \Delta x \tag{2.16}$$

$$\text{s.t.} \quad J \, \Delta x \; + \; c^{\mathrm{E}} \; = \; 0 \tag{2.17}$$

$$R \, \Delta x \; + \; c^{\mathrm{I}} \; \leq \; 0 \tag{2.18}$$

with $R = \nabla c^{\mathrm{I}}(x)$. However, due to the high computational cost associated with solving the generic QP at each iteration, this approach is expensive, especially for large-scale problems. Several open-source toolkits such as qpOASES [60], PIQP [61], OSQP [62], SCS [63], PROXQP [64] are available for solving the QP subproblem (2.16).

The second approach involves selecting an active set $\mathcal{A}(x)$ of constraints at each iteration and solving only equality-constrained QP subproblems. In this approach, inequality constraints in the active set $\mathcal{A}(x)$ are treated as equality constraints, while all the others are ignored. The active set is updated at each iteration, either based on Lagrange multiplier estimates or by solving an auxiliary subproblem. This approach is computationally more efficient, as equality-constrained QP subproblems are less costly to solve compared to the general QP program.

However, the number of possible active sets $\mathcal{A}(x)$ at a point x can be extremely large, i.e., up to $2^{N^{\mathrm{I}}}$, where N^{I} is the number of inequality constraints. This estimate arises from the fact that each inequality constraint can either be included in $\mathcal{A}(x)$ or left out. This exponential growth, referred

to as the *combinatorial difficulty* of NLP [46, Ch. 15.2], makes it impractical for large-scale problems.

2.3.2 Interior Point Method

Interior point methods (IPM), also known as barrier methods, have proven to be highly effective for nonlinear optimization, similar to their success in linear programming. One of the key advantages of interior-point methods is that they bypass the combinatorial difficulty in identifying active constraints, discussed in the previous section (Section 2.3.1). Given a NLP Problem (2.1), instead of dealing directly with inequality constraints, IPM incorporates the inequality constraints into the objective function through a logarithmic barrier term, expressed as:

$$\min_{x,s} \quad \varphi(x,s,\mu) = f(x) - \mu \sum_{m\in\mathcal{C}^{\mathrm{I}}} \ln(s^{(m)}) \tag{2.19a}$$

$$\text{s.t.} \quad c^{\mathrm{E}}(x) = 0 \qquad | \ \gamma, \tag{2.19b}$$

$$c^{\mathrm{I}}(x) + s = 0 \qquad | \ \kappa, \tag{2.19c}$$

The KKT conditions (2.3) for the barrier Problem (2.19) can be written as

$$\nabla f(x) + R^\top \kappa + J^\top \gamma = 0 \tag{2.20a}$$

$$-\mu \mathcal{S}^{-1}\mathbb{1} + \kappa = 0 \tag{2.20b}$$

$$c^{\mathrm{E}}(x) = 0 \tag{2.20c}$$

$$c^{\mathrm{I}}(x) + s = 0 \tag{2.20d}$$

where

$$R = \nabla c^{\mathrm{I}}(x)^\top, \ J = \nabla c^{\mathrm{E}}(x)^\top, \text{ and } \mathcal{S} = \operatorname{diag}(s),$$

and $\mathbb{1} \in \mathbb{R}^{N^{\mathrm{I}}}$ represent a column vector of ones.

Note that (2.20) is also called *perturbed KKT condition*. If μ in (2.20b) driven to zero, together with

$$s \geq 0 \text{ and } \kappa \geq 0, \tag{2.21}$$

the condition (2.20) is equivalent to the KKT conditions (2.3) for the original Problem (2.1). Therefore, if local minimizers x^* of the original Problem (2.1) exist, then

$$\lim_{\mu\to 0} \ x^\star(\mu) = x^*,$$

where $x^\star(\mu)$ represents the solutions to the barrier Problem (2.19) with a given μ:

$$x^\star(\mu) = \min_x \ (2.19a), \quad \text{s.t. } (2.19b) - (2.19c).$$

Appling a decreasing sequence of barrier parameters μ converging to zero, IPM can also be viewed as a homotopy method applied to the primal-dual equations (2.20).

To solve the barrier Problem (2.19) for a given fixed value μ of the barrier parameter, Newton's method is applied to find the root of the nonlinear systems (2.20):

$$\mathcal{F}^\mu(x, s, \gamma, \kappa) = \begin{bmatrix} \nabla f(x) + R^\top \kappa + J^\top \gamma \\ -\mu \mathbb{1} + \mathcal{S}\kappa \\ c^{\mathrm{E}}(x) \\ c^{\mathrm{I}}(x) + s \end{bmatrix} = 0$$

Similar to (2.9), the resulting Newton-step can be written as

$$\begin{bmatrix} \nabla^2_{xx}\mathcal{L} & 0 & J^\top & R^\top \\ 0 & \mathcal{K} & 0 & \mathcal{S} \\ J & 0 & 0 & 0 \\ R & I & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta s \\ \Delta \gamma \\ \Delta \kappa \end{bmatrix} = - \begin{bmatrix} r^x \\ r^s \\ r^{\mathrm{E}} \\ r^{\mathrm{I}} \end{bmatrix} \tag{2.22}$$

with $\mathcal{K} = \operatorname{diag}(\kappa)$ and

$$\begin{aligned} r^x &= \nabla \mathcal{L}(x, \gamma, \kappa), \\ r^s &= \mathcal{K}\mathcal{S} - \mu \mathbb{1}, \\ r^{\mathrm{E}} &= c^{\mathrm{E}}(x) \\ r^{\mathrm{I}} &= c^{\mathrm{I}}(x) + s. \end{aligned}$$

Remark 2.4. *The linear system (2.22) is referred to as the primal-dual system. In this dissertation, we focus on primal-dual IPM, as they offer greater numerical stability compared to the primal IPM. For a more detailed discussion, please refer to [46, Ch. 19.3].*

Instead of solving the nonsymmetric linear system (2.22) directly, the proposed method computes the solution equivalently by first solving the smaller, symmetric linear system by eliminating Δs, we have

$$\begin{bmatrix} \nabla^2_{xx}\mathcal{L} & J^\top & R^\top \\ J & 0 & 0 \\ R & 0 & -\kappa \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\gamma \\ \Delta\kappa \end{bmatrix} = - \begin{bmatrix} r^x \\ r^{\mathrm{E}} \\ \hat{r}^{\mathrm{I}} \end{bmatrix}$$

with $\hat{r}^{\mathrm{I}} = c^{\mathrm{I}}(x) + \mu\mathcal{K}^{-1}\mathbb{1}$. It can be further reformulated by eliminating $\Delta\kappa$:

$$\begin{bmatrix} H & J^\top \\ J & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\gamma \end{bmatrix} = - \begin{bmatrix} \hat{r}^x \\ r^{\mathrm{E}} \end{bmatrix}, \tag{2.23}$$

with

$$\begin{aligned} H &= \nabla^2_{xx}\mathcal{L}(x,\gamma,\kappa) + R^\top \mathcal{S}^{-1}\mathcal{K}R \\ \hat{r}^x &= \nabla\mathcal{L}(x,\gamma,\kappa) + R^\top \mathcal{S}^{-1}(\mathcal{K}c^{\mathrm{I}}(x) + \mu\mathbb{1}). \end{aligned}$$

Note that the resulting Newton step (2.23) resembles the one by SQP method (2.20), with

$$\begin{bmatrix} H & J^\top \\ J & 0 \end{bmatrix}$$

referred to as the perturbed KKT matrix. Once the linear system (2.23) is solved, Δs and $\Delta\kappa$ can be recovered by

$$\begin{aligned} \Delta s &= -c^{\mathrm{I}} - s - R\,\Delta x \\ \Delta\kappa &= -\kappa + \mathcal{S}^{-1}(\mu\, e - \mathcal{K}\Delta s). \end{aligned}$$

Having computed the Newton direction, primal and dual steplengths $\alpha^{\mathrm{p}}, \alpha^{\mathrm{d}} \in (0,1]$ have to be determined to obtain the next iterate as

$$x^+ = x + \alpha^{\mathrm{p}}\Delta x \tag{2.24a}$$
$$s^+ = s + \alpha^{\mathrm{p}}\Delta s \tag{2.24b}$$
$$\kappa^+ = \kappa + \alpha^{\mathrm{d}}\Delta\kappa \tag{2.24c}$$
$$\gamma^+ = \gamma + \alpha^{\mathrm{d}}\Delta\gamma \tag{2.24d}$$

To maintain positive definiteness of s and κ, we use the fraction-to-the-boundary rule:

$$\alpha^{\text{p,max}} = \{\alpha \in (0,1] : \; s + \alpha \Delta s \geq \max(1 - \tau_{\min}, \; \mu)\; s\} \tag{2.25a}$$

$$\alpha^{\text{d,max}} = \{\alpha \in (0,1] : \; \kappa + \alpha \Delta \kappa \geq \max(1 - \tau_{\min}, \; \mu)\; \kappa\}, \tag{2.25b}$$

where τ denotes the fraction-to-boundary parameter.

Remark 2.5 (Global Convergence of IPM). *Convergence of IPM for NLP typically involves two main steps. First, with a fixed barrier parameter μ, IPM solves an equality-constrained quadratic programming, similar to the performance analysis of SQP. Then, a globalization strategy is applied to ensure convergence of the method to the solution $x^\star(\mu)$ globally. Once convergence is achieved, the value of μ is decreased monotonically, and this process is repeated until μ approaches zero. For more detailed convergence analysis, we refer to [65, 66, 67, 68, 69]*

Remark 2.6. *The trajectory followed by the sequence $x^\star(\mu)$ as it approaches the solution is known as the central path [70]. The most effective performance has been observed with "long-step" methods, where IPM approximates the barrier problems (2.19) rather than finding exact solutions, and reduces μ at a faster rate. A numerically stable approach for this is detailed in [71], while adaptive strategies for updating the barrier parameter μ offer additional robustness [72, 73].*

IPMs perform particularly well in large-scale problems, often surpassing active-set methods by leveraging the consistent block structure of the linear systems solved at each iteration [74]. However, one drawback of IPMs is that they process all constraints at each step, even those irrelevant to the solution, which can lead to high computational costs in some cases. Despite their strengths, IPM are sensitive to the choice of the initial point, scaling of the problem, and the strategy for updating the barrier parameter [75]. If iterates approach the boundaries of the feasible region too quickly, convergence may be slow. There is a vast literature on nonlinear interior-point methods. We refer the reader to the surveys [76, 77] for a comprehensive list of references.

Numerous software packages are available for implementing nonlinear IPM, which have become strong contenders against active-set and augmented Lagrangian techniques. Those include LOQO [78], KNITRO [79], IPOPT [71]. KNITRO also offers a trust-region algorithm and can switch between interior-point and active-set modes. Some recent research focuses on using GPU-based acceleration for solving linear KKT systems [80, 81], and developing IPOPT-type GPU nonlinear solver, i.e., MadNLP [82].

2.3.3 Techniques for the QP subproblems

In the previous section, we discussed iterative methods for nonlinear optimization, all of which involve sequentially solving quadratic programming (QP) subproblems. In this section, we introduce methods and technologies used for solving these QP subproblems, specifically focusing on the linear systems:

$$\begin{bmatrix} H & J^\top \\ J & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \gamma \end{bmatrix} = - \begin{bmatrix} r^x \\ r^\gamma \end{bmatrix} \tag{2.26}$$

with $\Delta x \in \mathbb{R}^n$ and $\Delta \gamma \in \mathbb{R}^m$.

We first present a practical condition under which the existence and uniqueness of a local solution to the given equality QP subproblems (2.26) is guaranteed, along with a method to validate this condition. Additionally, we introduce dimension reduction strategies for these equality QP subproblems, which are useful for both convergence analysis and distributed algorithms, and will be applied in the following Chapter 5.

Inertia of KKT Matrix

Before we go into Hessian regularization, we introduce the definition of inertia of a symmetric matrix:

Definition 2.6 (Inertia [83]). *The inertia of a symmetric matrix K is the triple*

$$\textit{inertia}(K) = (n^+, n^-, n^0).$$

where n^+, n^-, and n^0 denote the number of positive, negative, and zero eigenvalues of K, respectively.

Theorem 2.4 ([83, Thm. 2.1]). *Let $K \in \mathbb{R}^{(m+n)\times(m+n)}$ be the KKT matrix*

$$K = \begin{bmatrix} H & J^\top \\ J & 0 \end{bmatrix},$$

let the Jacobian $J \in \mathbb{R}^{m\times n}$ be full row rank, and let $Z \in \mathbb{R}^{n\times(n-m)}$ be the null space of J. Then, if

$$\textit{inertia}(K) = (n,\ m,\ 0), \tag{2.27}$$

$Z^\top H Z$ is positive definite, and the corresponding QP has a strict minimizer.

Theorem 2.4 demonstrates that the inertia of the KKT matrix indicates whether the problem is locally convex at the current iteration [84]. In convex cases, the KKT matrix always possesses the correct inertia, allowing for the use of standard sparse factorization routines where pivot selection is based on sparsity and numerical stability. In nonconvex cases, however, it is essential to determine the inertia during the factorization process to adjust the matrix a posteriori.

For iterative methods applied to NLP, ensuring that the inertia condition (2.27) is satisfied and modifying the inertia when necessary during iterations is critical. A practical and efficient strategy for interior point method (IPM) is implemented in IPOPT [71], a well-known open-source software package for large-scale nonlinear optimization. The inertia correcttion algorithm used in IPOPT is outlined in Algorithm 1.

Algorithm 1 Inertia Correction [71, Alg. IC]

Input: δ_x^{last}, $\bar{K} = \begin{bmatrix} H & J^\top \\ J & 0 \end{bmatrix}$

$(n^+, n^-, n^0) = \text{inertia}(\bar{K})$
if the inertia condition (2.27) satisfied **then**
    $\delta_x^{\text{last}} = 0$
else
    if $\delta_x^{\text{last}} = 0$ **then**
        $\eta^{\text{inc}} = \eta^{\text{fast}}$ and $\delta_x = \delta_x^{\text{init}}$
    else
        $\eta^{\text{inc}} = \eta^{\text{slow}}$ and $\delta_x = \max\left(\eta^{\text{red}}\, \delta_x^{\text{init}},\ \delta_x^{\text{min}}\right)$
    end if
    if $n^0 = 0$ **then**
        $\delta_\gamma = 0$
    else
        $\delta_\gamma = \delta_\gamma^{\text{init}}$
    end if
    while the inertia condition (2.27) not satisfied **do**
        $(n^+, n^-, n^0) = \text{inertia}(\bar{K}) = \text{inertia}\left(\begin{bmatrix} H + \delta_x I & J^\top \\ J & -\delta_\gamma I \end{bmatrix}\right)$
        $\delta_x^{\text{last}} = \delta_x$
        $\delta_x = \eta^{\text{inc}} \delta_x$
    end while
end if
Return $\delta^{\text{last}} = \delta_x$

Algorithm 1 starts in Line 2 by checking whether the unmodified KKT matrix has the desired inertia. This ensures that the Newton search direction can be used whenever possible. If this check fails, progressively larger values of the regularization parameter, δ_x, are applied. The initial trial value of δ_x (Lines 5–9) is based on the value from the previous Newton step, which preserves the perturbation from the last modification of the iteration matrix. This method aims to find the smallest necessary perturbation (within a factor) while avoiding unnecessary factorizations for values of δ_x that are too small.

Additionally, in Lines 5–9, a larger factor η^{inc} is used to increase parameter δ_x later in Line 18 if there was no inertia correction in the previous iteration. This strategy minimizes the number of trial factorizations early in the process when the scale of the problem and the appropriate perturbation size are still uncertain.

A nonzero value of δ_γ is always chosen when the unmodified iteration matrix has a zero eigenvalue (Lines 10–14), assuming that the singularity is caused by a rank-deficient constraint Jacobian.

The work by Wächter and Biegler [71] also provides an empirical parameter setting that performs well in practice for generic NLP (2.1). In the IPOPT package, the recommended parameter settings are

$$\delta_x^{\text{init}} = 10^{-4},\ \delta_\gamma = 10^{-8},\ \eta^{\text{red}} = \frac{1}{3},\ \eta^{\text{fast}} = 100,\ \text{and}\ \eta^{\text{slow}} = 8$$

For more details, we refer to [71, Sec. 3].

Remark 2.7. *Recent research has shifted towardsinertia-free approaches [85, 81] for solving quadratic subproblems, particularly when leveraging graphics processing units (GPU) to accelerate generic NLP solvers, such as MadNLP [82].*

Schur-Complement Method

Assuming that H is nonsingular and LICQ (Definition 2.3) holds, the linear system (2.26) can be written as

$$\begin{bmatrix} H & J^\top \\ 0 & -JH^{-1}J^\top \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\gamma \end{bmatrix} = - \begin{bmatrix} r^x \\ \hat{r}^\gamma \end{bmatrix} \tag{2.28}$$

with

$$\hat{r}^\gamma = r^\gamma - JH^{-1}r^x$$

by applying block Gaussian elimination. Following linear algebra terminology, the matrix $-JH^{-1}J^\top$ is the Schur complement of the KKT matrix K.

The resulting linear system (2.28) can be solved by first obtaining the optimal dual step by

$$\Delta\gamma^* = \left(JH^{-1}J^\top\right)^{-1}\left(r^\gamma - JH^{-1}r^x\right). \tag{2.29}$$

Once (2.29) is solved, the optimal primal step can be recovered by

$$\Delta x^* = -H^{-1}\left(r^x + J\Delta\gamma^*\right). \tag{2.30}$$

The Schur complement method involves operations with H^{-1} and factoring the $m \times m$ matrix $JH^{-1}J^\top$. It is most efficient when H is easy to invert (e.g., diagonal or block-diagonal), H^{-1} is available via a quasi-Newton update, or the number of equality constraints m is small, reducing the need for backsolves. Although its effectiveness can vary depending on the problem, it is generally recommended when H is positive definite and $JH^{-1}J^\top$ can be computed efficiently.

Remark 2.8. *The Schur complement method is particularly effective when*

1. *H is easy to invert*
2. *the number of equality constraints, denoted by m, is small, which reduces the complexity of computing $H^{-1}J^\top$.*

2.4 Introduction to Distributed Algorithms for Nonlinear Programming

Given an NLP Problem (2.1), centralized optimization addresses the NLP Problem (2.1) as a single, unified problem, whereas distributed optimization tackles separate subproblems locally and coordinates them through synchronous or asynchronous updates.

Although centralized optimization is straightforward, it aggregates all information in a single entity. This limitation makes distributed optimization techniques crucial, particularly for large-scale or privacy-sensitive problems. Distributed optimization often uses affinely coupled problem structures, where the original NLP problem (2.1) can be reformulated as:

$$\min_{x} \quad f(x) = \sum_{\ell \in \mathcal{R}} f_\ell(x_\ell) \tag{2.31a}$$

$$\text{s.t.} \quad \sum_{\ell \in \mathcal{R}} A_\ell x_\ell = b \quad | \quad \lambda \tag{2.31b}$$

$$c_\ell^{\mathrm{E}}(x_\ell) = 0 \quad | \quad \gamma_\ell, \quad \forall \ell \in \mathcal{R} \tag{2.31c}$$

$$c_\ell^{\mathrm{I}}(x_\ell) \leq 0 \quad | \quad \kappa_\ell, \quad \forall \ell \in \mathcal{R} \tag{2.31d}$$

Here, $\mathcal{R}$ denotes the set of regions resulting from the partition, with x partitioned into N^{reg} local subvectors x_ℓ for all $\ell \in \mathcal{R}$. The local objectives $f : \mathbb{R}^{N_\ell^x} \to \mathbb{R}$, equality constraints $c^{\mathrm{E}} : \mathbb{R}^{N_\ell^x} \to \mathbb{R}^{N_\ell^{\mathrm{E}}}$, and inequality constraints $c^{\mathrm{I}} : \mathbb{R}^{N_\ell^x} \to \mathbb{R}^{N_\ell^{\mathrm{I}}}$ are defined for each region $\ell \in \mathcal{R}$. The consensus constraints (2.31b) ensure the consistency of the coupling variables across different regions. This consistency is crucial to guarantee that the solutions obtained from the distributed formulation (2.31) are equivalent to those of the original centralized NLP (2.1).

The *Lagrangian function* of the distributed Problem (2.31) can be written as

$$\begin{aligned}\mathcal{L}(x, \lambda, \kappa, \gamma) =& f(x) + \gamma^\top c^{\mathrm{E}}(x) + \kappa^\top c^{\mathrm{I}}(x) + \lambda^\top (Ax - b) \\ =& \sum_{\ell \in \mathcal{R}} \left\{ f_\ell(x_\ell) + \gamma_\ell^\top c_\ell^{\mathrm{E}}(x_\ell) + \kappa_\ell^\top c_\ell^{\mathrm{I}}(x_\ell) \right\} + \lambda^\top (Ax - b).\end{aligned}$$

where the matrix A is formed by horizontally concatenating A_ℓ for all $\ell \in \mathcal{R}$, i.e.,

$$A = \mathrm{horzcat}(A_\ell),$$

while c^{E} and c^{I} are vertically concatenated vectors of equality and inequality constraints, respectively, for all $\ell \in \mathcal{R}$. The corresponding KKT conditions read

$$\nabla f_\ell(x_\ell) + R_\ell^\top \kappa + J_\ell^\top \gamma_\ell + A_\ell^\top \lambda = 0, \quad \forall \ell \in \mathcal{R} \tag{2.32a}$$

$$c_\ell^{\mathrm{E}}(x_\ell) = 0, \quad \forall \ell \in \mathcal{R} \tag{2.32b}$$

$$c_\ell^{\mathrm{I}}(x_\ell) \leq 0, \ \kappa^\top c_\ell^{\mathrm{I}}(x_\ell) = 0, \ \kappa \geq 0, \quad \forall \ell \in \mathcal{R} \tag{2.32c}$$

$$\sum_{\ell \in \mathcal{R}} A_\ell \, x_\ell = b \tag{2.32d}$$

with Jacobian matrix

$$R_\ell = \nabla c_\ell^{\mathrm{I}}(x_\ell)^\top \text{ and } J_\ell = \nabla c_\ell^{\mathrm{E}}(x_\ell)^\top.$$

2.4.1 Dual Decomposition

Dual decomposition is a foundational concept in distributed optimization, originating in the early 1960s through the work on large-scale linear programming [86][87]. The general idea of dual decomposition is first stated in [88]. Bertsekas provides a thorough reference on dual-based methods and decomposition for nonlinear programming [70]. Boyd et al.'s influential booklet [10] later unified these ideas within the ADMM framework, highlighting its broad applicability and extending convergence analysis from earlier works.

The main idea of dual decomposition is to solve a problem without local equality & inequality constraints:

$$\min_{x} \quad f(x) := \sum_{\ell \in \mathcal{R}} f_\ell(x_\ell) \tag{2.33a}$$

$$\text{s.t.} \quad \sum_{\ell \in \mathcal{R}} A_\ell x_\ell = b \quad | \quad \lambda \tag{2.33b}$$

It proceeds by solving its dual problem

$$\max_{\lambda} \; d(\lambda) \tag{2.34a}$$

with

$$d(\lambda) = \min_{x} \sum_{\ell \in \mathcal{R}} \left\{ f_\ell(x_\ell) + \lambda^\top A_\ell x_\ell - \frac{1}{N^{\text{reg}}} \lambda^\top b \right\}.$$

Note that its Lagrangian function is separable:

$$\mathcal{L}(x, \lambda) = \sum_{\ell \in \mathcal{R}} \mathcal{L}_\ell(x_\ell, \lambda) = \sum_{\ell \in \mathcal{R}} \left\{ f_\ell(x_\ell) + \lambda^\top A_\ell x_\ell - \frac{1}{N^{\text{reg}}} \lambda^\top b \right\}.$$

In a word, solves the dual maximization problem by using the dual gradient method as follows,

$$x_\ell^{k+1} = \arg\min_{x} \mathcal{L}_\ell(x_\ell, \lambda) \tag{2.35a}$$

$$\lambda^{k+1} = \lambda^k + \alpha^k \left(A x^{k+1} - b \right) \tag{2.35b}$$

where $\alpha^k > 0$ denotes the steplength of the dual gradient method. If the steplength α is chosen appropriately and certain assumptions hold [10], the primal variable x^k and dual variable λ^k converge. However, these assumptions often do not hold in practice, limiting the use of dual ascent. For more details about dual decomposition and its recent development, we refer to a recent review [12].

2.4.2 Augmented Lagrangian Method

The augmented Lagrangian method (ALM) provides an alternative to choose the steplength for dual decomposition. The concept of augmented Lagrangians and the method of multipliers for constrained optimization was first introduced in the late 1960s by Hestenes [89] and Powell [90].

Consider an NLP with linear equality constraint similar to (2.33):

$$\min_{x} \quad f(x) \tag{2.36a}$$

$$\text{s.t.} \quad Ax - b = 0 \quad | \quad \lambda \tag{2.36b}$$

and the KKT condition reads

$$\nabla f(x^*) + A^\top \lambda^* = 0 \tag{2.37a}$$

$$Ax - b = 0. \tag{2.37b}$$

The corresponding *augmented Lagrangian function* is defined as

$$\mathcal{L}^\rho(x, \lambda) = f(x) + \lambda^\top (Ax - b) + \frac{\rho}{2} \|Ax - b\|_2^2$$

and is equivalent to the Lagrangian function of an auxiliary problem

$$\min_{x} \quad f(x) + \frac{\rho}{2} \|Ax - b\|_2^2 \tag{2.38a}$$

$$\text{s.t.} \quad Ax - b = 0 \tag{2.38b}$$

because for any feasible x, the $L2$ term added to the objective is zero. The *dual function* d^ρ with respect to the auxiliary problem (2.38) can then be written as

$$d^\rho(\lambda) = \min_{x} \mathcal{L}^\rho(x, \lambda) = \mathcal{L}^\rho\left(x^\rho(\lambda), \lambda\right)$$

where $x^\rho(\lambda)$ defined as

$$x^\rho(\lambda) = \arg\min_{x} \mathcal{L}^\rho(x, \lambda).$$

Iterations of ALM consist of two steps:

$$x^{k+1} = \arg\min_{x} \mathcal{L}^\rho(x, \lambda^k) \tag{2.39a}$$

$$\lambda^{k+1} = \lambda^k + \rho \left(Ax^{k+1} - b\right) \tag{2.39b}$$

In the x-minimization step (2.39a), the primal state variables are updated by minimizing the augmented Lagrangian function $\mathcal{L}^\rho$ with current estimation of dual variables λ^k. Consider that x^{k+1} minimizing $\mathcal{L}^\rho(x, \lambda^k)$, then

$$\begin{aligned} 0 &= \nabla_x \mathcal{L}^\rho(x^{k+1}, \lambda^k) \\ &= \nabla f(x^{k+1}) + A^\top \left(\lambda^k + \rho \left(Ax^{k+1} - b\right)\right). \end{aligned}$$

Therefore, updating $\lambda^k + \rho \left(Ax^{k+1} - b\right)$ is a better estimation of dual variables at the point x^{k+1} to approximate the optimal condition (2.37a). ALM can be interepreded as gradient ascent for solving the dual problem:

$$\max_{\lambda} \; d^\rho(\lambda).$$

Compared with (2.35b), the penalty ρ is used as the steplength in the dual update to ensure that the iterates (x^{k+1}, y^{k+1}) remain dual feasible. As the algorithm progresses, the primal residual $c(x^k)$ converges to zero, leading to optimality. While the ALM offers significantly better convergence compared to dual decomposition, this improvement comes at a cost. When f is separable, the augmented Lagrangian L_ρ is no longer separable, which prevents the x-minimization step (2.39a) from being solved in parallel for each x_i. This limitation means that the basic ALM cannot be applied directly for decomposition.

The challenge of non-separability can be addressed, as we will explore in the next section. For more details about ALM, we refer to a recent review [70].

2.4.3 Alternating Direction Method of Multipliers

To handle the non-separability, the alternating direction method of multipliers (ADMM) combines dual decomposition and augmented Lagrangian method (ALM). We first discuss the simplest case, i.e. a 2-block problem.

Two-Block ADMM

The alternating direction method of multipliers (ADMM) is designed for solving problems in a two-block form:

$$\min_{x, \bar{x}} \quad f(x) + h(\bar{x}) \tag{2.40a}$$

$$\text{s.t.} \quad Bx + D\bar{x} = b, \quad | \; \lambda \tag{2.40b}$$

with primal variables $x \in \mathbb{R}^{N^x}$ and $\bar{x} \in \mathbb{R}^{N^z}$ and dual variable $\lambda \in \mathbb{R}^{N^\lambda}$, which is the Lagrangian multiplier with respect to the affine consensus constraints (2.40b). The corresponding augmented Lagrangian function can be written as

$$\mathcal{L}^\rho(x, \bar{x}, \lambda) = f(x) + h(\bar{x}) + \langle \lambda, Bx + D\bar{x} \rangle + \frac{\rho}{2} \|Bx + D\bar{x} - b\|_2^2 .$$

Different from performing a joint minimization with respect to x and $\bar{x}$ in the first step (2.39a) by ALM, ADMM iterates

$$x^{k+1} = \arg\min_x \mathcal{L}^\rho(x, \bar{x}^k, \lambda^k) \tag{2.41a}$$

$$\bar{x}^{k+1} = \arg\min_{\bar{x}} \mathcal{L}^\rho(x^{k+1}, \bar{x}, \lambda^k) \tag{2.41b}$$

$$\lambda^{k+1} = \lambda^k + \rho(Bx^{k+1} + D\bar{x}^{k+1} - b) \tag{2.41c}$$

update primal variables x and $\bar{x}$ in an alternative fashion.

A disadvantage of the classic ADMM (2.41) is that the blocks are updated one after another, which is not amenable for parallelization [91]. Moreover, a direct extension of ADMM (2.41) for multi-block problems is not necessarily convergent, even for convex problems [92].

Parallelizable Multi-Block ADMM

To solve a multi-block problem by ADMM in parallel, a popular choice is to introduce a local state variable x and a global variable z, and to reformulate the original distributed optimization problem (2.33) as

$$\min_{x, z \in \mathcal{Z}} \quad \sum_{\ell \in \mathcal{R}} f_\ell(x_\ell) \tag{2.42a}$$

$$\text{s.t.} \quad x_\ell - z_\ell = 0 \quad | \ \lambda_\ell, \ \forall \ell \in \mathcal{R} \tag{2.42b}$$

with

$$\mathcal{Z} = \{z \mid \sum_{\ell \in \mathcal{R}} A_\ell z_\ell = 0\}.$$

The resulting augmented Lagrangian function

$$\mathcal{L}^\rho(x, z, \lambda) = \sum_{\ell \in \mathcal{R}} \left\{ f_\ell(x_\ell) + \lambda_\ell^\top (x_\ell - z_\ell) + \frac{\rho}{2} \|x_\ell - z_\ell\|_2^2 \right\}, \ \forall z \in \mathcal{Z} \tag{2.43}$$

is separable and ADMM can be applied here to solve the $|\mathcal{R}|+1$ blocks problems, as outlined in Algorithm 2. Step 1 solves $|\mathcal{R}|$ decoupled problems constructed according to the augmented Lagrangian function (2.43) with respect to x, i.e.,

$$x_\ell^+ = \arg\min_{x_\ell} \mathcal{L}(x, z, \lambda), \quad \forall \ell \in \mathcal{R}$$

Step 2 updates the dual iterate λ_ℓ based on the gradient ascent method. Notice that both the Step 1 and 2 can be executed in parallel. Step 3 deals with a consensus QP, i.e.,

$$z^+ = \arg\min_{z \in \mathcal{Z}} \mathcal{L}(x^+, z, \lambda^+),$$

whose solution can be worked out analytically, requiring one to collect x_ℓ^+ and λ_ℓ^+. Once (2.45) is solved, the solution z^+ is broadcast to local agents, and the algorithm returns to Step 1.

Algorithm 2 ADMM

Input: z, λ, $\rho > 0$

Repeat:

1 update x_ℓ by solving $|\mathcal{R}|$ decoupled NLP problems

$$x_\ell^+ = \arg\min_{x} \quad f_\ell(x_\ell) + \lambda_\ell^\top x_\ell + \frac{\rho}{2} \|x_\ell - z_\ell\|_2^2 \tag{2.44}$$

2 compute $\lambda_\ell^+ = \lambda_\ell + \rho(x_\ell - z_\ell)$, for all $\ell \in \mathcal{R}$.

3 update z by solving the coupled averaging step

$$z^+ = \arg\min_{z} \quad \sum_{\ell \in \mathcal{R}} \left\{ -z_\ell^\top \lambda_\ell^+ + \frac{\rho}{2} \left\| x_\ell^+ - z_\ell \right\|_2^2 \right\} \tag{2.45a}$$

$$\text{s.t.} \quad \sum_{\ell \in \mathcal{R}} A_\ell z_\ell = b \tag{2.45b}$$

Performance in Practical Applications

ADMM is particularly well-suited for distributed convex optimization, especially in large-scale problems from computer vision [93], statistic [94], machine learning [95], and related fields. It functions at a higher level of abstraction compared to traditional optimization algorithms like Newton's method, where operations focus on solving small convex problems, often with simple analytical solutions. For instance, in large-scale model fitting, each ADMM update reduces the

task to a smaller, regularized problem, which can be efficiently handled using standard algorithms suited for small to medium-sized problems. This modular coordination allows simpler algorithms to collaborate in solving global problems too large for individual methods.

However, while ADMM is effective in distributed settings and for reaching modest accuracy, it has notable drawbacks when high precision is required. Compared to algorithms like Newton's method, which achieves high-accuracy solutions in a reasonable time, ADMM's convergence slows down near the optimal solution. Moreover, ADMM can exhibit unstable behavior or fail to converge in nonconvex problems, unless specific conditions are met. In many nonconvex problems, standard ADMM may require modifications to achieve acceptable convergence, including proximal terms or adaptive parameter updates. Even with these adjustments, the algorithm often requires thousands of iterations to achieve acceptable results, even for small- to medium-sized problems, as noticed in power systems applications [29, 96, 32, 6]. These challenges place significant demands on communication infrastructure, requiring frequent data exchanges within short time periods.

Therefore, while ADMM is a powerful tool for distributed convex optimization, particularly for large-scale problems where moderate accuracy is sufficient, it may not be the ideal choice in scenarios that demand high precision, involve nonconvex problems, or have limited communication capacity.

2.5 Summary

This chapter lays out the mathematical framework required for understanding nonlinear programming problems. It covers essential notation, terminology, and methodology throughout this dissertation. Conventional centralized methods like sequential quadratic programming (SQP) and interior point method (IPM) are reviewed, as well as methods for solving the corresponding quadratic program (QP) subproblems. It also introduces the key elements of distributed optimization, including dual decomposition, alternating direction method of multipliers (ADMM), etc. The distributed optimization algorithms introduced in this chapter rely on first-order curvature information, which is computationally efficient for iterative processes and communication. However, while this approach performs well for convex problems, it may struggle with nonconvex feasible sets, and even when successful, it often requires thousands of iterations to converge, increasing the communication burden. This motivates the further development of distributed optimization algorithms.

3 ALADIN: A Distributed SQP Approach

As introduced in Chapter 2, distributed optimization algorithms are extensively studied for convex problems arising in statistics, machine learning, and related areas [10]. However, when applied to nonconvex optimization problems, first-order distributed algorithms like the alternating direction method of multipliers (ADMM) often exhibit suboptimal performance. As a second-order alternative, the augmented Lagrangian based alternating direction inexact Newton (ALADIN) algorithm was developed for generic distributed nonlinear programming. By integrating elements of ADMM and sequential quadratic programming (SQP), ALADIN provides significant advantages in terms of convergence guarantees and speed, but its scalability and practical performance remain limited.

This chapter provides a detailed overview of the standard ALADIN algorithm and potential limitations affecting its scalability and efficiency. This serves as the foundation for the enhancements introduced in subsequent chapters.

Outline. This chapter begins with the motivation for adopting ALADIN over ADMM in Section 3.1. The standard ALADIN framework is then introduced in Section 3.2, followed by an analysis of its local and global convergence properties in Section 3.3. Section 3.4 discusses potential challenges associated with the standard ALADIN approach. Section 3.1 is partially based on [6], and Section 3.4 is based on discussion from [36, 37, 38].

3.1 Motivation: From ADMM to ALADIN

Existing distributed optimization algorithms—such as dual decomposition methods, the alternating direction method of multipliers (ADMM), and their variants—often struggle with nonconvex problems. Dual decomposition methods become ineffective in these cases due to the potential existence of a duality gap. Similarly, despite ADMM's success in convex optimization, as discussed

in Section 2.4.3, studies like [33, 6] have shown that ADMM can diverge when solving nonconvex problems. Moreover, in contrast to general Newton-type methods discussed in Section 2.3, ADMM is not invariant under scaling. This means that without applying a preconditioner to scale the optimization variables before running ADMM, the method may converge slowly or even diverge [10].

This leads to an important consideration: should second-order information be utilized in distributed or large-scale optimization? On the one hand, avoiding the computational overhead of linear algebra and matrix decomposition is desirable for large-scale problems. On the other hand, incorporating approximate second-order information—as done in many Newton-type methods discussed in Section 2.3—can improve convergence rates and robustness, especially regarding the scaling of optimization variables. Quasi-Newton methods like limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) demonstrate the potential benefits of approximating second-order terms [97, 98], offering a competitive alternative to purely gradient-based methods in large-scale optimization.

By integrating SQP into a distributed optimization framework, we can develop robust, competitive algorithms in terms of scaling. This brings us to ALADIN, a distributed algorithm with several key advantages:

- **Local superlinear or quadratic convergence rate:** ALADIN is locally equivalent to a Newton-type method, achieving superlinear or quadratic convergence under certain regularity conditions with appropriate Hessian approximations. If the objective or constraints are non-differentiable and these conditions are not met, ALADIN still guarantees convergence, albeit with slower rates.
- **Effective handling of nonconvex nonlinear problems:** While ADMM has been shown to perform well only under specific conditions [99, 32], ALADIN's local convergence properties remain robust regardless of adjustments to the augmented Lagrangian parameter. Numerical comparisons based on power network optimization problems [100, 6, 101] demonstrate that ALADIN successfully finds solutions in cases where ADMM fails to converge. Moreover, even when both methods converge, ALADIN achieves the same level of accuracy with significantly fewer iterations than ADMM.
- **Enhanced scalability in the absence of inequality constraints:** ALADIN demonstrates excellent scalability when inequality constraints are absent. Without the combinatorial complexities by SQP methods discussed in Section 2.3.1, ALADIN shows strong potential for handling large-scale problems. This is evidenced in the original paper [33] and in large-scale applications within power systems [6, 39], indicating its effectiveness for complex, real-world scenarios.

3.2 Standard ALADIN Algorithm

This section introduces the standard ALADIN outlined in Algorithm 3 and its globalization strategy in Algorithm 4, as well as some implementation details. Note that our discussion in this chapter assumes the QP and NLP subproblems are solved with high precision; exploring variants that solve these subproblems inexactly is beyond the scope of this chapter.

The ALADIN algorithm is proposed to solve generic distributed optimization problems in the form

$$\min_{x} \quad f(x) := \sum_{\ell \in \mathcal{R}} f_\ell(x_\ell) \tag{3.1a}$$

$$\text{s.t.} \quad \sum_{\ell \in \mathcal{R}} A_\ell x_\ell = b \qquad | \ \lambda \tag{3.1b}$$

$$c_\ell(x_\ell) \leq 0 \qquad | \ \kappa_\ell, \ \ell \in \mathcal{R}, \tag{3.1c}$$

where the region set is denoted as $\mathcal{R}$. The objective function $f_\ell : \mathbb{R}^{N_\ell^x} \to \mathbb{R}$ and local constraint $c_\ell : \mathbb{R}^{N_\ell^x} \to \mathbb{R}^{N_\ell^1}$ are assumed to be twice continuously differentiable for all $\ell \in \mathcal{R}$, but not necessarily convex. It is also assumed that Problem (3.1) is feasible and that all local minimizers are regular KKT points (Definition 2.5). The aim is to find a local minimizer numerically. The matrix $A_\ell \in \mathbb{R}^{N^{\text{cons}} \times N_\ell^x}$ and vector $b \in \mathbb{R}^{N^{\text{cons}}}$ in the (3.1b) are assumed to be given, modeling the dependencies between subsystems. The dual variables (Lagrangian multipliers) associated with the constraints (3.1b) and (3.1c) are denoted by $\lambda \in \mathbb{R}^{N^{\text{cons}}}$ and $\kappa_\ell \in \mathbb{R}^{N^1}$, respectively.

3.2.1 Standard ALADIN

The most basic variant of ALADIN proceeds by repeating two main steps. In the first step, one solves decoupled NLP problems for each region $\ell \in \mathcal{R}$ of the form

$$\begin{aligned} \min_{x_\ell} \quad & f_\ell(x_\ell) + \lambda^\top A_\ell x_\ell + \frac{\rho}{2} \|x_\ell - z_\ell\|_{\Sigma_\ell}^2 \\ \text{s.t.} \quad & c_\ell(x_\ell) \leq 0 \qquad | \ \kappa_\ell, \end{aligned}$$

where $\lambda \in \mathbb{R}^{N^{\text{cons}}}$ is the dual iterate associate with the multiplier of the consensus constraint (3.1b) and remains constant in the decoupled step. The decoupled NLPs are formulated following the idea of the augmented Lagrangian method. To enhance performance during iterations, we can adjust either the scaling matrices Σ_ℓ or penalty parameter ρ during the iterations to improve performance. A practical approach for updating ρ can be found in [34].

Note that the optimization problem is called *decoupled*, because the objective and constraints depend solely on the local objective function f_ℓ, local constraints c_ℓ and the matrix A_ℓ specific to each agent $\ell \in \mathcal{R}$. Consequently, these decoupled problems (3.2) can be solved in parallel without requiring communication between agents.

Algorithm 3 Standard ALADIN [33, Alg. 2]

Input: z, λ, $\rho > 0$, $\delta > 0$ and scaling symmetric matrices $\Sigma_\ell \succ 0$

Repeat:

1: solve the following decoupled NLPs for all $\ell \in \mathcal{R}$ ▷ *Parallel*

$$\min_{x_\ell} \quad f_\ell(x_\ell) + \lambda^\top A_\ell x_\ell + \frac{\rho}{2} \|x_\ell - z_\ell\|^2_{\Sigma_\ell} \tag{3.2a}$$

$$\text{s.t.} \quad c_\ell(x_\ell) \leq 0 \qquad | \ \kappa_\ell, \tag{3.2b}$$

2: compute the Jacobian matrix J_ℓ based on the active set at the local solution x_ℓ by ▷ *Parallel*

$$[J_\ell]_i = \begin{cases} \partial \, [c_\ell(x_\ell)]_i, & i \in \mathcal{A}_\ell(x_\ell) \\ 0, & i \notin \mathcal{A}_\ell(x_\ell) \end{cases} \tag{3.3}$$

where $[J_\ell]_i$ denotes the i-th row of matrix J_ℓ and $A_\ell(x_\ell) = \{i \mid [c_\ell(x_\ell)]_i = 0\}$ denotes the local active set, and gradient $g_\ell = \nabla f_\ell(x_\ell)$. Choose Hessian approximation

$$H_\ell \approx \nabla^2 \left\{ f_\ell(x_\ell) + \kappa_\ell^\top c_\ell(x_\ell) \right\} \succ 0. \tag{3.4}$$

3: solve coupled QP ▷ *Centralized*

$$\min_{\Delta x, s} \sum_{\ell \in \mathcal{R}} \left\{ \frac{1}{2} \Delta x_\ell^\top \ H_\ell \ \Delta x_\ell + g_\ell^\top \Delta x_\ell \right\} + \lambda^\top \ s + \frac{\delta}{2} \|s\|_2^2 \tag{3.5a}$$

$$\text{s.t.} \quad \sum_{\ell \in \mathcal{R}} A_\ell (x_\ell + \Delta x) = b + s \quad | \ \lambda^{\text{QP}} \tag{3.5b}$$

$$J_\ell \Delta x_\ell = 0, \quad \ell \in \mathcal{R} \ . \tag{3.5c}$$

4: update primal and dual variable z, λ by ▷ *Synchronize*

$$z^+ = x + \alpha_1 (x - z) + \alpha_2 \Delta x, \tag{3.6a}$$

$$\lambda^+ = \lambda + \alpha_3 (\lambda^{\text{QP}} - \lambda), \tag{3.6b}$$

In the context of ALADIN, the computation of decoupled NLP solutions is alternated with the second main step, namely, the solution of coupled subproblems in the coordination step:

$$\min_{\Delta x} \quad \mathcal{L}^{\text{PH}}(x + \Delta x, \lambda) \tag{3.7a}$$

$$\text{s.t.} \quad c(x + \Delta x) \leq 0 \tag{3.7b}$$

where c represents the vertically concatenated constraints c_ℓ for all $\ell \in \mathcal{R}$ and $\mathcal{L}^{\text{PH}}$ is *Powell-Hestenes augmented Lagrangian* [89, 90] with respect to the consensus constraints (3.1b):

$$\mathcal{L}^{\text{PH}}(x, \lambda) = \sum_{\ell \in \mathcal{R}} f_\ell(x_\ell) + \lambda^\top \left(\sum_{\ell \in \mathcal{R}} A_\ell x_\ell - b \right) + \frac{\delta}{2} \left\| \sum_{\ell \in \mathcal{R}} A_\ell x_\ell - b \right\|_2^2$$

with the penalty parameter δ.

Introducing the Powell-Hestenes augmented Lagrangian function ensures that the coupled subproblems remain feasible, regardless of whether the original Problem (3.1) is feasible. This guarantees robustness in the optimization process.

Instead of solving (3.7) directly, the coordination step of ALADIN solves an approximated QP subproblem at current iterate (x, λ), utilizing an active set method to manage inequalities. The resulting-equality constrained QP subproblems reads

$$\begin{aligned} \min_{\Delta x, s} \; & \sum_{\ell \in \mathcal{R}} \left\{ \frac{1}{2} \Delta x_\ell^\top \; H_\ell \; \Delta x_\ell + g_\ell^\top \Delta x_\ell \right\} + \lambda^\top \; s + \frac{\delta}{2} \left\| s \right\|_2^2 \\ \text{s.t.} \quad & \sum_{\ell \in \mathcal{R}} A_\ell (x_\ell + \Delta x_\ell) + r_\ell = s \quad \mid \lambda^{\text{QP}} \\ & J_\ell \Delta x_\ell = 0, \quad \ell \in \mathcal{R} \end{aligned}$$

with local consensus residual $r_\ell = A_\ell x_\ell$ for all $\ell \in \mathcal{R}$, where the curvature information, i.e., Jacobian of active constraints*

$$J_\ell = \frac{\partial c_\ell^{\text{act}}}{\partial x_\ell}(x_\ell),$$

the gradient of the local objective

$$g_\ell = \nabla f_\ell(x_\ell),$$

and the Hessian of local Lagrangian function

$$H_\ell = \nabla^2 \left\{ f(x_\ell) + \kappa_\ell^\top \; c_\ell(x_\ell) \right\}$$

are evaluated and provided by local agent ℓ.

The coordination step (Step 3) is the primary difference between ADMM and ALADIN. Differing from ADMM's dual gradient approach, ALADIN involves solving a quadratic programming (QP) problem during the coordination step.

*The active constraint $c_\ell^{\text{act}}(x_\ell)$ at the current iteration encompasses the inequality constraints for all $i \in \{i \mid [h_\ell(x_\ell)]_i = 0\}$.

In practice, solving the equality-constrained QP (3.5) requires only basic linear algebra routines, such as those provided by `LAPACK` [102], since it is equivalent to solving the associated linear equations of KKT system

$$\begin{bmatrix} H & A^\top & J^\top \\ A & -\delta I & 0 \\ J & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \lambda^{\mathrm{QP}} - \lambda \\ \kappa^+ \end{bmatrix} = - \begin{bmatrix} g + \lambda^\top A \\ Ax - b \\ 0 \end{bmatrix} \tag{3.8}$$

with identity matrix $I \in \mathbb{R}^{N^{\mathrm{cons}} \times N^{\mathrm{cons}}}$, Hessian $H = \mathrm{blkdiag}(H_\ell) \in \mathbb{R}^{N^x \times N^x}$, Jacobian $J = \mathrm{horzcat}(J_\ell)$ and consensus matrix $A = \mathrm{horzcat}(A_\ell)$. This makes the computational implementation efficient. When full-step is applied in Step 3, only the dual variable update (3.6b) requires communication between agents; transmitting all sensitivity information is unnecessary. Therefore, if a central coordinator is available—either a central entity or a local agent acting in that role—Step 3 can be executed efficiently.

Furthermore, when only neighbor-to-neighbor communication is feasible, [103] proposed bi-level distributed variants of ALADIN. These approaches utilize methods like the Schur complement (Section 2.3.3), decentralized ADMM, and decentralized conjugate gradient to solve (3.5) in a fully decentralized manner. Such adaptations extend the applicability of ALADIN to scenarios with communication constraints, making it suitable for a wide range of distributed optimization problems.

Step 4 updates the primal and dual iterates with running line-search routine [33, Alg. 3] or using full step size, i.e., $\alpha_i = 1, i = 1, 2, 3$. In many applications such as distributed AC-OPF, numerical results on large-scale problems illustrate that the full step size leads to good performance with a proper initial guess [34, 39, 36]. However, if the initial guess is not close to the local minimizer, employing a globalization strategy [33, Alg. 3] becomes necessary to guarantee convergence. This strategy will be introduced in the following section.

Remark 3.1. *In practice, if the quasi-Newton Hessian approximation such as Broyden–Fletcher–Goldfarb–Shanno is used to compute H_ℓ, the communication cost can be significantly reduced. For more details and numerical evaluations, we refer to [34].*

Remark 3.2. *Unlike in ADMM, the decoupled NLPs are not as critical for the convergence of ALADIN. ADMM-type algorithms must solve augmented Lagrangian subproblems with a penalty parameter ρ to determine the step length for updating dual variables during dual ascent. In contrast, ALADIN employs a Newton-type method, where convergence depends on the coupled QP step and can proceed even without solving the decoupled NLPs, making it more akin to a distributed SQP approach.*

3.2.2 Termination Condition

After the parallelizable steps, the termination condition is checked by the coordinator. The ALADIN algorithm will be terminated if the primal and dual conditions are satisfied, i.e., the primal and the dual residuals are smaller than the predefined tolerance ϵ

$$\|Ax - b\| \leq \epsilon \tag{3.9a}$$
$$\|\Sigma(x - z)\| \leq \epsilon \tag{3.9b}$$

where $\Sigma \in \mathbb{R}^{N^x \times N^x}$ is a block diagonal matrix consisting of scaling matrices Σ_ℓ for all $\ell \in \mathcal{R}$, i.e.,

$$\Sigma = \text{blkdiag}(\Sigma_\ell)$$

It indicates that the local solution x_ℓ satisfies the first-order optimality condition of the original Problem (3.1) under the user specified numerical accuracy $\mathcal{O}(\epsilon)$, i.e.,

$$\left\|\nabla \left\{f_\ell(x_\ell) + \kappa_\ell^\top h_\ell(x_\ell)\right\} + A_\ell^\top \lambda\right\| = \mathcal{O}(\epsilon). \tag{3.10}$$

Remark 3.3 ([104]). *In practice, the dual condition (3.9b) is sufficient to ensure the satisfying violation of the condition (3.10), when the predefined tolerance ϵ is small enough. This due to the fact that the local solution x_ℓ satisfy the first-order optimality condition of decoupled NLP (3.2)*

$$\nabla \left\{f_\ell(x_\ell) + \kappa_\ell^\top h_\ell(x_\ell)\right\} + A_\ell^\top \lambda + \rho\, \Sigma\, (x - z) = 0.$$

3.2.3 Globalization Strategy

As the standard globalization routine [46], the primary goal of Algorithm 4 is to prevent Algorithm 3 from being executed infinitely often. To achieve this, this algorithm utilizes the L1-penalty function $\Phi(x)$ as a merit function to assess the quality of a given iterate x

$$\Phi(x) = \sum_{\ell \in \mathcal{R}} f_\ell(x_\ell) + \bar{\lambda} \left\| \sum_{\ell \in \mathcal{R}} A_\ell x_\ell - b \right\|_1 + \bar{\kappa} \sum_{\substack{\ell \in \mathcal{R} \\ i \in \mathcal{C}_\ell}} \max\{0, [c_\ell(x_\ell)]_i\}, \tag{3.11}$$

where the positive penalty parameters $\bar{\lambda},\ \bar{\kappa}$ are assumed to be sufficiently large such that Φ is an exact penalty function for Problem (3.1).

Algorithm 4 Globalization Strategy for Standard ALADIN [33, Alg. 3]

Initialization: set $\alpha_1 = \alpha_2 = \alpha_3 = 1$.

a) If the iterates from Step 4 in Algorithm 3 satisfies

$$\Phi(z) - \Phi(x^+) \geq \eta \left(\sum_{\ell \in \mathcal{R}} \frac{\rho}{2} \|x_\ell - z_\ell\|^2_{\Sigma_\ell} + \bar{\lambda} \left\| \sum_{\ell \in \mathcal{R}} A_\ell x_\ell - b \right\|_1 \right) , \tag{3.12}$$

return $\alpha_1 = \alpha_2 = \alpha_3 = 1$.

b) If the full step is not accepted, set $x^+ = x$ and $\lambda^+ = \lambda$. If inequality (3.12) holds, return $\alpha_1 = 1$ and $\alpha_2 = \alpha_3 = 0$.

c) If both **a)** and **b)** failed, set $x^+ = z$ and choose $\alpha_3 \in (0, 1]$ by solving

$$\alpha_3^* = \arg \max_{\alpha_3 \in (0,1]} V_\rho(z, \lambda + \alpha_3(\lambda^{\text{QP}} - \lambda)) \tag{3.13}$$

with the objective function defined by a parametric optimization problem

$$V_\rho(\bar{x}, \lambda) = \min_x \quad \sum_{\ell \in \mathcal{R}} \left\{ f_\ell(x_\ell) + \lambda^\top A_\ell x_\ell + \frac{\rho}{2} \|x_\ell - \bar{x}_\ell\|^2_{\Sigma_\ell} \right\} - \lambda^\top b \, . \tag{3.14a}$$

$$\text{s.t.} \quad c_\ell(x_\ell) \leq 0, \tag{3.14b}$$

return $\alpha_1 = \alpha_2 = 0$ and $\alpha_3 = \alpha_3^*$.

The following consideration aims to show that Algorithm 4 enforces convergence after a finite number of steps:

$$\Phi(x) - \Phi(x^+) \geq \eta \left(\sum_{\ell \in \mathcal{R}} \frac{\rho}{2} \|x_\ell - z_\ell\|^2_{\Sigma_\ell} + \bar{\lambda} \left\| \sum_{\ell \in \mathcal{R}} A_\ell x_\ell - b \right\|_1 \right) ,$$

This follows the fact that the righthand side of (3.2.3) is lower bounded by

$$\eta \left(\sum_{\ell \in \mathcal{R}} \frac{\rho}{2} \|x_\ell - z_\ell\|^2_{\Sigma_\ell} + \bar{\lambda} \left\| \sum_{\ell \in \mathcal{R}} A_\ell x_\ell - b \right\|_1 \right) \geq \eta \min \left\{ \frac{\sigma \epsilon^2}{2\rho}, \bar{\lambda} \epsilon \right\} ,$$

with constants $0 < \eta \ll 1$ and $\sigma > 0$.

There are three main steps in Algorithm 4. Step **a)** returns full step size if the descent condition (3.12) holds. Otherwise, Step **b)** checks the decent condition by only updating the primal

iterates using the decoupled solution. If both Step **a)** and **b)** are failed, Step **c)** only updates the dual variables by approximating the solution to the dual problem

$$\lambda^+ = \max_{\lambda} V_\rho(z, \lambda).$$

This step provides a better estimate of the dual variables λ. In Section 3.3.2, we will provide a detailed analysis demonstrating why it achieves global convergence.

Remark 3.4. *The main disadvantage of Algorithm 4 lies in Step c), as its computational cost is comparable to solving the original Problem (3.1).*

3.3 Local & Global Convergence Analysis

This section summarizes global and local convergence analysis detailed in [33]

3.3.1 Local Convergence

Similar to Assumption 2.1 for SQP in Section 2.3.1, we consider the following assumption for the distributed problem (3.1).

Assumption 3.1. *The primal-dual point $(x^*, \lambda^*, \kappa^*)$ is a local solution of Problem (2.1) at which the following conditions holds*

(a) For each region $\ell \in \mathcal{R}$, the local objective f_ℓ and constraint c_ℓ are twice differentiable in a neighbourhood of x_ℓ^* with Lipchitz continuous second derivatives.

(b) The linear independence constraint qualification (Definition 2.3) holds at x^*.

(c) The strict complementarity condition (Definition 2.4) holds at $(x^*, \lambda^*, \kappa^*)$.

(d) The second order sufficient condition (Theorem 2.2) holds at $(x^*, \lambda^*, \kappa^*)$.

An additional assumption on the approximated Hessians is necessary when using Hessian approximation or regularization methods. This condition is slightly stronger than the compatibility condition in Assumption 2.2 but ensures a local quadratic convergence rate, as discussed in Remark 2.3.

Assumption 3.2. *Let the Hessian approximation H_ℓ be accurate enough so that*

$$H_\ell = \nabla^2 \left\{ f_\ell(x_\ell) + \kappa_\ell^\top c_\ell(x_\ell) \right\} + \mathcal{O}\left(\|x_\ell - z_\ell\| \right) \tag{3.15}$$

holds for all $\ell \in \mathcal{R}$.

Remark 3.5. *The term $\mathcal{O}(\|x_\ell - z_\ell\|)$ in (3.15) is introduced to represent some regularization term used for numerical robustness, e.g., inertia correction (Algorithm 1). Despite these heuristic tricks for regularization, the locally quadratic convergences can always be observed in verifying the condition (3.15) numerically.*

With these assumptions, we can state the local convergence properties of Algorithm 3.

Theorem 3.1 (Local convergence of ALADIN [36]). *Let $(z^*, \lambda^*, \kappa^*)$ be a KKT point for Problem (3.1) that satisfies Assumption 3.1. Suppose that for all $\ell \in \mathcal{R}$, the parameter ρ being sufficiently large so that:*

$$\nabla^2 \left\{ f_\ell(x_\ell) + \kappa_\ell^\top c_\ell(x_\ell) \right\} + \rho \Sigma_\ell \succ 0 \tag{3.16}$$

Additionally, assume that the Hessian approximations meet Assumption 3.2. The iterate (x, λ) given by Algorithm 3 converges locally to (x^*, λ^*) at a quadratic rate with a sufficient large δ.

Proof. The proof of Theorem 3.1 follows two steps. First, the local minimizer of subproblems (3.2) are closely related to augmented Lagrangian method (ALM). Similar to the analysis in [70, Prop 4.2.3] [46, Thm 17.6], x_ℓ is parametric with (z, λ) and the solution maps are Lipschitz continuous, i.e., there exist constants $\chi_1, \chi_2 > 0$ such that

$$\|x - z^*\| \leq \chi_1 \|z - z^*\| + \chi_2 \|\lambda - \lambda^*\|. \tag{3.17}$$

Then, (3.5) is equivalent to a step of the SQP method. From the local convergence analysis (Theorem 2.3), we have

$$\left\| \begin{bmatrix} x^+ - x^* \\ \lambda^+ - \lambda^* \end{bmatrix} \right\| \leq \left\| H - \nabla^2 \left\{ f(x) + \kappa^\top c(x) \right\} \right\| \cdot \mathcal{O}\left(\|x - x^*\| \right) + \mathcal{O}(\|x - x^*\|^2).$$

Since the Hessian approximation is accurate enough (Assumption (3.15)), the quadratic contractions

$$\left\| x^+ - x^* \right\| \leq \mathcal{O}(\|x - x^*\|^2), \tag{3.18a}$$
$$\left\| \lambda^+ - \lambda^* \right\| \leq \mathcal{O}(\|x - x^*\|^2), \tag{3.18b}$$

can be established. By combining (3.17) and (3.18), locally quadratic convergence of Algorithm 3 can be guaranteed. For more details refer to [104, Sec. 4.1] [33, Sec. 7]. □

Remark 3.6. *Note that this local convergence result relies solely on regularity conditions at the solution point and does not require any convexity assumptions.*

3.3.2 Global Convengence

In this section, we address the global convergence of the ALADIN algorithm, starting with assuming the following:

Assumption 3.3 (Duality Condition [33, Lem. 1]). *Let ρ be sufficiently large so that Problem (A.3) has no duality gap.*

This assumption leads to a crucial result, summarized in the following lemma, which is key to the global convergence of the algorithm:

Lemma 3.1 ([33]). *Suppose that the only dual variable is always updated according to Step **c)** in Algorithm 4 without updating primal variable z. Then the solutions of the decoupled NLP step (step 1 in Algorithm 3) converge to a limit point $x^\star$ that satisfies the strict descent direction (3.12)*

$$\Phi(z) - \Phi(x^\star) \geq \eta \left(\sum_{\ell \in \mathcal{R}} \frac{\rho}{2} \left\| x_\ell^\star - z_\ell \right\|_{\Sigma_\ell}^2 + \bar{\lambda} \left\| \sum_{\ell \in \mathcal{R}} A_\ell x_\ell^\star - b \right\|_1 \right)$$

with a fixed constant $0 < \eta \ll 1$

The proof of Lemma 3.1 is given in Appendix A.2 for the sake of completeness. We now proceed to the main theorem, which establishes the global convergence of the ALADIN algorithm.

Theorem 3.2 (Global convergence of ALADIN [33, Thm. 2]). *Assume that problem 3.1 is feasible and bounded from below, ensuring a minimum exists. If the conditions specified in Lemma 3.1 are satisfied, the parameter ρ is chosen sufficiently large, and the line-search parameters are adjusted according to Algorithm 4, then Algorithm 3 will terminate after a finite number of iterations.*

Proof. The proof is divided into two parts. First, when only the dual variables λ are updated, the dual variables and local solutions of the decoupled NLP subproblems (2.5) converge to a limit

point that satisfies the strict descent condition (3.12), as stated in Lemma 3.1. This implies that Step **c)** in Algorithm 4 is not executed infinitely.

Second, due to the strict descent condition (3.12), each time Step **a)** or Step **b)** is executed, the objective function $\Phi(x)$ decreases by a positive constant. Since Φ is bounded from below, it cannot decrease indefinitely. A detailed analysis is provided in [33, Sec. 6].

□

3.4 Limitations of the ALADIN Algorithm

Despite its advantages, the ALADIN algorithm faces limitations that affect its scalability and efficiency. One key challenge arises from handling inequality constraints in large-scale systems. While ALADIN works well for small to medium-sized problems with inequality constraints, it encounters difficulties, like active-set SQP discussed in Section 2.3.1, as the problem grows in size. Active set methods within ALADIN select a subset of constraints to treat as equalities, simplifying each iteration. However, the number of possible active sets can be extremely large, i.e., up to $2^{N^{\mathrm{I}}}$ for the NLP problem (3.1), where N^{I} is the number of inequality constraints. This estimate arises from the fact that each inequality constraint can either be included in the local active set $\mathcal{A}_\ell(x_\ell)$ in (3.3), or be left out. This exponential growth, known as the combinatorial difficulty of NLP [46, Ch. 15.2], is NP-hard and makes ALADIN impractical for large-scale problems.

Another significant issue with ALADIN is its globalization strategy (Algorithm 4). Although the globalization approach is theoretically sound, it is often too complex for practical use. When initial guesses, especially dual variables as discussed in the original ALADIN paper [33], are not close to the optimal solution, ALADIN may diverge or require excessive computation (step **c)** in Algorithm 4), which is often comparable to solving the original nonconvex problems (3.1) directly. This limitation reduces its practicality for solving large-scale optimization problems where good initial estimates are not always available.

Communication overhead is an additional challenge. ALADIN requires sensitivity information from each subproblem to construct a second-order approximated model of the original problems (3.1) in the coordinator. While this accelerates convergence compared with first-order algorithms like ADMM, it significantly increases computational and communication demands. As system size grows, evaluating the second-order derivatives and coordinating communication between local agents and a central coordinator become significant bottlenecks. Previous research has focused mainly on the numerical performance of the ALADIN algorithm, with limited exploration of distributed implementations. This leaves the impact of communication overhead on

practical performance unclear. Additionally, optimizing over networks, such as power systems, may reveal the network topology of the power networks through the sparsity pattern of local derivative information J_ℓ, raising privacy concerns.

4 Advancing ALADIN in Power System Applications

Solving NLP problems in a distributed manner enables grid operators to compute the overall grid state without sharing full grid models—this is a practical challenge to which the industry does not have off-the-shelf answers [6]. This chapter explores and advances the applications of the ALADIN algorithm in power systems, focusing on AC power flow (PF) and AC optimal power flow (OPF) problems.

The first idea is to introduce a Gauss-Newton ALADIN algorithm for AC PF problems, achieving faster convergence through problem reformulation and demonstrating high scalability for NLPs in the absence of inequality constraints. The second idea is to improve the globalization framework by proposing a robust strategy to enhance convergence from wider initial conditions, improving global reliability without compromising local efficiency. The third demonstrates the deployment of ALADIN in distributed computing environments for geographically distributed AC OPF problems, validating its practicality with a balance between performance and data privacy.

Outline. Section 4.1 briefly introduces the three research aspects in power system applications. Section 4.2 presents a distributed reformulation of the optimization problem for power systems. Section 4.3 introduces the Gauss-Newton method to approximate AC power flow (PF) solutions for large-scale power systems. Section 4.4 proposes an improved globalization strategy. Section 4.5 examines the deployment of the distributed optimization algorithm for AC optimal power flow (OPF) problems in distributed systems, supported by a case study in a geographically distributed computing environment. Section 4.3 is based on [39], Section 4.4 on [37], and Section 4.5 on [40].

4.1 Introduction

The increasing complexity of power systems, driven by renewable energy integration, challenges traditional centralized methods like Gauss-Seidel and Newton-Raphson [105], which struggle with scalability, privacy concerns, and single points of failure [4]. Distributed optimization

algorithms, such as alternating direction method of multipliers (ADMM), address these issues by solving subproblems locally and coordinating global updates, preserving data privacy [106, 29]. While effective for convex problems, ADMM often fails to converge for nonconvex problems based on AC models due to sensitivity to initial conditions and tuning parameters [34, 6].

In contrast, the ALADIN algorithm is tailored for nonconvex problems, offering local convergence guarantees and leveraging second-order information for faster, more reliable results [6]. However, improvements in both its computational efficiency and globalization strategy are necessary to enhance its applicability to large-scale power systems. These improvements should allow the algorithm to maintain its local convergence properties while being able to handle the complexity and size of real-world energy systems more effectively.

The three papers [39, 37, 40] attempt to handle the limitations of the ALADIN algorithm and expand its applicability to power system optimization. The first paper [39] introduces a Gauss-Newton-based ALADIN algorithm designed specifically for solving AC PF problems. This approach leverages a reformulation of the AC PF problem as a zero-residual least-squares problem, where the optimal dual variables are known to be zero. Without evaluating second-order derivatives, Gauss-Newton ALADIN has lower computation and communication overhead, making it significantly more efficient for large-scale systems. However, while this method improves convergence speed for AC PF problems, ALADIN still faces challenges in terms of scalability when applied to generic NLPs.

The second paper [37] addresses ALADIN's globalization issues. It proposes a novel globalization strategy that ensures the algorithm can converge from a wider range of initial conditions, making it more robust for real-world applications. This enhanced global ALADIN variant maintains the potential for local quadratic convergence while offering more reliable global performance. The new strategy balances global reliability with the algorithm's existing strengths in local efficiency, extending ALADIN's utility for large-scale, nonconvex optimization problems.

The third paper [40] shifts focus to deploying ALADIN in distributed computing environments, particularly for solving AC-OPF problems in geographically distributed power systems. This involves deploying the ALADIN algorithm within the *e*nergy system *Co-Sim*ulation (eCoSim) module in the eASiMOV framework [107]. Comprehensive evaluations across multiple system scenarios reveal a marginal performance slowdown compared to the centralized approach and the distributed approach executed on single machines—a justified trade-off for enhanced data privacy. This investigation serves as empirical validation of the successful execution of distributed AC OPF within a geographically distributed environment, highlighting potential directions for future research.

Together, these three papers represent significant advancements in the development and application of the ALADIN algorithm for power system optimization. They address the key challenges of scalability, global convergence, and practical deployment in distributed environments, making ALADIN a more viable tool for solving complex, real-world energy system problems.

4.2 Distributed Reformulation with Component Sharing

Two essential steady-state problems in power system operation are introduced in Appendix B: the AC power flow (PF) problem, which determines all electrical quantities at steady state by solving nonlinear systems (B.22), and the AC optimal power flow (OPF) problem, which can be represented as nonlinear program (NLP) (B.28), minimizing generation costs while ensuring system safety. In this chapter, we reformulate these problems for distributed approaches by decentralizing the nonlinear power flow equations.

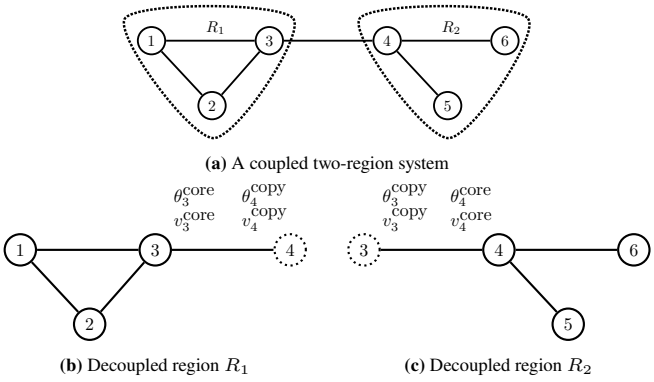
This approach involves sharing components between neighboring regions to ensure consistency, following [6, 39]. Consider a power system $\mathcal{S} = (\mathcal{N},\ \mathcal{L},\ \mathcal{R})$, where $\mathcal{N}$ represents the set of buses, $\mathcal{L}$ the set of branches and $\mathcal{R}$ the set of regions into which the network is partitioned. Let n^{reg}, n^{bus}, and n^{line} denote the number of regions, buses, and branches, respectively. In a specific region $\ell \in \mathcal{R}$, $\mathcal{N}_\ell^{\text{core}}$ denotes the set of core buses that are entirely local, $\mathcal{N}_\ell^{\text{copy}}$ denotes the set of copy buses shared by neighboring regions, and thus the set of all buses in the region ℓ can be represented as $\mathcal{N}_\ell = \mathcal{N}_\ell^{\text{core}} \cup \mathcal{N}_\ell^{\text{copy}}$. Moreover, let $\mathcal{L}_\ell$ denote the set of all regional branches.

To illustrate the distributed formulation, consider a 6-bus system partitioned into two regions, R_1 and R_2, as illustrated in Figure 4.1. To establish a self-contained AC OPF subproblem for the region R_1, the nodal power balance at the core buses $\{1, 2, 3\}$ must be included as local constraints. Additionally, the complex voltage of the copy bus $\{4\}$, shared by the neighboring region R_2, is required for the nodal balance at the core bus 3. Similarly, an AC optimal power flow can be established for the region R_2. Finally, an additional affine consensus constraint must be added to ensure physical consistency between core and copy buses:

$$v_3^{\text{copy}} = v_3^{\text{core}}, v_4^{\text{copy}} = v_4^{\text{core}}, \theta_3^{\text{copy}} = \theta_3^{\text{core}}, \theta_4^{\text{copy}} = \theta_4^{\text{core}} \tag{4.1}$$

Note that this example uses voltage in polar coordinates as defined in (B.1).

Remark 4.1 (Physical consistency [6]). *The concept of* core buses *and* copy buses *allows to compose the distributed power flow problem in a physically consistent manner: no additional modeling assumptions are introduced or required. If the correct solution to the distributed problem*



(a) A coupled two-region system

(b) Decoupled region R_1

(c) Decoupled region R_2

Figure 4.1: Decomposition by sharing components between neighboring regions.

is found, then this will also be the solution to the respective centralized power flow problem. In other words, the concept of copy buses *and* core buses does not *introduce a structural numerical error [108]. Other approaches, such as "cutting" connecting tie lines and enforcing equality of the electrical state at the intersection [109], are in general* not *physically consistent (only in the absence of line capacitance). Hence, even if the true solution to the distributed problem is found, this solution is not numerically identical to the solution of the centralized power flow problem. In other words, the concept of* cutting lines does *introduce a structural numerical error, generally speaking.*

Remark 4.2 (Privacy [6]). *To formulate the power flow equations for region i, the voltage information of the copy buses needs to be shared among neighboring regions; this is inherent to the idea of* core *and* copy *buses. Although this means having to share data, the copy bus voltage data (i) does not contain a wealth of privacy information yet (ii) allows for a physically consistent problem formulation, see Remark 4.1.*

By this approach, AC PF Problem (B.22) can be reformulated as

$$c_\ell(x_\ell) = 0, \ \forall \ell \in \mathcal{R} \tag{4.2a}$$

$$\sum_{\ell \in \mathcal{R}} A_\ell x_\ell = b. \tag{4.2b}$$

and AC OPF Problem (B.28) can be reformulated as

$$\min_{x} \quad f(x) := \sum_{\ell \in \mathcal{R}} f_\ell(x_\ell) \tag{4.3a}$$

$$\text{s.t.} \quad c_\ell(x_\ell) \leq 0, \quad \forall \ell \in \mathcal{R} \tag{4.3b}$$

$$\sum_{\ell \in \mathcal{R}} A_\ell x_\ell = b \tag{4.3c}$$

where local state x_ℓ includes the local state variables in the region $\ell \in \mathcal{R}$. f_ℓ denotes the local cost function with respect to core generators in the region $\ell \in \mathcal{R}$, while c_ℓ collects the local constraints in the region $\ell \in \mathcal{R}$. The consensus constraint (4.3c) ensures consistency of core and copy variables between neighboring regions.

Remark 4.3. *In this chapter, the distributed problems (4.2) (4.3) are initialized with a flat start, where the voltage angles and magnitudes are set to zero and 1.0 p.u. respectively [110]. For this initialization strategy, it is demonstrated numerically that it can provide a good initial guess for distributed AC PF [6, 39] and distributed AC OPF [34, 101, 32].*

4.3 Rapid Distributed AC Power Flow

4.3.1 Distributed Least-Squares Formulation

The distributed nonlinear systems (4.2) can be rewritten in a standard least-squares formulation with affine consensus constraint

$$\min_{x} \quad f(x) := \sum_{\ell \in \mathcal{R}} f_\ell(x_\ell) = \frac{1}{2} \sum_{\ell \in \mathcal{R}} \|c_\ell(x_\ell)\|_2^2 \tag{4.4a}$$

$$\text{s.t.} \quad Ax = b \quad | \ \lambda \tag{4.4b}$$

with the consensus matrix $A = (A_1, A_2, \cdots, A_{N^{\text{reg}}})$ and the state $x = (x_1^\top, x_2^\top, \cdots, x_{N^{\text{reg}}}^\top)^\top$. The problem (4.4) can be classified as a zero-residual least-squares problem, since all the power flow residuals are equal to zero at the PF solution x^*.

Proposition 4.1. *Let the power flow Problem (B.22) be feasible, i.e., a primal solution x^* to Problem (4.4) exists such that the power flow residual $c_\ell(x_\ell^*) = 0$ for all $\ell \in \mathcal{R}$ bounded by consensus constraint (4.4b), and let linear independence constraint qualification (LICQ) hold at x^*. Then the dual variable $\lambda^* = 0$ with the primal solution x^* satisfies the KKT conditions, i.e., $(x^*, \lambda^* = 0)$ is a KKT point.*

The derivatives of the objective $f_\ell(x_\ell)$ can be expressed as

$$\nabla f_\ell(x_\ell) = J_\ell(x_\ell)^\top c_\ell(x_\ell) \tag{4.5a}$$
$$\nabla^2 f_\ell(x_\ell) = J_\ell(x_\ell)^\top J_\ell(x_\ell) + Q_\ell(x_\ell) \tag{4.5b}$$

with

$$J_\ell(x_\ell) = \left[\nabla c_{\ell,1}, \nabla c_{\ell,2}, \cdots, \nabla c_{\ell,N^{\text{pf}}}\right]^\top \tag{4.6a}$$
$$Q_\ell(x_\ell) = \sum_{m=1}^{N^{\text{pf}}} c_{\ell,m}(x_\ell)\nabla^2 c_{\ell,m}(x_\ell). \tag{4.6b}$$

The Gauss-Newton approximation can be written as

$$\nabla^2 f_\ell(x_\ell) \approx J_\ell(x_\ell)^\top J_\ell(x_\ell) \tag{4.7}$$

In practice, the first term (4.6a) dominates the second one (4.6b), either because the residuals $c_{\ell,m}$ are close to affine near the solution, i.e., $\nabla^2 c_{\ell,m}$ are relatively small, or because of small residuals, i.e.,$c_{\ell,m}$ are relatively small [46].

Remark 4.4. *For solving the zero-residual least-squares problem, the Gauss-Newton approximation is exact at the solution, and can converge to the exact Hessian during iterations rapidly.*

4.3.2 Gauss-Newton ALADIN Framework

This section presents the standard ALADIN algorithm and its new variant for zero-residual least-squares problems.

Standard ALADIN without Local Constraints

ALADIN for Problem (4.4) is outlined in Algorithm 5. The algorithm has two main steps, i.e., a decoupled step 1 and a consensus step 3. Pursuing the idea of augmented Lagrangian, the local problem is formulated as (4.9) in step 1, where ρ is the penalty parameter and Σ_ℓ is the positive definite scaling matrix for the region ℓ. Based on the result from local NLPs (4.9), the ALADIN algorithm terminates if both the primal and the dual residuals are smaller than tolerance ϵ

$$\left\| \sum_{\ell \in \mathcal{R}} A_\ell x_\ell - b \right\|_\infty \leq \epsilon \text{ and } \max_\ell \left\| \Sigma_\ell (x_\ell - z_\ell) \right\|_\infty \leq \epsilon \tag{4.8}$$

Compared with a simple averaging step of ADMM in the coordinator, ALADIN builds a coupled quadratic program (QP) (4.11) based on curvature information (4.10) to coordinate the results of the decoupled step from all regions. Additionally, a slack variable s and a corresponding penalty parameter μ are introduced in the consensus step to ensure the feasibility of the coupled QP. Consequently, ALADIN achieves fast and guaranteed convergence. The local convergence analysis mirrors Theorem 3.1 when there are no local constraints. For a more detailed analysis, we refer to [33].

Algorithm 5 Standard ALADIN (Unconstrained)

Initialization: $\lambda, \rho, \mu, z_\ell, \Sigma_\ell \succ 0$ **for all** $\ell \in \mathcal{R}$,

Repeat:

1: Solve decoupled NLPs

$$\min_{x_\ell} \; f_\ell(x_\ell) + \lambda^\top A_\ell x_\ell + \frac{\rho}{2} \|x_\ell - z_\ell\|^2_{\Sigma_\ell} \tag{4.9}$$

and compute local sensitivities for all $\ell \in \mathcal{R}$

$$g_\ell = \nabla f_\ell(x_\ell) \text{ and } H_\ell \approx \nabla^2 f_\ell(x_\ell) \tag{4.10}$$

2: Check termination condition (4.8)

3: Solve coupled QP

$$\min_{\Delta x, s} \; \frac{1}{2}\Delta x^\top H \Delta x + g^\top \Delta x + \lambda^\top s + \frac{\mu}{2}\|s\|_2^2 \tag{4.11a}$$

$$\text{s.t.} \; A(x + \Delta x) = b + s \tag{4.11b}$$

where Hessian $H = \text{diag}\{H_\ell\}_{\ell \in \mathcal{R}}$ and gradient g with components g_ℓ

4: Update primal and dual variables with full-step

$$z^+ = x + \Delta x, \tag{4.12a}$$

$$\lambda^+ = \lambda^{QP}. \tag{4.12b}$$

Gauss-Newton ALADIN

Based on the framework of standard ALADIN, we propose a tailored version specific for solving zeros-residual least-squares problem in this chapter, see Algorithm 6. Since optimal values of Lagrange multipliers are equal to zero $\lambda^* = 0$ according to *Proposition 4.1*, the terms $\lambda^\top A_\ell x_\ell$ and $\lambda^\top s$ can be omitted by fixing dual iterates $\lambda = 0$ at the cost of convergence rate. In this way, both coupled and decoupled steps can be viewed as adding a residual to the original problems respectively, and can be solved by equivalent linear systems efficiently.

Algorithm 6 Gauss-Newton ALADIN [39]

Initialization: λ, ρ, μ, z_ℓ, $\Sigma_\ell \succ 0$ **for all** $\ell \in \mathcal{R}$,

Repeat:

1: Solve decoupled linear systems and update primal variables x_ℓ

$$\left(J_\ell^{z\top} J_\ell^z + \rho I\right) p_\ell = -J_\ell^{z\top} c_\ell^z \tag{4.13}$$

with Gauss-Newton step $p_\ell = x_\ell - z_\ell$, as well as compute local sensitivities for all $\ell \in \mathcal{R}$

$$g_\ell = J_\ell(\hat{x}_\ell)^\top c_\ell(\hat{x}_\ell) \text{ and } H_\ell = J_\ell(\hat{x}_\ell)^\top J_\ell(\hat{x}_\ell) \tag{4.14}$$

2: Check termination condition (4.8)

3: Solve the linear system of coupled QP

$$\left(H + \mu A^\top A\right) \Delta x = -\mu A^\top \left(A\hat{x} - b\right) - g \tag{4.15}$$

where Hessian $H = \text{diag}\{H_\ell\}_{\ell \in \mathcal{R}}$ and gradient g with components g_ℓ

4: Update primal variables with full step

$$z^+ = \hat{x} + \Delta x. \tag{4.16}$$

For the decoupled step 1, the objective function (4.9) can be approximated by a quadratic model by applying the Gauss-Newton method

$$M_\ell(p_\ell) = \frac{1}{2} p_\ell^\top \left(J_\ell^{z\top} J_\ell^z + \rho I\right) p_\ell + J_\ell^{z\top} c_\ell^z p_\ell + f_\ell(z_\ell) \tag{4.17}$$

with Gauss-Newton step $p_\ell = x_\ell - z_\ell$, Jacobian matrix $J_\ell^z = J_\ell(z_\ell)$ and residual vector $c_\ell^z = c_\ell(z_\ell)$ at the initial point z_ℓ in every iterate. Accordingly, the decoupled NLP (4.9) is solved by a linear system (4.13), where x_ℓ is an inexact solution to this problem.

For the coupled step 3, the objective function can be rewritten as

$$\min_{\Delta x} \ \frac{1}{2} \Delta x^\top H \Delta x + g^\top \Delta x + \frac{\mu}{2} \left\| A\left(\hat{x} + \Delta x\right) - b \right\|_2^2 \tag{4.18}$$

In the corresponding linear system (4.15), Δx in coupled step 3 is locally equivalent to a standard Gauss-Newton step of the original coupled Problem (4.4), where the slack variable $s = A\left(\hat{x} + \Delta x\right) - b$ can be viewed as an additional weighted residual.

4.3.3 Local Convergence Analysis

Since a flat start can provide a good initial guess (Remark 4.3), we focus on the local convergence in this section. To study local convergence, we assume that the starting point and the iterates

are all located in a small neighborhood of the optimizer, within which the solution has physical meaning. The convex set Ω contains all the points in the bounded neighborhood. Besides, the power flow equation c of the original coupled Problem (4.4) is continuously differentiable, and $\left\|\nabla^2 f(x)\right\|$ is bounded for all $x \in \Omega$. Then, according to Lipschitz theorem (Definition A.1), there exists a constant $L > 0$

$$\|\nabla f(x) - \nabla f(z^*)\| = \left\|\nabla^2 f(\tilde{x})\right\| \|x - z^*\| \leq L \|x - z^*\| \tag{4.19}$$

with $\tilde{x} = x - t(x - z^*) \in \Omega$ for some $t \in (0, 1)$. Hence, the function $f(x) = \frac{1}{2} \|c(x)\|_2^2$ is twice Lipschitz-continuously differentiable in the neighborhood Ω.

Remark 4.5. *In practice, the constraints in the AC PF problems satisfies linear independence constraint qualification (LICQ). Consequently, second order sufficient condition (SOSC) of Problem (4.4) is fulfilled, ensuring that $(x^* = z^*, \lambda^* = 0)$ is a regular KKT point (Definition 2.5).*

Next, let's turn to the local convergence property of Algorithm 6. For the analysis of local decoupled step 1, we introduce $\bar{x}$ as the exact solution and $\hat{x}$ as the inexact solution of the decoupled NLPs (4.9), whereas $x^* = z^*$ is the primal optimizer of the original coupled Problem (4.4).

Theorem 4.1. *Let the minimizer $(x^* = z^*, \lambda^* = 0)$ be a regular KKT point of Problem (4.4), let the initial guess located in the small neighborhood of the optimizer Ω, and let μ sufficient large such that $\frac{1}{\mu} \leq O(\|\hat{x} - z^*\|)$, then the iterates $\hat{x}$ of Algorithm 6 converge quadratically to a local solution.*

Proof. Proof of *Theorem 4.1* can be established by three steps, following the analysis by [34]. First, because the local inexact solution $\hat{x}_\ell$ is obtained by the Gauss-Newton method, the $\hat{x}$ is a linear contraction to the exact solution $\bar{x}$, i.e., there exists a constant $\eta_1 > 0$ such that

$$\|\hat{x} - \bar{x}\| \leq \eta_1 \|z - \bar{x}\| . \tag{4.20}$$

Second, analogous to (3.17), we have

$$\|\bar{x} - z^*\| \leq \eta_2 \|z - z^*\| , \ \exists \eta_2 > 0 \tag{4.21}$$

This differs from standard ALADIN by a fixed dual variable $\lambda = 0$.

Third, because the coupled step of Algorithm 6 is a standard Gauss-Newton step of the original coupled Problem (4.4), as well as the Lipschitz continuity of f and sufficient large μ such that

$\frac{1}{\mu} \leq O(\|\hat{x} - z^*\|)$, we obtain the following inequality according to the convergence analysis of the standard Gauss-Newton method [46, Section 10.3]

$$\left\|z^+ - z^*\right\| \leq \left\|H(z^*)^{-1}Q(z^*)\right\| \|x - z^*\| + O(\|x - z^*\|^2) \tag{4.22}$$

with $Q = \text{diag}\{Q_\ell\}_{\ell \in \mathcal{R}}$. For Problem (4.4), all the optimal residuals are equal to zero, then we have $Q_\ell(z_\ell^*) = 0$ for all $\ell \in \mathcal{R}$. As a result,

$$\left\|z^+ - z^*\right\| \leq O(\|\hat{x} - z^*\|^2) \tag{4.23}$$

The statement of *Theorem 4.1* follows from the combining of(4.20), (4.21) and (4.23). □

4.3.4 Case Study

In this section, we illustrate the performance of the two variants of ALADIN algorithm. We use the original distributed power flow model with standard ALADIN (Algorithm 5) as a benchmark. Towards practical implementation, several test cases by [6] are also modified—multiple connecting tie lines are added, and the graph of regions is transferred from radial to meshed topology. Besides, we introduce a 10224-bus test case to exhibit the performance for a large-scale implementation.

The framework is built on MATLAB-R2021a and the case studies are carried out on a standard desktop computer with `Intel® i5-6600K CPU @ 3.50GHz` and 16.0 GB installed ram. To run the benchmark, adding Matpower to MATLAB search path is necessary. The CasADi toolbox [111] is used in MATLAB, and IPOPT [71] is used as the solver for decoupled NLPs. To solve the linear system, a conjugate-gradient technique [46, Algorithm 7.2] is implemented in order to avoid matrix-matrix multiplications, i.e., $J^\top J$.

Following [34], the quantities in the following are used to illustrate the convergence behavior

1 The deviation of optimization variables from the optimal value $\|x - x^*\|_\infty$.

2 The primal residual, i.e., the violation of consensus constraint $\|Ax - b\|_\infty = \left\|\sum_{\ell \in \mathcal{R}} A_\ell x_\ell - b\right\|_\infty$.

3 The dual residual $\gamma = \max_{\ell \in \mathcal{R}} \|\Sigma_\ell(x_\ell - z_\ell)\|_\infty$.

4 The solution gap calculated as $|f(x) - f(x^*)|$, where $f(x^*)$ is provided by the centralized approach.

The user-defined tolerance ϵ is set to 10^{-8}, while the penalty parameters ρ and μ are set to 10^2.

Buses	N^{reg}	N^{conn}	N^x	Computing Time [s]		
				Algorithm 5	Algorithm 6	Matpower (centralized)
53	3	5	126	0.114	0.011	0.004
418	2	8	868	0.315	0.028	0.014
2708	2	30	5536	2.149	0.109	0.051
4662	5	130	9844	5.694	0.228	0.129
10224	13	242	21416	14.392	0.591	0.257

Table 4.1: Computing time for solving power flow problem

For fair comparison, the primal variables x are initialized with the initial guess provided by Matpower [112], while the dual variable λ is set to zero. RUNPF from Matpower is used to represent a centralized approach.

Table 4.1 displays the computing time of different approaches. What stands out in this table is the fast computing time of the Gauss-Newton ALADIN (Algorithm 6). In contrast to solving NLP in a decoupled step of Algorithm 5, Algorithm 6 solves the equivalent linear systems of a quadratic approximation in both decoupled and coupled steps by exploiting the structure of the problem formulation. Consequently, the computation effort has been reduced dramatically. As a result, the computing time of solving the distributed PF model by using Algorithm 6 is in the same order of magnitude as the centralized approach and can be further improved by implementing parallel computing.

Note that all the test cases can converge within half a dozen iterations, whatever model or algorithm is applied. In summary, the dimensional reduction and the inexact approach has little impact on convergence rate, but can reduce the computational effort remarkably.

Next, we study the convergence behavior of the largest test case, i.e., 10224-bus system. The test case is composed of six 1354-bus Matpower test cases, and seven 300-bus Matpower test cases. Thereby, there are 13 regions and 242 connecting-tie lines between neighboring regions, as presented in Table 4.1. Its connection graph of regions are shown in Fig. 4.2. To solve the AC

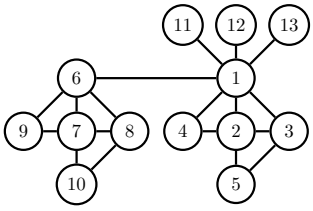


Figure 4.2: Connection graph of 10224-bus test case

	θ [rad]	v [p.u.]	p [p.u.]	q [p.u.]
$\|\cdot\|_\infty$	1.7×10^{-8}	7.5×10^{-9}	5.7×10^{-7}	3.2×10^{-6}

Table 4.2: The deviation of the 10224-bus system from the optimizer by the Gauss-Newton ALADIN

PF problem of the 10224-Bus system, we use the Gauss-Newton ALADIN algorithm. Fig. 4.3 shows the four quantities in every iterate, i.e., the deviation of current variables from the optimal value, the primal residual, the dual residual and the solution gap. Within half a dozen iterates, the new ALADIN algorithm converges to the optimal solution with high accuracy, as presented in Table 4.2. At the same time, a locally quadratic convergence rate can be observed from Fig. 4.3.

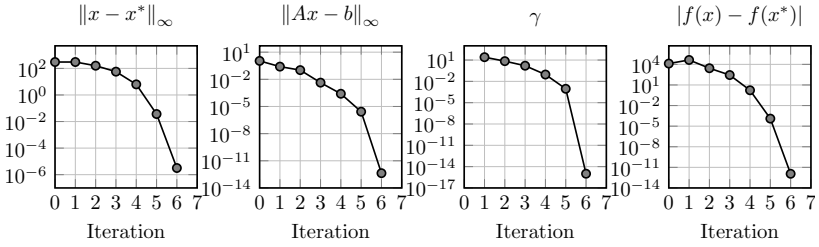


Figure 4.3: Convergence behavior of 10224-bus system using the Gauss-Newton ALADIN

4.4 ALADIN with an Enhanced Globalization Strategy

In the previous section, we demonstrated the fast local convergence of ALADIN-type algorithms. However, executing the linesearch step (3.13) becomes problematic, as discussed in Remark 3.4, when the initial guess is not sufficiently accurate, as highlighted in [33].

In this section, we shift our focus from local convergence performance to developing a globalization strategy for ALADIN-type algorithm. Similar to Section 4.3.1, we continue the discussion for NLP problems without local constraints:

$$\min_x \quad f(x) := \sum_{\ell \in \mathcal{R}} f_\ell(x_\ell) \tag{4.24a}$$

$$\text{s.t.} \quad Ax = b \quad | \ \lambda \tag{4.24b}$$

For a detailed adaptation of the proposed method to NLP with local constraints, we refer to [37, Sec. 4].

4.4.1 Algorithm Framework

Based on the ALADIN framework, we propose a novel variant of ALADIN that guarantees global convergence for Problem (4.24). The key steps of the proposed method are outlined in Algorithm 7. Unlike the standard ALADIN approach for solving Problem (4.24), our method introduces two primal iterates, x and z, in Algorithm 7, and incorporates a new computationally efficient globalization strategy to update both primal and dual iterates. Furthermore, a rigorous proof of global convergence is provided for the proposed Algorithm 7.

Step 1 solves decoupled NLPs for all $\ell \in \mathcal{R}$,

$$\min_{x_\ell} \; f_\ell(x_\ell) + \lambda^\top A_\ell x_\ell + \frac{\rho}{2} \|x_\ell - z_\ell\|^2_{\Sigma_\ell} + \frac{1}{2} \|x_\ell - \chi_\ell\|^2_{H_\ell} ,$$

where $\frac{\rho}{2} \cdot \|x_\ell - z_\ell\|^2_{\Sigma_\ell}$ and $\frac{1}{2} \cdot \|x_\ell - \chi_\ell\|^2_{H_\ell}$ introduce two augmented Lagrangian terms defined for primal iterate x and z, respectively.

Similar to the condition (3.16), we consider the scaling parameter $\rho > 0$ is sufficiently large:

Assumption 4.1. *For any given z_ℓ and scaling matrices $\Sigma_\ell \succ 0$, there exists a positive finite scalar $\rho \in [0, \infty)$ such that the functions*

$$F_\ell(\xi) := f_\ell(\xi) + \frac{\rho}{2} \|\xi - z_\ell\|^2_{\Sigma_\ell} \tag{4.31}$$

are strongly convex for all $\ell \in \mathcal{R}$.

Remark 4.6. *Matrices Σ_ℓ can be adjusted during the iterations. For example, one can choose $\Sigma_\ell = H_\ell$ in the initialization and update it when H_ℓ is updated.*

If Assumption 4.1 holds, the decoupled NLPs (4.25) are strongly convex such that they have unique solutions. Later on, we will recall this assumption in the convergence analysis. Once the decoupled problem is solved, Step 3 computes gradient

$$g_\ell = \rho \Sigma_\ell (z_\ell - x_\ell) + H_\ell (\chi_\ell - x_\ell) - A_\ell^\top \lambda.$$

This is based on the first-order optimality condition of the decoupled NLPs (4.25):

$$\nabla f_\ell(x_\ell) + A_\ell^\top \lambda + \rho \Sigma_\ell (x_\ell - z_\ell) + H_\ell (x_\ell - \chi_\ell) = 0$$

and optimally update the scaling matrices H_ℓ. Here, we do not update H_ℓ at every iteration to guarantee the global convergence, while locally, to improve the convergence speed, H_ℓ needs

Algorithm 7 Global ALADIN for Distributed Nonconvex Optimization [37]

Initialization:

- Choose initial guess $\chi_\ell \in \mathbb{R}^{n_i}$, $i = 1, ..., N$ and $\lambda \in \mathbb{R}^m$, a termination tolerance $\epsilon > 0$.
- Project the initial guess such that $\sum_{\ell \in \mathcal{R}} A_\ell \chi_\ell = b$ and set $z \leftarrow x$.
- Initialize scaling matrices $\Sigma_\ell, H_\ell \in \mathbb{S}^{n_i}_{++}$, $i = 1, ..., N$, and choose a sufficient large $\rho \geq 0$ satisfying Assumption 4.1.
- Set $\Phi(z) = \sum_{\ell \in \mathcal{R}} f_\ell(z_\ell)$ and choose a constant $0 < \gamma \ll 1$.

Repeat:

1 Solve decoupled NLPs

$$\min_{x_\ell} \; f_\ell(x_\ell) + \lambda^\top A_\ell x_\ell + \frac{\rho}{2} \|x_\ell - z_\ell\|^2_{\Sigma_\ell} + \frac{1}{2} \|x_\ell - \chi_\ell\|^2_{H_\ell} \tag{4.25}$$

for all $i = 1, \ldots, N$ in parallel.

2 If $\rho \|\Sigma_\ell(x_\ell - z_\ell)\|_1 \leq \epsilon$ and $\|H_\ell(x_\ell - \chi_\ell)\|_1 \leq \epsilon$ holds for all $i = 1, \ldots, N$, terminate with returning $x^* = z$ as a numerical solution.

3 Evaluate for all $i = 1, ..., N$,

$$g_\ell = \rho \Sigma_\ell(z_\ell - x_\ell) + H_\ell(\chi_\ell - x_\ell) - A_\ell^\top \lambda \,, \tag{4.26}$$

and optionally update $H_\ell \succ 0$.

4 Solve coupled equality constrained QP

$$x^{\text{QP}} = \arg\min_x \quad \sum_{\ell \in \mathcal{R}} \left\{ \frac{1}{2} (\chi_\ell - x_\ell)^\top H_\ell (\chi_\ell - x_\ell) + g_\ell^\top \chi_\ell \right\} \tag{4.27a}$$

$$\text{s.t.} \quad \sum_{\ell \in \mathcal{R}} A_\ell \chi_\ell = b \;\mid\; \lambda^{\text{QP}}. \tag{4.27b}$$

5 Update:

a) $x^+ \leftarrow x^{\text{QP}}$, $z^+ \leftarrow x^{\text{QP}}$, and $\lambda^+ \leftarrow \lambda^{\text{QP}}$, if the following inequality holds,

$$\Phi(z) - \Phi(x^{\text{QP}}) \geq \gamma \left(\sum_{\ell \in \mathcal{R}} \frac{\rho}{2} \|x_\ell - z_\ell\|^2_{\Sigma_\ell} + \frac{1}{2} \|x_\ell - \chi_\ell\|^2_{H_\ell} \right) . \tag{4.28}$$

b) Otherwise, set

$$\lambda^+ \leftarrow \lambda^{\text{QP}} + \rho M^\dagger \left(\sum_{\ell \in \mathcal{R}} A_\ell H_\ell^{-1} \Sigma_\ell (z_\ell - x_\ell) \right) , \tag{4.29a}$$

$$\chi_\ell^+ \leftarrow \chi_\ell^{\text{QP}} + \rho H_\ell^{-1} \Sigma_\ell (z_\ell - x_\ell) - H_\ell^{-1} A_\ell^\top (\lambda^+ - \lambda^{\text{QP}}). \tag{4.29b}$$

i) And set $z^+ \leftarrow x^+$ if the following inequality holds,

$$\Phi(z) - \Phi(x^+) \geq \gamma \left(\sum_{\ell \in \mathcal{R}} \frac{\rho}{2} \|x_\ell - z_\ell\|^2_{\Sigma_\ell} + \frac{1}{2} \|x_\ell - \chi_\ell\|^2_{H_\ell} \right) . \tag{4.30}$$

ii) Otherwise, set $z^+ \leftarrow z$.

Then, continue with Step 1.

to be evaluated based on the second-order derivative of f_ℓ under some regularity conditions. More details will be discussed in the following section. Step 4 solves a consensus QP like in Algorithm 5. Then, if the value of merit function

$$\Phi(x) = \sum_{\ell \in \mathcal{R}} f_\ell(x_\ell) + \bar{\lambda} \left\| \sum_{\ell \in \mathcal{R}} A_\ell x_\ell - b \right\|_1$$

satisfies the strict descent condition (4.28), Step 5a updates both the primal iterates

$$z^+ = x^+ = x + \alpha_1(y - x) + \alpha_2(x^{\mathrm{QP}} - y) \,,$$

and the dual iterates

$$\lambda^+ = \lambda + \alpha_3(\lambda^{\mathrm{QP}} - \lambda)$$

with full-step, i.e., $\alpha_i = 1$, $i = 1, 2, 3$. Otherwise, we first update x and λ by (4.29). Then, we re-check the descent condition (4.30) for x^+, and if it holds, we update z in Step 5b-i).

Remark 4.7. *The pseudo-inverse $M^\dagger$ has to be used as the matrix $A = [A_1, ..., A_N]$ might not be full row rank, i.e., LICQ does not hold for* (4.24).

Before further discussion on the update (4.29), we introduce the auxiliary problem

$$\xi^\star(z) := \arg\min_{\xi} \sum_{\ell \in \mathcal{R}} F_\ell(\xi_\ell) \overset{(4.31)}{=} f_\ell(\xi_\ell) + \frac{\rho}{2} \|\xi_\ell - z_\ell\|^2_{\Sigma_\ell} \tag{4.32a}$$

$$\text{s.t.} \sum_{\ell \in \mathcal{R}} A_\ell \xi_\ell = b \,, \tag{4.32b}$$

where $\xi^\star(\cdot)$ denotes the solution map for the given z. If Assumption 4.1 holds, Problem (4.32) is strictly convex by choosing a sufficiently large $\rho < \infty$. When the descent condition (4.28) is not satisfied, the update (4.29) is, in principle, a standard ALADIN step to solve (4.32) defined at the current z-iterate.

Lemma 4.1. *Let $\rho > 0$ such that Assumption 4.1 holds. An iteration of Algorithm 7, with primal and dual variables updated by* (4.29)*, is equivalent to an iteration of Algorithm 5 with full step size for solving convex Problem* (4.32).

Detailed proof of Lemma 4.1 can be found in Appendix A.3.

In contrast to the original globalization strategy (Algorithm 4), which requires solving Problem (3.13), the parametric optimization problem (3.14), Step 5 of Algorithm 7 is computationally

tractable. To implement this step only requires an appropriate linear algebra routine for implementation.

4.4.2 Convergence Analysis

Global Convergence

Lemma 4.1 indicates that if we run Algorithm 7 with only update x and λ, and without update z, the iterate x can still converge to the optimal solution $\xi^\star(z)$ of the auxiliary problem see (4.32) under an additional continuity assumption in f_ℓ (Assumption 3.1). We summarize this technical result below.

Theorem 4.2. *Let $\rho > 0$ such that Assumption 4.1 hold and let the functions f_ℓ be continuously differentiable in a local neighborhood of $\xi^\star(z)$. If Algorithm 7 runs with only update x and λ by (4.29), i.e., Step 5 in Algorithm 7 only executes Step 5b-ii), the x-iterate converges to the optimal solution $\xi^\star(z)$.*

Proof. As Theorem 3.1 reviewed, we have the x-iterate converge to $\xi^\star(z)$ such that we have locally

$$\left\|\nabla f_\ell(x_\ell) + \rho\Sigma_\ell(x_\ell - z_\ell) + A_\ell^\top \lambda^\star(z)\right\|_{H_\ell}^2 \to 0$$

as ∇f_ℓ exists in a local neighbor of $\xi^\star(z)$ according to the continuity assumption (Assumption 3.1). Here, $\lambda^\star(z)$ denotes the associated dual solution map of (4.32). Then, we use the first order optimality condition of (4.25) to write down ∇f_ℓ explicit such that

$$\begin{aligned}
&\sum_{\ell\in\mathcal{R}} \left\|\nabla f_\ell(x_\ell) + \rho\Sigma_\ell(x_\ell - z_\ell) + A_\ell^\top \lambda^\star(z)\right\|_{H_\ell^{-1}}^2 \\
=\ &\sum_{\ell\in\mathcal{R}} \left\|H_\ell(\chi_\ell - x_\ell) - A_\ell^\top(\lambda - \lambda^\star(z))\right\|_{H_\ell^{-1}}^2 \\
=\ &\underbrace{(\lambda - \lambda^\star(z))^\top M(\lambda - \lambda^\star(z)) + \sum_{\ell\in\mathcal{R}} \|\chi_\ell - x_\ell\|_{H_\ell}^2}_{\geq 0} + 2(\lambda - \lambda^\star(z))^\top \underbrace{\left(\sum_{\ell\in\mathcal{R}} A_\ell x_\ell - b\right)}_{\to 0}.
\end{aligned}$$

This indicates that the dual residual $\sum_{\ell\in\mathcal{R}} \|\chi_\ell - x_\ell\|_{H_\ell}^2$ converges to 0, which satisfies the optimal condition for solving the auxiliary problem using standard ALADIN (Remark 3.3). □

To ensure the sufficient descent, the condition

$$\Phi(z) - \Phi(z^+) \geq \gamma \left(\sum_{\ell \in \mathcal{R}} \frac{\rho}{2} \|x_\ell - z_\ell\|_{\Sigma_\ell}^2 + \frac{1}{2} \|x_\ell - \chi_\ell\|_{H_\ell}^2 \right)$$

is checked in Step 5. Here, $0 < \gamma \ll 1$ is same as the one in (4.28). If the termination criterion from Step 2 of Algorithm 7 is violated, the right-hand side of the inequality (4.28) is bounded below by

$$\gamma \left(\sum_{\ell \in \mathcal{R}} \frac{\rho}{2} \|x_\ell - z_\ell\|_{\Sigma_\ell}^2 + \frac{1}{2} \|x_\ell - \chi_\ell\|_{H_\ell}^2 \right) \geq \gamma \frac{\sigma \epsilon^2}{2} ,$$

where $\sigma > 0$ is a constant that depends on the choice of the norm in the termination criterion only.

Theorem 4.3. *Let the assumptions in Theorem 4.2 hold and let Problem (4.24) be feasible and bounded from below such that a minimum exists. If $\rho > 0$ is sufficiently large such that Problem (4.32) is strongly convex for any given z, then Algorithm 7 terminates after a finite number of iterations.*

Proof. As the original Problem (4.24) is assumed to be feasible, all decoupled Problems (4.25) and coupled QP (4.27) are feasible such that the iterates of Algorithm 7 are well-defined. Next, we prove this theorem by following two steps.

First, we show that Step 5b-ii) in Algorithm 7 is executed at most for a finite number of steps by contradiction. To this end, let us first assume that Algorithm 7 applies Step 5b without z update infinitely often, i.e., conditions (4.28) and (4.30) never hold. As Theorem 4.2 shows, we have in this case that the x-iterate converges to the optimal solution $\xi^\star(z)$ of (4.32) at current primal iterate z. Moreover, as z is feasible for the affine equality constraints, which implies that z is a feasible solution of (4.32), we can have

$$\begin{aligned} \sum_{\ell \in \mathcal{R}} f_\ell(z_\ell) &\geq \sum_{\ell \in \mathcal{R}} \left\{ f_\ell(\xi_\ell^\star(z)) + \frac{\rho}{2} \|\xi_\ell^\star(z) - z_\ell\|_{\Sigma_\ell}^2 \right\} \\ \implies \Phi(z) - \Phi(\xi^\star(z)) &\geq \frac{\rho}{2} \sum_{\ell \in \mathcal{R}} \|\xi_\ell^\star(z) - z_\ell\|_{\Sigma_\ell}^2 . \end{aligned}$$

Notice that if $x = \xi^\star(z)$, we have the local solution y satisfy $y = \xi^\star(z)$. This means that the limit point $\xi^\star(z)$ leads to strict descent condition

$$\Phi(z) - \Phi(x) \geq \sum_{\ell \in \mathcal{R}} \left(\frac{\rho}{2} \|\xi_\ell^\star(z) - z_\ell\|_{\Sigma_\ell}^2 + \frac{1}{2} \|x_\ell - \chi_\ell\|_{H_\ell} \right)$$

Thus, by choosing $0 < \gamma \ll 1$, the condition (4.30) must hold after a finite number of iterations, i.e., z cannot be fixed infinitely often. With z update, Step 5b-ii) cannot be executed infinitely often as the right-hand side of (4.30) is bounded below. This contradicts the above assumption that the algorithm applies Step 5b-ii) for infinite iterations.

The second step follows the fact that the right-hand of the descent condition (4.28) is bounded below. As a result, Steps 5a and 5b-i) cannot be executed infinitely often. By now, we have already excluded the cases in which neither Step 5a nor Step 5b is applied for an infinite number of iterations. Therefore, Algorithm 7 must terminate after a finite number of steps. □

Remark 4.8. *By comparing Step 5 with Algorithm 4, both strategies construct a strict descent condition based on the solutions of auxiliary Problem* (3.14) *and* (4.32)*, respectively. Moreover, the proximal convex function F_ℓ is used to formulate both* (3.14) *and* (4.32)*. In spite of this, Algorithm 7 does not really need to solve* (4.32) *at every iteration but Algorithm 4 needs to solve* (3.13) *to determine the step size α_3. One can say that in the worst case, Algorithm 7 may keep on running with Step 5.b) for solving* (4.32) *to satisfy the descent condition* (4.28)*. However, in practice, it always happens that only running a few iterations with Step 5.b) can result in an iterate x^{QP} satisfying* (4.28)*.*

Remark 4.9. *The existing state-of-the-art line-search routines used in the SQP method or interior point method, are typically implemented by back-tracking or other heuristics. This may lead to step rejections and additional computations. In contrast to these methods, the proposed strategy in Algorithm 7 never rejects steps or slows down the progress of the ALADIN iterations.*

Local Convergence with Matrix Update

As discussed in Theorem 3.1, if the functions f_ℓ are twice continuously differentiable at the local solution of Problem (4.24), which is assumed to be a regular KKT point, Algorithm 7 has the same local quadratic convergence rate as the standard ALADIN if Step 3 update H_ℓ satisfying (3.15). This result is summarized in the following theorem.

Proposition 4.2. *Let the assumptions in Theorem 4.3 hold and let the functions f_ℓ be twice continuously differentiable with the local optimal solution (x^*, λ^*) of Problem* (4.24) *being a regular KKT point. Whenever Step 5a is applied at the current iteration, if Step 3 updates matrices H_ℓ satisfying* (3.15) *at the next iteration while the update locally satisfies* (2.15)*, the Algorithm converges globally and has local quadratic convergence rate in a neighborhood of (x^*, λ^*).*

This statement follows the fact that if the iterate $x = z$ in Algorithm 7 contracts with quadratic rate, the function Φ descents in every step in the neighborhood of a regular minimizer [46], i.e., inside such a neighborhood, (4.28) is always satisfied.

4.4.3 Case Study

This section demonstrates the application of Algorithm 7 to solve distributed AC power flow (PF) problems with a poor dual initial guess, a known challenge in distributed optimization when augmented Lagrangian method (ALM) is applied [33].

Problem Setting

The initialization of the dual variables λ^0 plays a pivotal role in ensuring the convergence of the distributed optimization algorithm [33]. Hence, in the case study, we explore the structure of distributed AC PF problems and evaluate the performance of the proposed Algorithm 7 under various initialization configurations for the dual variables λ^0. This comparison, against Algorithm 5, showcases the performance improvements achieved by adopting a globalization strategy.

Following (4.3.1), we formulate the AC PF problems (B.22) in a standard least-squares formulation with affine consensus constraint (4.4) written as

$$\begin{aligned} \min_{x} \quad & f(x) := \sum_{\ell \in \mathcal{R}} f_\ell(x_\ell) = \frac{1}{2} \sum_{\ell \in \mathcal{R}} \|c_\ell(x_\ell)\|_2^2 \\ \text{s.t.} \quad & Ax = b \quad | \ \lambda, \end{aligned}$$

where c_ℓ represents power flow constraints (4.2a) in region $\ell \in \mathcal{R}$.

Due to Proposition 4.1 in Section 4.3.1, it is understood that if a feasible solution for the AC PF problem is identified, the optimal dual variable $\lambda^\star = 0$. Utilizing this knowledge, we employ AC PF problems as benchmarks to analyze the impact of various initial guesses for dual variables. Here, all primal variables are initialized using the flat start strategy, a standard practice for optimization problems in power systems as described in Remark 4.3.

Simulation Results

We evaluate Algorithm 7 by initializing it with 3000 different dual variable settings λ^0, to explore the impact of dual initial guess. The dual deviations $||\lambda^0 - \lambda^\star||_\infty$ range from 0 to more than 10^5. As depicted in Figure 4.4, the full-step ALADIN 7 fails to converge when the initial deviation

of the dual variable exceeds 10^2. In contrast, the proposed ALADIN variant in Algorithm 7 achieves convergence in 99% of cases with different initializations. It is worth noting that all 60 test cases that did not converge experienced failures due to the `Casadi` toolbox's inability to provide deviation information via automatic differentiation, owing to numerical issues.

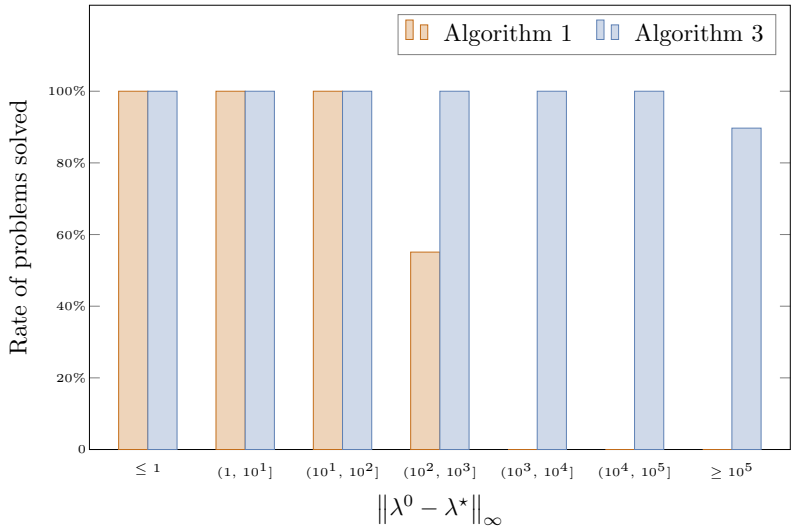


Figure 4.4: Performance improvement by using the proposed globalization strategy

Figure 4.5 demonstrates the significant improvements made possible by employing the proposed globalization strategy. The proximal-step and reserve-step mechanisms are not triggered when the initial dual variable is a close estimate until approaching $||\lambda^0 - \lambda^\star||_\infty \approx 100$, beyond which the full-step ALADIN variant begins to struggle with convergence. As the deviation increases, the reserved step becomes increasingly crucial in ensuring convergence. For the largest deviations, it accounts for almost 80% of the total iterations required for convergence, highlighting its effectiveness in globalizing the solution process when faced with poor initial guesses of dual variables.

Convergence Performance

Figure 4.6 offers deeper insights into the mechanism behind the globalization strategy and depicts its effectiveness in achieving convergence in a test case with a poor initial guess. The performance is evaluated using several metrics:

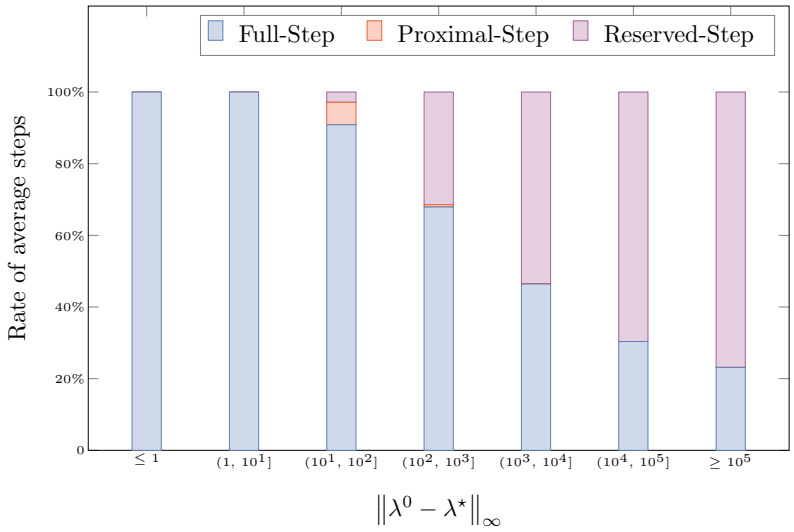


Figure 4.5: Average Steps for the converged test case (Proximal-Step: Step 5b-i) in Algorithm 7, Reserved-Step: Step 5b-ii) in Algorithm 7)

- Deviations to optimizers in terms of primal variables $\|Ax - b\|$, dual variables $\|\lambda - \lambda^\star\|$, and reserved variables $\|z - z^\star\|$;
- Primal residuals $\|Ax - b\|$ and dual residuals $\|x - y\|$;
- The distance between the optimizers of the original Problem (4.24) and the auxiliary Problem (4.32) $\|\xi^\star(z) - z^\star\|$, and between the current iterates and optimizers of the auxiliary Problem (4.32) $\|x - \xi^\star(z)\|$;
- The optimality gap in terms of objective values $|f - f^\star|$.

In this test case, the initial guess for the dual variables is extremely poor, with $\left\|\lambda^0 - \lambda^\star\right\|_2 \geq 10^4$, leading to the failure of full-step ALADIN variant to converge. The large deviation in dual variables prevents the QP step (4.27) and the correction step via linear algebra routine (4.29) from improving the merit function during the first 15 iterations, and thus the z remains unchanged. Without updates to z_ℓ and the Hessian H_ℓ, the iterates x gradually move closer to the solutions of the auxiliary Problem (4.32), as indicated by the slow reduction in $\|x - \xi^\star(z)\|$ in Figure 4.6. Once x enters the neighborhood of the solutions to the auxiliary Problem (4.32), for example when $\|x - \xi^\star(z)\| \leq 10^{-1}$ in this case, one of the descent conditions (4.28) or (4.30) is met, leading to updates in z_ℓ and the Hessian H_ℓ. It confirms the assertions made in Remark 4.8. In

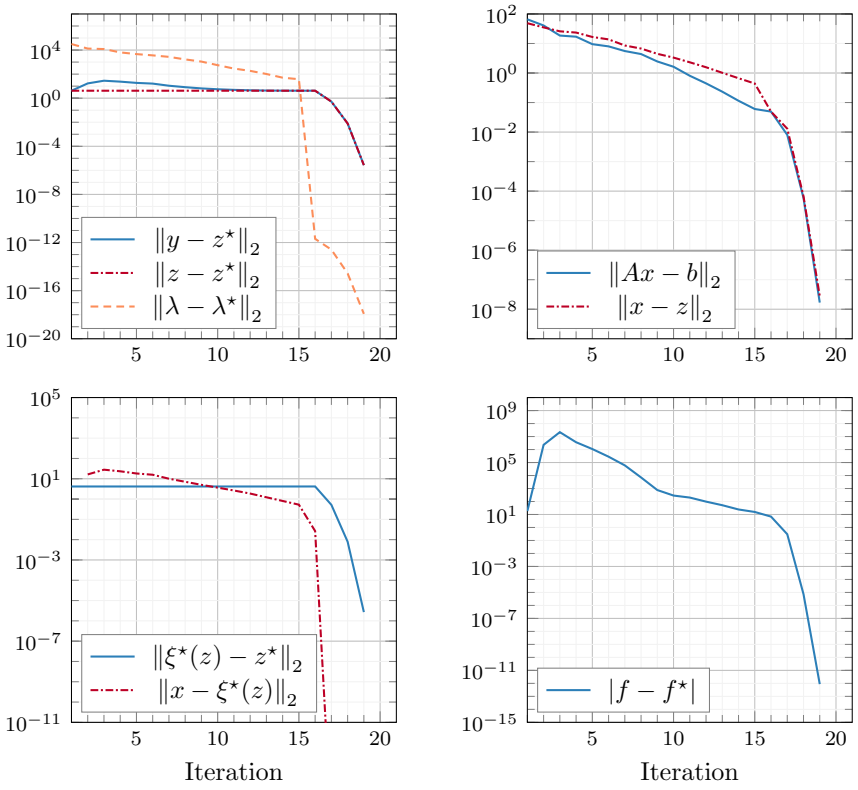


Figure 4.6: Convergence behavior of Algorithm 7 with pool initial guess for the dual variables

the final iterations, full steps are accepted, and Algorithm 7 demonstrates a rapid convergence rate, as anticipated for local convergence behavior.

4.5 Practical Deployment on Distributed System

In this section, we turn from the numerical performance of ALADIN-type algorithms to the practical deployment of the standard ALADIN on distributed systems, taking communication into consideration.

Integrating distributed energy resources (DERs) into modern power grids brings substantial challenges, particularly in balancing the need for efficient management with the protection of sensitive data. In a country like Germany, where four transmission system operators (TSOs) and distribution system operators (DSOs) coexist, the complexity of managing grid operations

is amplified by privacy regulations that restrict centralized data sharing [36]. This situation underscores the need for optimization methods that respect privacy while enabling effective coordination across operators.

One of the most critical tasks in power system operations in this context is solving the AC optimal power flow (OPF) problem. Traditionally, AC OPF has been tackled using centralized methods, but the increasing emphasis on data privacy has shifted attention to distributed optimization approaches. Unlike traditional centralized algorithms, distributed approaches divide the problem across multiple local agents, each responsible for solving a local subproblem. The solutions are then coordinated centrally without exchanging detailed grid information, ensuring data privacy.

Within this framework, the eASiMOV framework's Co-Simulation (eCoSim) module [107] provides a convenient platform for implementing ALADIN. eCoSim is designed to integrate various energy domains, such as electricity, gas, and heat, in a distributed simulation environment. The key benefit of this framework is its ability to maintain model privacy by allowing system operators to keep their data locally while contributing to a global solution through a co-simulation master.

Building on these ideas, we adapt the distributed framework for coupling geographically distributed models and solving AC OPF by using ALADIN. This offers several key advantages in a distributed setup, particularly its robustness to single-point failures. Unlike traditional centralized approaches, where the failure of a central coordinator can disrupt the entire system, ALADIN allows any local agent to solve the coupled QP subproblems, significantly reducing system vulnerability. Additionally, its deployment within the eASiMOV framework ensures the effective coupling of geographically distributed models, achieving scalability and convergence.

4.5.1 Distributed Co-Simulation Framework

The co-simulation framework is designed to enable the integration of various solvers by utilizing different tools or platforms to model individual components of energy systems. This allows different subsystems, modeled using diverse techniques and technologies, to interact and communicate efficiently[113]. A key challenge compared to traditional simulation approaches is ensuring high computational performance, straightforward configuration, and tool compatibility [114]. Another critical issue is maintaining data privacy, particularly when simulations are geographically distributed.

The eASiMOV framework—short for *e*nergy system *A*nalysis, *Si*mulation, *M*odelling, *O*ptimization and *V*isualization [107]—features a module called *e*nergy system *Co-Sim*ulation (eCoSim), which facilitates the co-simulation of multimodal energy systems [115]. This framework enables co-simulation in geographically distributed environments while preserving data privacy. The benefit

of this setup is that experts can contribute to complex simulations without needing to modify their tools or models, ensuring flexibility and collaboration across different environments.

The structure of eCoSim consists of two main components: a Simulation Module, which operates as a stand-alone simulator containing a model and solver, and a Simulation Master, which orchestrates data exchange between modules. The Simulation Master coordinates the entire process, initiating time steps and handling communication between modules using Transmission Control Protocol (TCP). Each simulation module is a "black box," meaning its internal structure is hidden from other modules and the Simulation Master. This ensures data privacy during the simulation process. A central database records all results, data transfer statistics, and system performance metrics such as CPU and memory usage.

4.5.2 Adaptation to Solving Distributed AC OPF using ALADIN

Since the co-simulation framework was originally built to support energy system analysis via coupling simulators, adjustments were necessary to apply it to source code-based simulators, such as those used for distributed AC optimal power flow (OPF) problems. In the following, we propose a solution that uses a MATLAB implementation, although the approach can easily be transferred to other programming languages.

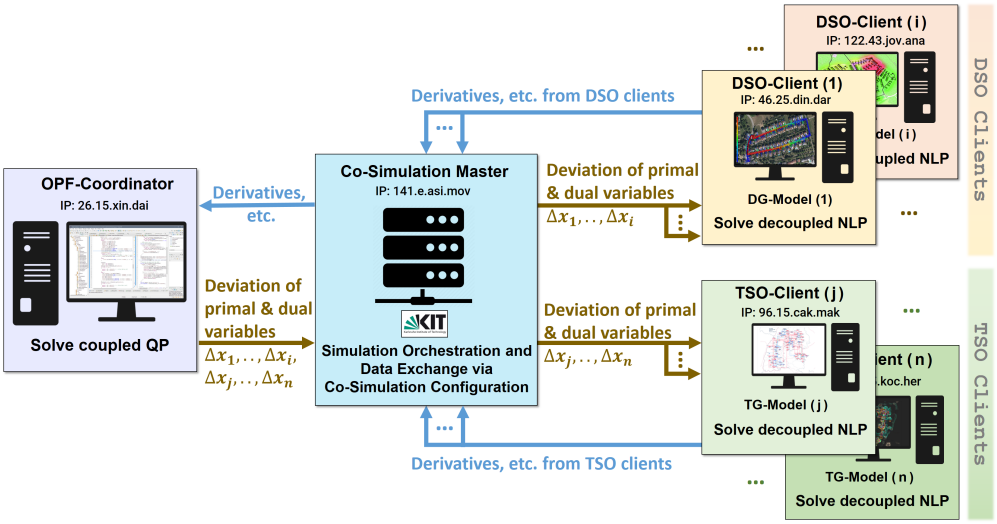


Figure 4.7: The eASiMOV-eCoSim co-simulation architecture enables geographically distributed AC OPF calculation with respect to data and model privacy.

The proposed architecture integrates eCoSim with the OPF calculation by ALADIN is shown in Fig. 4.7. Deploying the standard ALADIN (Algorithm 3) within this architecture for AC OPF involves two primary steps. First, local agents independently solve their respective decoupled nonlinear program (NLP) subproblems (3.2) based on their local grid data. Each agent provides derivative information about its local solution, which is then sent to an OPF-Coordinator. The coordinator solves a coupled quadratic program (QP) (3.5) using this derivative information, and the results—adjustments to the primal and dual variables—are transmitted back to the local agents. This iterative process continues until convergence is achieved, ensuring that data privacy is maintained at all stages, as local agents only share derivative information, not detailed grid or customer data.

This setup allows for distributed AC OPF calculations across geographically separated simulations, maintaining synchronization and coordination via eCoSim's orchestration and communication framework.

4.5.3 Implmenetation Details

The co-simulation framework eCoSim provides a wrapper for the MATLAB code to initialize, run a single step, and stop the MATLAB execution by using standardized function names. By standardizing the interfaces, any MATLAB code that allows for parallelization can be distributed on this platform. The process interactions among coordinator and local agents in time sequence are summarized in Fig. 4.8.

The eCoSim wrapper code is shown in the Algorithms 8 and 9, which illustrate the process from the perspective of a single simulation module—either a client or a coordinator. The inputs for each simulation module include eCoSim commands, a Boolean *sim_running* indicating the current status of a simulation module, and a pre-defined error margin ε for the local clients as a threshold to stop the OPF calculation of the respective module.

Once the eCoSim master setup is complete, a sim_setup command is sent to each connected client. This initializes the clients as shown in lines 3-5 in both algorithms by executing corresponding MATLAB code, containing the initial settings for the TSOs and DSOs.

The execution of one simulation step for the clients (simulation modules) is initiated via a sim_step command sent by eCoSim master. The execution order depends on the co-simulation configuration and guarantees the correct orchestration of the simulation modules. The LDB stops the execution of certain simulation modules until other simulations finish their simulation step and provide their output as an input to the depending simulators. A single simulation step for a local client is shown in lines 7-15. It executes `run_localClient_i.m` where the decoupled NLP

Algorithm 8 Control of TSO-/DSO-Client

Input: eCoSim $command$, $sim\_running$, sim_step, error margin ε
Output: $sim\_running$

```
// Process Control Message
switch command do
    // Initialize Client
    case sim_setup
        init_localClient_i.m
    // Perform Simulation Step
    case sim_step
        if sim_running then
            // Solve decoupled NLP (3.2) (Algorithm 3)
            run_localClient_i.m
            error = error_localClient_i.m
            if error < ε then
                return sim_running = false
            end if
        end if
    // Stop Client
    case sim_stop
        error_global_localClient_i.m
        STOP_SIMULATION_MODULE
end switch
```

Algorithm 9 Control of OPF-Coordinator

Input: eCoSim $command$, $sim\_running$, sim_step

```
// Process Control Message
switch command do
    // Initialize Coordinator
    case sim_setup
        init_coordinator.m
    // Perform Simulation Step
    case sim_step
        if sim_running then
            // Solve Coupled QP (3.5) in Algorithm 3
            run_coordinator.m
        end if
    // Stop Coordinator
    case sim_stop
        STOP_SIMULATION_MODULE
end switch
```

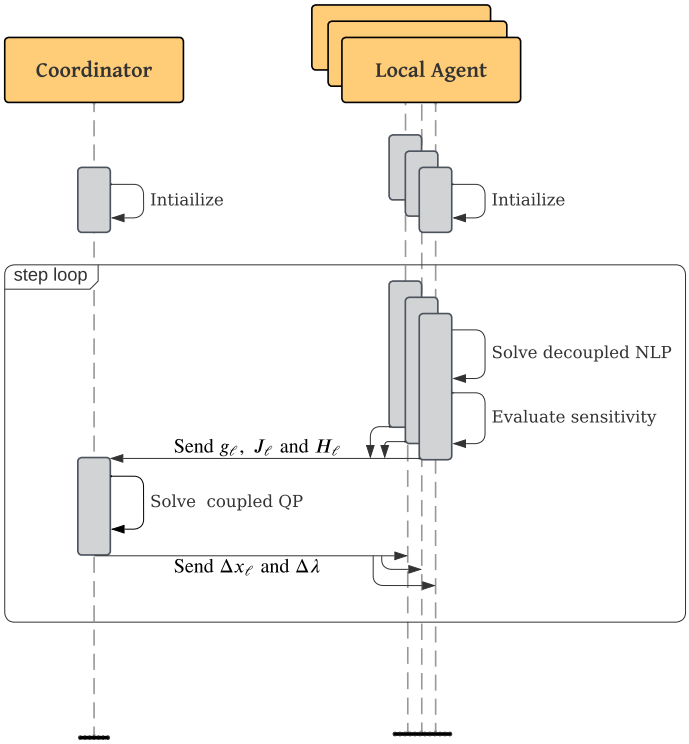


Figure 4.8: Process interactions of the standard ALADIN (Algorithm 3)

subproblems (3.2) is solved. After each simulation step, the error of the local client is calculated and compared to the threshold ε, which signals eCoSim master to stop the local client.

Similarly, the coordinator's simulation step is shown in line 7-11. In contrast to the local client, the coordinator changes in the second iteration to *sim_running* = `True` since it waits for the first results from the clients and executes `run_coordinator.m` where the coupled QP subproblems (3.5) is solved. As soon as all local clients signal the end of their computations for one iteration step, all simulation modules (local clients and the coordinator) are stopped via the sim_stop command sent by eCoSim master (lines 17-20 and lines 13-15 respectively).

Fig. 4.9 illustrates the integration of MATLAB OPF code into eCoSim. The TSO and DSO clients solve the decoupled NLP problems, while the coordinator handles the coupled QP problem. Communication between the eCoSim master and the simulation modules is managed using three primary commands. Data exchange occurs through network storage, with each of the N simulation

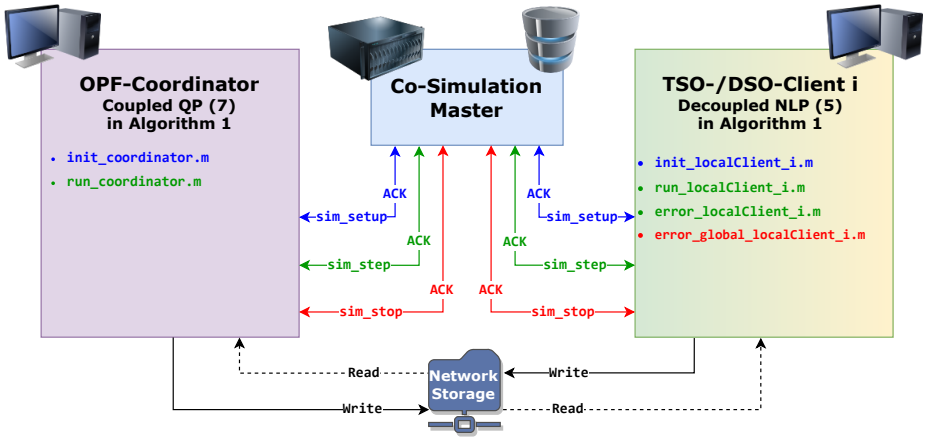


Figure 4.9: Integration of MATLAB OPF code (Algorithm 1) into the eCoSim control code (Algorithm 2 and 3).

modules utilizing two files: one for the primal and dual variable deviations (produced by the coordinator) and one for derivative data (produced by the clients) as depicted in Fig. 4.7.

During each sim_step, the local clients read from the storage, perform their calculations, and then write their results back to their respective files. The coordinator follows the same procedure but writes to and reads from the opposite set of files used by the clients. Data privacy is maintained throughout the simulation process since no sensitive data is exchanged during these read-and-write operations.

4.5.4 Simulation Result and Discussion

The OPF framework is built on MATLAB-R2020b, the ITD systems are merged based on the open-source toolbox rapidPF [6]* and power systems model is built with the assistance of Matpower toolbox [112]. The case study is carried out on a standard laptop computer with `Intel® Core™ i7-8850H CPU @ 2.60GHz` and `16GB` installed RAM. CasADi toolbox [111] is used for modeling optimization problems and IPOPT [71] are used as nonlinear solver. For tuning parameters in the proposed method, an adaptive heuristics approach is adopted, as discussed in [34]. The numerical test case is built upon the IEEE benchmarks, where the TSO model uses a 57-bus transmission system from PGLib [116] and two DSO models use 33-bus distribution systems from the Matpower package [112]. For both the local and the truly distributed setups using eCoSim, the same configuration is used for the integration. The configuration inside eCoSim

*The code is available on `https://github.com/xinliang-dai/rapidPF`

consists of three local clients, one coordinator, and one LDB. The local clients are connected to the coordinator via the LDB, and the coordinator is connected to the local clients in return. The LDB ensures the correct data exchange between the local clients and the coordinator inside the storage network.

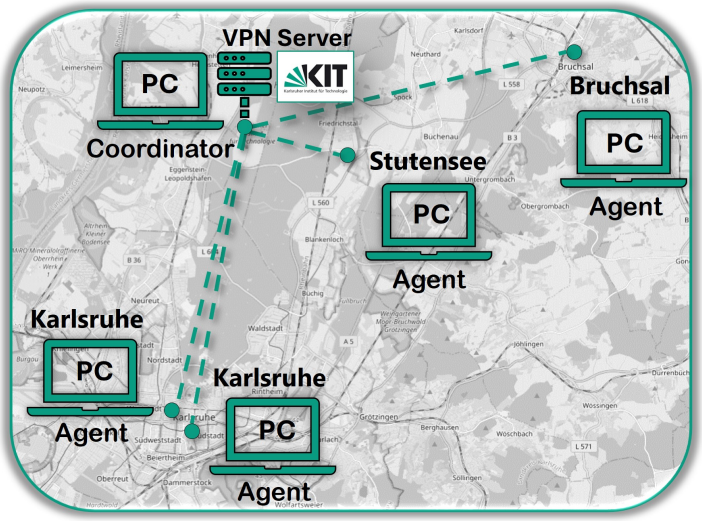


Figure 4.10: Geographically distributed computation environment within a 15km radius of KIT (eCoSim-KIT5 setup)

The simulation is conducted in five distinct setups, as depicted in Fig. 4.11. The first setup, shown in Fig.4.11(a), employs IPOPT for centralized optimization on a single computer. The next two configurations utilize ALADIN for distributed optimization, also on a single computer, differing in their approach to coordination and communication; specifically, eCoSim-KIT1 introduces geographically distributed co-simulation and utilize network storage at KIT for data exchanges. The last two setups demonstrate true distributed execution by distributing the eCoSim master module and simulation modules across multiple computers. The distinction lies in that eCoSim-Geo5 is configured similarly to eCoSim-KIT5 but incorporates a geographical distance among the computers, all located within a 15km radius of KIT as illustrated in Fig. 4.10, exploring distributed computing effects over short geographical distances.

We first compare the runtime behavior of the five different setups explained in the previous subsection, as shown in Fig. 4.11. The y-axis shows the runtime in seconds for the introduced setups. The two MATLAB-based setups IPOPT and ALADIN have a total runtime of 0.420 and 0.679 seconds, respectively. Both of them are executed sequentially on a single machine without communication effort. The other three setups for ALADIN-eCoSim are divided into the time for the OPF (calculation), writing-/reading the network storage (network storage operations) and

the eCoSim synchronization-/overhead time (synchronization). For each of these three cases, the runtime consists of an average of ten runs. The coordinator is chosen as the reference for the runtimes evaluation. It represents the best runtime, as it needs to interact with each local client's network storage. The calculation runtime of the locally distributed AC OPF is about the same as the MATLAB implementations. The differences in the calculation runtime can be attributed to the different execution environments and, therefore, resulting in different measuring methods. The averaged total runtime for the ALADIN-eCoSim-KIT1 case is 3.208 seconds, whereas the calculation time is 0.571, the synchronization time is 1.701 and the time for network storage operations is 0.936 seconds. In the distributed case ALADIN eCoSim-KIT5 with five computers, the time for calculation is 0.589 seconds, the time for the network storage operations is 1.763 seconds, and the time for the synchronization is 3.114 seconds. In the ALADIN-eCoSim-KIT5 solution, the total runtime is 5.466 seconds. The calculation time is 0.589 seconds, the synchronization takes 3.114 seconds, and the time for the network storage operations is 1.763 seconds. Compared to the eCoSim-KIT1 solution, distributing the modules onto different computers raises the time effort for the synchronization and network storage operations. For the geographically distributed computing over the VPN at the KIT (ALADIN-eCoSim-Geo), the total time is 18.152 seconds, whereas the calculation time is 0.797, the synchronization time is 7.216 and the time for network storage operations is 10.139 seconds. The communication over the VPN is significantly higher, which in turn, is compensated by data security and privacy. One reason for this is the network storage location at the KIT and thus the additional time needed to access the network storage from outside the KIT over a VPN. Another reason for this could be the amount of concurrent users in the KIT VPN.

The numerical convergence performance of Alg. 3 is illustrated in Fig. 4.12, for which the centralized approach (IPOPT) is used as the reference solution. After five iterations, the ALADIN algorithm can approach the reference solution with very high accuracy with respect to state deviation $\|x - x^*\|$ and objective value $|(f - f^*)/f^*|$. Meanwhile, the primal residuals $\|Ax - b\|$ and dual residuals $\|x - z\|$ approach zeros, indicating the algorithm converges to a very small neighborhood of the reference solution with negligible violation of coupling constraints. The solution accuracy by applying ALADIN is demonstrated in Table 4.3, affirming that all three approaches by applying ALADIN converge to the same reference solution computed by IPOPT.

The proposed distributed framework can maintain data privacy and decision-making independence. The case study shows that the distributed co-simulation environment effectively keeps model topology private in exchange for higher runtime, which might be significantly reduced in the future with a direct data exchange without the use of network storage.

This deployment showcases a scalable and privacy-preserving solution to the AC OPF problem, one that can handle the nonconvex nature of the problem while adhering to modern data privacy

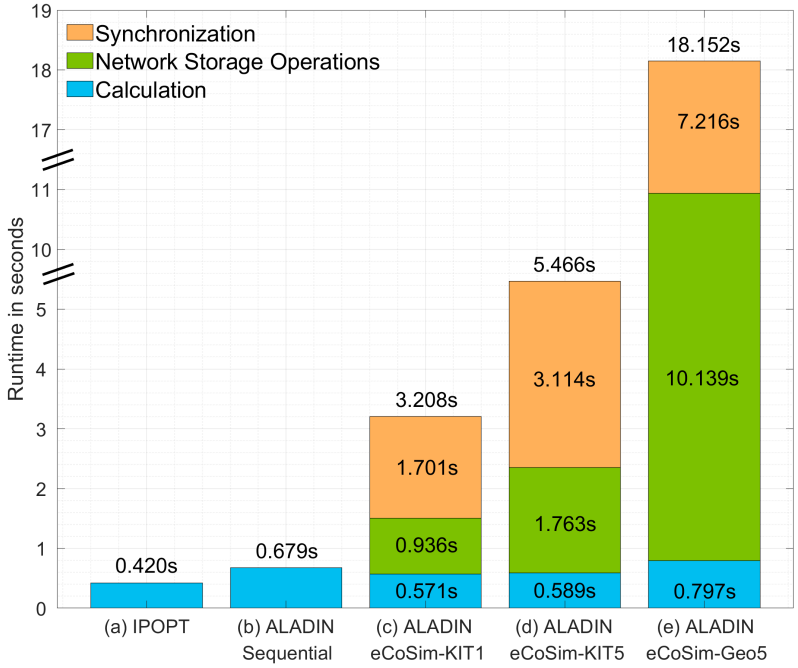


Figure 4.11: Runtime comparison for the use cases with a serial MATLAB implementation in (a) IPOPT and (b) ALADIN, a distributed execution with eCoSim on one computer in the KIT network in (c) eCoSim-KIT1, on five computers in the KIT network in (d) eCoSim-KIT5 and a geographically distributed co-simulation with access to the network storage located at KIT over a VPN connection in (e) eCoSim-Geo5. The clients are distributed over three cities with a geographical distance of up to 15 km to KIT (the internet routing Runtimes are measured at the coordinator software module located at KIT.

Table 4.3: Comparison Numerical Results

	IPOPT	ALADIN		
		Sequential	eCoSim-1	eCoSim-5
Cost	34 210.54	34 210.55	34 210.55	34 210.55
Optimality Gap	-	5.07×10^{-8}	2.46×10^{-8}	2.46×10^{-8}
Primal Res.	-	3.22×10^{-8}	8.70×10^{-8}	8.70×10^{-8}
Dual Res.	-	6.41×10^{-6}	4.64×10^{-7}	4.64×10^{-7}
$\|x - x^*\|$	-	7.21×10^{-7}	2.88×10^{-7}	2.88×10^{-7}

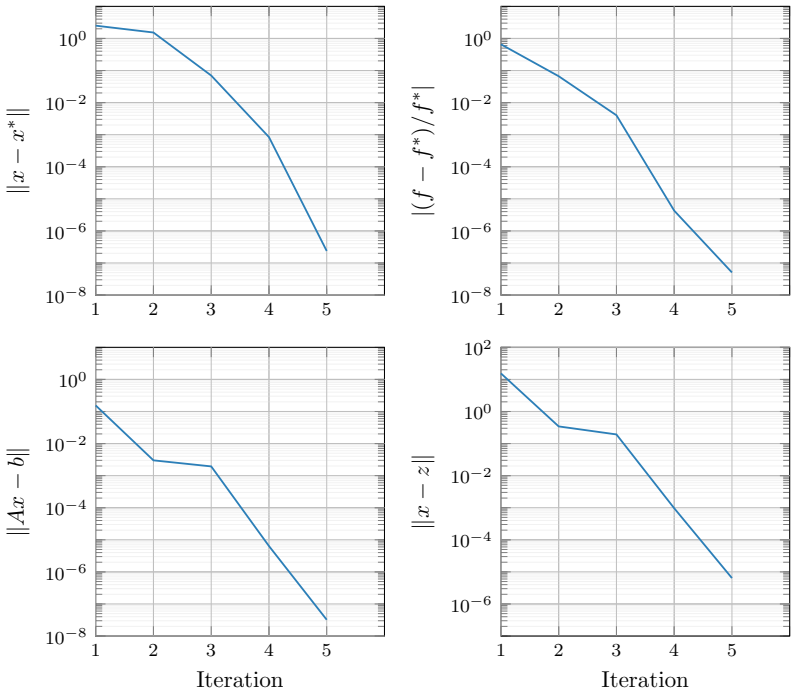


Figure 4.12: Numerical Results by proposed distributed algorithm.

requirements. The methodology outlined provides a foundation for future research and real-world applications, as it can easily be extended to other areas of distributed optimization in power systems.

4.6 Summary

This chapter focuses on how ALADIN is applied application of to power systems, c to power systems, including efforts to enhance both local and global performance of ALADIN-type algorithms as well as one example of practical deployment in geographically distributed settings. These offer insights and potential directions for future research in the practical deployment of distributed approaches.

However, like standard sequential quadratic programming (SQP) methods, ALADIN faces scalability limitations when handling inequality constraints. The active-set method introduces combinatorial challenges [46, Ch. 15.2], as the number of possible active sets increases exponentially

with the number of inequalities. These challenges highlight the need to develop scalable distributed algorithms to efficiently handle inequality constraints, as noted in [36] [117, Ch. 5.2].

5 BALADIN: A Scalable Distributed Barrier Approach

In the previous chapters, we introduced the foundational concepts of nonlinear programming (NLP) and distributed optimization. Among these, the ALADIN algorithm, which combines centralized SQP with classical ADMM, stands out for its theoretical advantages in distributed optimization and its fast local convergence rate. However, as discussed in Section 3.4, ALADIN encounters scalability challenges when handling inequality constraints [36] [117, Ch. 5.2], a limitation similar to the combinatorial complexity of active-set SQP [46, Ch. 15.2]. Additionally, ALADIN encounters significant communication challenges, as it requires transferring the full approximated model to the coordinator. This not only leads to large communication overhead but also raises potential security concerns, as sensitive local data may be exposed.

This chapter introduces a novel distributed optimization algorithm designed for large-scale nonlinear programming with a convergence guarantee and rapid convergence rate. Building on centralized IPM principles, the proposed distributed optimization algorithm employs barrier-based smoothing techniques to manage inequality constraints and uses ALADIN to solve the smoothed barrier problems, effectively circumventing combinatorial challenges. By gradually reducing the barrier parameter to zero, the algorithm converges to local minima. To further mitigate communication overhead and improve data privacy, a condensing step is incorporated. This step reduces communication requirements while safeguarding network topology information embedded in the sparsity pattern of the derivatives.

Furthermore, optimal network decomposition enables efficient parallel processing under the single program multiple data (SPMD) paradigm. Extensive simulations on large-scale AC optimal power flow (OPF) benchmarks demonstrate that the proposed framework outperforms the state-of-the-art centralized solver IPOPT [71], even on modest hardware such as a 48-core workstation.

Outline. Section 5.1 reviews prior research on distributed approaches for large-scale AC OPF problems. Section 5.2 presents the proposed distributed optimization algorithm. Section 5.3 provides a convergence analysis, building on the analysis of the standard ALADIN presented

in Section 3.3. Practical considerations for implementation, including network decomposition, distributed automatic differentiation (AD), and resilience to single-point failures, are discussed in Section 5.4. Section 5.5 discusses the large-scale simulation results across various operating scenarios in detail, while Section 5.6 examines communication efforts under both theoretical and practical scenarios. Finally, Section 5.7 concludes the chapter. This chapter is mainly based on the article [38].

5.1 Introduction

Distributed optimization has recently attracted significant attention due to its scalability advantages. By decomposing large-scale problems into smaller subproblems that can be solved in parallel, these approaches preserve data privacy and reduce communication overhead by eliminating the need for centralized data sharing [4, 12, 5, 23]. While these algorithms often perform well for convex problems, they encounter convergence challenges when applied to nonconvex cases like the AC optimal power flow (OPF) problem. AC OPF is a fundamental optimization task in electric power systems engineering [110] and is generally NP-Hard, even for radial power grids [24, 25].

To tackle the nonconvex AC OPF problems in a distributed fashion, some research leveraged specific network structures. For instance, studies on integrated transmission-distribution (ITD) systems [18, 118] focus on configurations where multiple distribution networks are connected to a single transmission network in a star-like pattern (types III and IV in Table 5.1). Others have examined fully radial networks (type V) [119, 120, 121]. These approaches, following master-slave-splitting [122], iteratively solve the overall problem by initializing and fixing boundary voltages, solving the decoupled AC OPF for the slave systems, and then integrating the solutions into the master system. However, their performance often depends on the master system's convergence properties and the convexity of each slave subsystem's feasible set. Moreover, these methods usually rely on one central master node. Recent work [123] improves scalability by applying an interior point method that smooths the feasible set and implicit sensitivities in the slave systems. Although some studies have extended these techniques to more complex topologies (type II) [36, 124], the testing is usually limited to standard IEEE cases with fixed connecting lines, leaving the impact of coupling density and their applicability to real-world systems unknown.

Instead of merging cases, in this chapter, we partition overall networks into multiple subgrids, creating more complex and realistic coupling structures (type I in Table 5.1). These approaches reformulate the overall problems in standard distributed forms and use generic distributed optimization algorithms to solve them. Such methods can be easily extended to ITD systems, distribution systems, and other optimization and control problems in power systems. Among these,

Table 5.1: Distributed problem formulation of AC OPF

Type	Network	Topology	Model	Problem Type
I	Partitioned	Generic	BIM	Nonconvex
II	Merged	Generic	BIM	Nonconvex
III	Merged	Stellate	BIM	Nonconvex
IV	Merged	Stellate	Hybrid	Partially Convexified
V	Merged	Stellate	BFM	Convexified

Table 5.2: Distributed optimization for solving large-scale AC OPF

Ref.	Type	Algorithm	Global/Local Convergence	N^{bus}	Execution	Speed	Accuracy
[18, 118]	IV	DCC	Local	10^3	Sequential	+++	+++
[119, 120, 121]	V	ADMM	*NaN*	10^3	Sequential	+	+
[123]	III	Distr. IPM	Local	10^7	Parallel	+++	
[124]	II	Distr. IPM	Local	10^2	Sequential	+++	+++
[36]	II	ALADIN	Global	10^3	Sequential	+++	+++
[29]	I	ADMM	*NaN*	10^2	Sequential	+	+
[96]	I	ADMM	*NaN*	10^3	Sequential	+	+
[125]	I	ADMM	*NaN*	10^4	Sequential	+	+
[32, 126]	I	ADMM	Global	10^4	Parallel	+	+
[100, 34]	I	ALADIN	Global	10^2	Sequential	+++	+++
[127]	I	ALADIN	Global	10^1	Sequential	+++	+++
[40]	I	ALADIN	Global	10^2	Distributed	++	+++

alternating direction method of multipliers (ADMM) has gained significant attention. Early work by [106, 29] proposed regional decomposition but was limited to test cases up to one hundred buses. To improve scalability, [128] suggested partitioning grids based on power flow sensitivity, making their approach feasible for larger problems in [96]. Further scalability improvements were achieved by [125] that introduced an acceleration scheme that enabled ADMM to solve the case13659 in MATPOWER[112]. Recent studies [32, 126] have bridged the theoretical gap by utilizing a two-level framework with global convergence guarantees for AC OPF, capable of solving the case30000 from the PGLib-OPF benchmark [116]. However, as a first-order optimization algorithm, the primary drawback of ADMM-type algorithms for nonconvex problems is the requirement for hundreds to thousands of iterations to achieve modest accuracy. Even if the accuracy is sufficient in practice, the intensive communication needed among distributed agents to solve one instance of the optimization problem requires an expensive communication infrastructure with high bandwidth and low delays.

To accelerate convergence, augmented Lagrangian based alternating direction inexact Newton (ALADIN), first proposed in [33] for generic nonconvex distributed optimization problems, combines ADMM and sequential quadratic programming (SQP) frameworks. This approach provides global convergence guarantees and achieves a locally quadratic convergence rate when Hessian is properly selected. ALADIN was first applied to AC OPF problems in [100, 34], requiring only 26 iterations to reach around 10^{-5} accuracy for case300. As a generic nonlinear solver with convergence guarantees, the algorithm was first employed in a geographically distributed computing environment in [40]. This study highlights significant synchronization delays caused by delivering the full KKT matrix information via network storage systems. In [127], further efforts were made to reduce communication effort by condensing derivatives using null space and Schur complement methods, but scalability remains limited. Although ALADIN has demonstrated good scalability in the absence of inequality constraints [33, 6, 39], its scalability is challenged due to using the active-set method, leading to combinatorial challenges [46, Ch. 15.2] as noted in [36] [117, Ch. 5.2].

In this chapter, we introduce a novel approach to address the challenges of solving large-scale nonconvex problems and evaluate it using the large-scale AC OPF benchmark from the PGLib-OPF library [116]. First, we propose a distributed problem reformulation based on a graph-based decomposition that minimizes connecting lines and enables parallel computing across local workers. This method can be generalized to other power system operation problems involving repetitive component types, such as buses, lines, and generators. We also develop a two-level distributed optimization algorithm for generic nonlinear program (NLP), combining barrier methods [46, Ch. 19] [71] to manage inequality constraints at the upper level with the ALADIN algorithm at the lower level for solving subproblems sequentially. By using the Schur complement [129], we reduce the communication effort, enhancing the distribution of computations. Lastly, extensive simulations based on the single program multiple data (SPMD) paradigm demonstrate the effectiveness of this approach, outperforming state-of-the-art centralized solvers like IPOPT [71] even in complex scenarios while accounting for communication overhead and synchronization delays.

5.2 Barrier ALADIN

In this section, we will first introduce generic problem formulation.

In contrast to the distributed form introduced for the ALADIN algorithm in (3.1), we propose a smoothing technique to solve generic nonlinear program (NLP) problems in the following form:

$$\min_{x} \quad f(x) = \sum_{\ell \in \mathcal{R}} f_\ell(x_\ell) \tag{5.1a}$$

$$\text{s.t.} \quad \sum_{\ell \in \mathcal{R}} A_\ell x_\ell = b \tag{5.1b}$$

$$c_\ell^{\mathrm{E}}(x_\ell) = 0 \tag{5.1c}$$

$$c_\ell^{\mathrm{I}}(x_\ell) \leq 0 \tag{5.1d}$$

where $f_\ell : \mathbb{R}^{N_\ell^x} \to \mathbb{R}$, $c_\ell^{\mathrm{E}} : \mathbb{R}^{N_\ell^x} \to \mathbb{R}^{N_\ell^{\mathrm{E}}}$, and $c_\ell^{\mathrm{I}} : \mathbb{R}^{N_\ell^x} \to \mathbb{R}^{N_\ell^{\mathrm{I}}}$ represent local objective, equality, and inequality constraints for region $\ell \in \mathcal{R}$, respectively. Here, N_ℓ^x, N_ℓ^{E} and N_ℓ^{E} denote the number of local state variables, equality constraints, and inequality constraints, respectively. The corresponding KKT conditions are given as:

$$\nabla f_\ell(x_\ell) + R_\ell^\top \kappa_\ell + J_\ell^\top \gamma_\ell + A_\ell^\top \lambda = 0, \quad \forall \ell \in \mathcal{R} \tag{5.2a}$$

$$c_\ell^{\mathrm{E}}(x_\ell) = 0, \quad \forall \ell \in \mathcal{R} \tag{5.2b}$$

$$c_\ell^{\mathrm{I}}(x_\ell) \leq 0, \quad \forall \ell \in \mathcal{R} \tag{5.2c}$$

$$\sum_{\ell \in \mathcal{R}} A_\ell x_\ell - b = 0, \tag{5.2d}$$

with the dual variables for inequality constraints satisfying:

$$\kappa_\ell \geq 0, \quad \forall \ell \in \mathcal{R}, \tag{5.2e}$$

and the complementary slackness condition:

$$\kappa_\ell^\top c_\ell^{\mathrm{I}}(x_\ell) = 0, \quad \forall \ell \in \mathcal{R}. \tag{5.2f}$$

Here, λ, κ_ℓ and γ_ℓ are Lagrangian multipliers with respect to consensus constraints (5.1b), local inequality constraints (5.1d) and local equality constraints (5.1c), respectively. The Jacobians of the local constraints are defined as

$$R_\ell = \nabla c_\ell^{\mathrm{I}}(x_\ell) \text{ and } J_\ell = \nabla c_\ell^{\mathrm{E}}(x_\ell),$$

where $R_\ell \in \mathbb{R}^{N_\ell^{\mathrm{I}} \times N_\ell^x}$ and $J_\ell \in \mathbb{R}^{N_\ell^{\mathrm{E}} \times N_\ell^x}$ represent the gradients of the local inequality and equality constraints, respectively.

Remark 5.1. *In the standard ALADIN, we use the distributed form* (3.1)*, where both equality and inequality constraints are collectively represented as* $c(x) \leq 0$*. This is because ALADIN*

employs the active-set method. Within this framework, equality and inequality constraints are treated similarly—during the formulation of the QP subproblems—active constraints are treated as equality constraints (Definition 2.1), while inactive constraints are ignored.

5.2.1 Algorithm Framework

To solve the Problem (5.1), we employ a two-level framework. At the upper level, we adopt barrier methods to handle inequality constraints, as outlined in Section 2.3.2, instead of relying on the active-set method. By applying smoothing techniques, the original problem (5.1) is transformed into a series of equality-constrained barrier problems:

$$\begin{aligned}
\min_{x,s} \quad & f^\mu(x,s) = \sum_{\ell\in\mathcal{R}} f_\ell^\mu(x_\ell, s_\ell) && && \text{(5.3a)}\\
\text{s.t.} \quad & \sum_{\ell\in\mathcal{R}} A_\ell x_\ell = b && \mid \quad \lambda && \text{(5.3b)}\\
& c_\ell^{\mathrm{E}}(x_\ell) = 0 && \mid \quad \gamma_\ell,\ \ell\in\mathcal{R} && \text{(5.3c)}\\
& c_\ell^{\mathrm{I}}(x_\ell) + s_\ell = 0 && \mid \quad \kappa_\ell,\ \ell\in\mathcal{R} && \text{(5.3d)}
\end{aligned}$$

where the local barrier objective is:

$$f_\ell^\mu(x_\ell, s_\ell) = f_\ell(x_\ell) - \mu \sum_{m\in\mathcal{C}_\ell^{\mathrm{I}}} \ln(s_\ell^{(m)}),$$

and $s_\ell \geq 0$ is implicitly enforced because minimization with the barrier term prevents s_ℓ from approaching zero for all $\ell \in \mathcal{R}$ [46].

The corresponding *perturbed* KKT conditions of the barrier problems (5.3) are:

$$\begin{aligned}
\nabla_x \mathcal{L}_\ell := \nabla f_\ell(x_\ell) + R_\ell^\top \kappa_\ell + J_\ell^\top \gamma_\ell + A_\ell^\top \lambda = 0, \quad & \forall \ell\in\mathcal{R} && \text{(5.4a)}\\
-\mu\mathbb{1} + \mathcal{S}_\ell\, \kappa_\ell = 0, \quad & \forall \ell\in\mathcal{R} && \text{(5.4b)}\\
c_\ell^{\mathrm{E}}(x_\ell) = 0, \quad & \forall \ell\in\mathcal{R} && \text{(5.4c)}\\
c_\ell^{\mathrm{I}}(x_\ell) + s_\ell = 0, \quad & \forall \ell\in\mathcal{R} && \text{(5.4d)}\\
\sum_{\ell\in\mathcal{R}} A_\ell\, x_\ell - b = 0 & && \text{(5.4e)}
\end{aligned}$$

with $\mathcal{S}_\ell = \mathrm{diag}(s_\ell)$, where $\mathbb{1}$ denotes the vector of all ones with the respective dimension. Note that the conditions (5.4) for $\mu = 0$, together with $\kappa_\ell,\ s_\ell \geq 0,\ \forall \ell \in \mathcal{R}$, are equivalent to (5.2), i.e., the KKT conditions for the original problem (5.1) [71].

Remark 5.2. *The solution to the problem* (5.3) *is given by*

$$x^\star(\mu) = \arg\min_x \ \text{(5.3a)}, \ \textit{s.t.} \ \text{(5.3b)} - \text{(5.3d)}$$

As the barrier parameter μ approaches zero, the solution $x^\star(\mu)$ converges to the solution of the original problem (5.1).

Then, in the lower level, ALADIN is used to solve a smoothed subproblem (5.3) for a fixed value of barrier parameter μ in a distributed manner. Once sufficient accuracy is achieved, the barrier parameter is decreased, and the process continues the solution of the next barrier problem from the approximate solution of the previous one, until the algorithm converges.

Decoupled Step

The proposed distributed algorithm is outlined in Algorithm 10 in detail. Firstly, decoupled barrier NLP subproblems (5.5) are solved locally for each region $\ell \in \mathcal{R}$. After solving these local subproblems, the derivatives are evaluated and condensed locally at the local solution $(x_\ell, \ s_\ell)$. These derivatives are later utilized in the next stages of the distributed optimization process to facilitate communication and coordination between regions. It is important to note that both Step 1 and Step 2 can be executed in parallel, enhancing the efficiency of the overall process.

Remark 5.3. *When solving the decoupled NLP subproblems, the fraction-to-boundary method* (2.25) *is essential to ensure that the dual variables κ_ℓ and the slack variables s_ℓ remain positive throughout the iteration. This is a critical requirement for the proper functioning of barrier-based approaches.*

Assessing Optimality Conditions

To assess the convergence of the inner ALADIN algorithm, we need to evaluate the perturbed KKT conditions (5.4). After solving the decoupled barrier NLP subproblems (5.5), we first evaluate the decoupled optimality residual (5.4a)-(5.4d) locally, defined by:

$$E_\ell^\mu(x_\ell, s_\ell, \lambda, \kappa_\ell, \gamma_\ell) = \max \left\{ \frac{\|\nabla_x \mathcal{L}_\ell(x_\ell, \lambda, \kappa_\ell, \gamma_\ell)\|_\infty}{s_\ell^d}, \frac{\|S_\ell \ \kappa_\ell - \mu \mathbb{1}\|_\infty}{s_\ell^c}, \left\| \begin{bmatrix} c_\ell^{\mathrm{E}}(x_\ell) \\ c_\ell^I(x_\ell) + s_\ell \end{bmatrix} \right\|_\infty \right\}, \tag{5.11}$$

with scaling parameters $s_\ell^d, \ s_\ell^c \geq 1$.

Algorithm 10 Barrier ALADIN

Initialization:

Input: initial primal and dual points $(z,\ \lambda)$, positive penalty parameters ρ, μ and scaling symmetric matrices $\Sigma_\ell \succ 0$

Repeat:

1: Solve decoupled NLPs for all $\ell \in \mathcal{R}$ ▷ *Parallel*

$$\min_{x_\ell, s_\ell} \quad f_\ell^\mu(x_\ell, s_\ell) + \lambda^\top A_\ell x_\ell + \frac{\rho}{2}\|x_\ell - z_\ell\|^2_{\Sigma_\ell} \tag{5.5a}$$

$$\text{s.t.} \quad c_\ell^{\mathrm{E}}(x_\ell) = 0 \qquad | \ \gamma_\ell, \tag{5.5b}$$

$$c_\ell^{\mathrm{I}}(x_\ell) + s_\ell = 0 \qquad | \ \kappa_\ell, \tag{5.5c}$$

with κ_ℓ and s_ℓ remaining positive.

2: Evaluate E_ℓ^μ and E_ℓ^0 from (5.11), and condense derivatives for all $\ell \in \mathcal{R}$ ▷ *Parallel*

$$W_\ell = -\bar{A}_\ell\, \bar{H}_\ell^{-1} \bar{A}_\ell^\top \tag{5.6a}$$

$$h_\ell = A_\ell x_\ell - \bar{A}_\ell\, \bar{H}_\ell^{-1} \bar{g}_\ell \tag{5.6b}$$

where $\bar{H}_\ell$, $\bar{g}_\ell$ and $\bar{A}_\ell$ are local curvature information at decoupled solution $(x_\ell,\ s_\ell)$; more detailed description see (5.16)-(5.20).

3: Send E_ℓ^μ, E_ℓ^0, $A_\ell x_\ell$, W_ℓ and h_ℓ to coordinator, and run Algorithm 11 to correct inertia if the inertia condition (5.24) is not satisfied ▷ *Synchronize*

4: Terminate if condition (5.13) is satisfied. ▷ *Centralized*

5: Update barrier parameter by (5.15) if condition (5.14) is satisfied. ▷ *Centralized*

6: Solve coordination problem ▷ *Centralized*

$$W \Delta\lambda = -h \tag{5.7}$$

with $W = \sum_{\ell \in \mathcal{R}} W_\ell$

7: Send $\Delta\lambda$ back to each local agent ▷ *Synchronize*

8: Recover local primal-dual step for all $\ell \in \mathcal{R}$ ▷ *Parallel*

$$\begin{pmatrix} \Delta x_\ell \\ \Delta \gamma_\ell \end{pmatrix} = -\bar{H}_\ell^{-1} \bar{A}_\ell^\top \Delta\lambda - \bar{g}_\ell, \tag{5.8a}$$

$$\Delta s_\ell = -c_\ell^{\mathrm{I}}(x_\ell) - s_\ell - R_\ell\, \Delta x_\ell, \tag{5.8b}$$

$$\Delta\kappa_\ell = -\kappa_\ell + \mathcal{S}_\ell^{-1}(\mu\, \mathbb{1} - \mathcal{K}_\ell \Delta s_\ell), \tag{5.8c}$$

and obtain local primal-dual steplength $(\beta_\ell^{\mathrm{p}},\ \beta_\ell^{\mathrm{d}})$ by using the fraction-to-boundary method

$$\beta_\ell^{\mathrm{p}} = \max\{\beta \in (0,1] :\ s_\ell + \beta \Delta s \geq (1-\tau) s_\ell\} \tag{5.9a}$$

$$\beta_\ell^{\mathrm{d}} = \max\{\beta \in (0,1] :\ \kappa_\ell + \beta \Delta\kappa \geq (1-\tau) \kappa_\ell\}. \tag{5.9b}$$

9: Update primal-dual variables by ▷ *Synchronize*

$$z_\ell^+ = z_\ell + \alpha_1 (x_\ell - z_\ell) + \alpha_2\, \beta^{\mathrm{p}} \Delta x_\ell, \qquad \forall \ell \in \mathcal{R} \tag{5.10a}$$

$$s_\ell^+ = s_\ell + \alpha_2\, \beta^{\mathrm{p}} \Delta s_\ell, \qquad \forall \ell \in \mathcal{R} \tag{5.10b}$$

$$\lambda^+ = \lambda + \alpha_3\, \beta^{\mathrm{p}} \Delta\lambda. \tag{5.10c}$$

with

$$\beta^{\mathrm{p}} = \min_{\ell \in \mathcal{R}} \beta_\ell^{\mathrm{p}} \tag{5.10d}$$

Here, $\beta^{\mathrm{p}} = \min_{\ell \in \mathcal{R}} \beta_\ell^{\mathrm{p}}$, and α_1, α_2, α_3 are defined by Algorithm 4 for the barrier problems (5.3). Alternatively, for full-step updates without globalization, set $\alpha_1 = \alpha_2 = \alpha_3 = 1$.

Each region $\ell \in \mathcal{R}$ sends its local residual E_ℓ^μ and the coupling term $A_\ell\, x_\ell$ to the coordinator, which computes the global optimality residual as:

$$E^\mu(x, s, \lambda, \kappa, \gamma) = \max \left\{ \max_{\ell \in \mathcal{R}} \left\{ E_\ell^\mu(x, s, \lambda, \kappa, \gamma) \right\}, \left\| \sum_{\ell \in \mathcal{R}} A_\ell x_\ell - b \right\|_\infty \right\}. \tag{5.12}$$

For the original problem (5.1), setting $\mu = 0$ gives the residual E^0. When the current primal-dual iterate $(x, s, \lambda, \kappa, \gamma)$ satisfy:

$$E^0(x, s, \lambda, \kappa, \gamma) \leq \epsilon, \tag{5.13}$$

where ϵ is a user-defined tolerance, and the slack-dual variables (s_ℓ, κ_ℓ) remain positive at the decoupled solutions (see Remark 5.3) (Remark 5.3), the solution also satisfies the KKT conditions of the original problem (5.2). Therefore, the algorithm terminates when condition (5.13) is met.

To achieve fast local convergence, we adopt the barrier update strategy outlined in [130, Strategy 2] [71], which guarantees superlinear convergence under the second order sufficient condition (SOSC) (Theorem 2.2). Let ν denote the iteration counter for the outer loop, and μ_ν the current barrier parameter. When the primal-dual iterates $(x, s, \lambda, \kappa, \gamma)$ of the barrier problem (5.3) satisfy:

$$E^{\mu_\nu}(x, s, \lambda, \kappa, \gamma) \leq \eta^- \, \mu_\nu \tag{5.14}$$

with $\eta^- = 10$ in our implementation, the barrier parameter is updated as follows:

$$\mu_{\nu+1} = \max \left\{ \frac{\epsilon}{10}, \min \left\{ \frac{\mu_\nu}{5}, \mu_\nu^{1.5} \right\} \right\} \tag{5.15}$$

This approach ensures that the barrier parameter decreases at a superlinear rate, thereby accelerating convergence.

Condensed Coordination Step

The coordination problem (5.7) is one iteration of the second-order multiplier method [70]. Consider applying a Newton step to the primal-dual perturbed KKT conditions (5.4):

$$\begin{bmatrix} \nabla^2_{xx}\mathcal{L} & 0 & J^\top & R^\top & A^\top \\ 0 & \mathcal{K} & 0 & \mathcal{S} & 0 \\ J & 0 & 0 & 0 & 0 \\ R & I & 0 & 0 & 0 \\ A & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta s \\ \Delta\gamma \\ \Delta\kappa \\ \Delta\lambda \end{bmatrix} = - \begin{bmatrix} g \\ \mathcal{K}\, s - \mu\, \mathbb{1} \\ c^{\mathrm{E}}(x) \\ c^{\mathrm{I}}(x) + s \\ Ax - b \end{bmatrix}, \tag{5.16}$$

where the KKT matrix is asymmetric.

Following the discussion in Section 2.3.2, we can obtain a symmetric linear system by eliminating the slack variables Δs and dual variables $\Delta\kappa$. This results in the following coupled system:

$$\begin{bmatrix} H & J^\top & A^\top \\ J & & \\ A & & \end{bmatrix} \begin{pmatrix} \Delta x \\ \Delta\gamma \\ \Delta\lambda \end{pmatrix} = - \begin{pmatrix} g \\ c^{\mathrm{E}}(x) \\ Ax - b \end{pmatrix} \tag{5.17}$$

with

$$\begin{aligned} H &= \mathrm{blkdiag}(H_\ell), \;\; H_\ell \approx \nabla_{xx} L_\ell + R_\ell^\top \mathcal{S}_\ell^{-1} \mathcal{K}_\ell R_\ell, \\ g &= \mathrm{vertcat}(g_\ell), \quad g_\ell = \nabla_x L_\ell + R_\ell^\top \mathcal{S}_\ell^{-1} \left(\mu \, \mathbb{1} + \mathcal{K}_\ell c_\ell^{\mathrm{I}}(x_\ell)\right), \\ J &= \mathrm{blkdiag}(J_\ell), \;\; A = \mathrm{horzcat}(A_\ell). \end{aligned}$$

The equivalent QP subproblem

$$\min_{\Delta x} \sum_{\ell \in \mathcal{R}} \left\{ \frac{1}{2} \Delta x_\ell^\top \; H_\ell \; \Delta x_\ell + g_\ell^\top \Delta x_\ell \right\} \tag{5.18a}$$

$$\text{s.t.} \quad \sum_{\ell \in \mathcal{R}} A_\ell \Delta x_\ell = b - A\, x \quad \mid \lambda + \Delta\lambda \tag{5.18b}$$

$$J_\ell \Delta x_\ell = -c_\ell^{\mathrm{E}}(x_\ell), \;\; \ell \in \mathcal{R} \, . \tag{5.18c}$$

is similar to the coupled QP subproblems (3.5) in the standard ALADIN (Algorithm 3).

Unlike solving the full-dimensional QP subproblems (3.5) in standard ALADIN (Algorithm 3), we further condense the linear systems (5.17) using the Schur complement to reduce both communication effort between agents and coordinator and computation effort in coordination step.

By reordering the elements in the Newton step (5.17), we can rewrite it as:

$$\begin{bmatrix} \bar{H}_1 & & & \bar{A}_1^\top \\ & \ddots & & \vdots \\ & & \bar{H}_{N^{\mathrm{reg}}} & \bar{A}_{N^{\mathrm{reg}}}^\top \\ \bar{A}_1 & \cdots & \bar{A}_{N^{\mathrm{reg}}} & 0 \end{bmatrix} \begin{pmatrix} \Delta\bar{x}_1 \\ \vdots \\ \Delta\bar{x}_{N^{\mathrm{reg}}} \\ \Delta\lambda \end{pmatrix} = - \begin{pmatrix} \bar{g}_1 \\ \vdots \\ \bar{g}_{N^{\mathrm{reg}}} \\ Ax - b \end{pmatrix}, \tag{5.19}$$

where local primal-dual iterates are $\Delta\bar{x}_\ell = \left(\Delta x_\ell^\top, \Delta\gamma_\ell^\top\right)^\top$ and curvature information is given by

$$\bar{H}_\ell = \begin{bmatrix} H_\ell & J_\ell^\top \\ J_\ell & \end{bmatrix}, \ \bar{g}_\ell = \begin{pmatrix} g_\ell \\ c_\ell^{\mathrm{E}}(x) \end{pmatrix} \text{ and } \bar{A}_\ell = \begin{bmatrix} A_\ell & 0 \end{bmatrix} \tag{5.20}$$

This system can be compactly written as:

$$\begin{bmatrix} \bar{H} & \bar{A}^\top \\ \bar{A} & 0 \end{bmatrix} \begin{pmatrix} \Delta\bar{x} \\ \Delta\lambda \end{pmatrix} = - \begin{pmatrix} \bar{g} \\ Ax - b \end{pmatrix} \tag{5.21}$$

with block diagonal matrix $\bar{H} = \mathrm{blkdiag}(\bar{H}_\ell)$. Using the Schur complement, we have

$$W\Delta\lambda = -h$$

with

$$W = \sum_{\ell\in\mathcal{R}} W_\ell, \ h = -b + \sum_{\ell\in\mathcal{R}} h_\ell$$

where the condensed curvature information W_ℓ and g_ℓ can be computed parallelly in each agent $\ell \in \mathcal{R}$:

$$\begin{aligned} W_\ell &= -\bar{A}_\ell \, \bar{H}_\ell^{-1} \bar{A}_\ell^\top \\ h_\ell &= A_\ell x_\ell - \bar{A}_\ell \, \bar{H}_\ell^{-1} \bar{g}_\ell. \end{aligned}$$

Once the coordinator receives respective dual Hessian W_ℓ and dual gradient h_ℓ from all local agent $\ell \in \mathcal{R}$, and a descent direction can be guaranteed, then we solve the coordination problem in dual space (5.7); otherwise, Algorithm 2 is called to correct the inertia. Details about the criterion of descent direction and corresponding modification will be discussed in the following section.

Remark 5.4. *Further condensing the KKT linear systems (5.17) serves to reduce the communication effort between agents and the coordinator. In contrast to the standard ALADIN approach, where local gradients, Jacobians, and Hessians must be sent to the coordinator (as illustrated in Fig. 4.8), only the dual Hessian W_ℓ and dual gradient g_ℓ need to be communicated. This not only reduces the data exchanged between agents and the coordinator but also distributes part of the coordinator's computational tasks to the local agents. By allowing these tasks to be handled in*

parallel by the agents, the overall computational burden on the coordinator is reduced, leading to an improvement in the total computation time.

Remark 5.5. *For solving AC OPF problems using the ALADIN algorithm 3, transferring the local Jacobian matrix J_ℓ is required. This raises concerns because the sparsity pattern of the local Jacobian matrix J_ℓ can reveal the topology of the local power systems. The proposed algorithm solves this issue by condensing derivatives (5.6). The condensed dual Hessian W_ℓ and dual gradient g_ℓ are dense matrices and do not reveal the underlying topology in dual space, since the Lagrangian multiplier λ represents only how the objective value changes with respect to a slight violation of the corresponding consensus constraints $Ax - b$ (Remark 2.2).*

Once the coordination problem (5.7) is solved, the dual step $\Delta\lambda$ is sent back to the local agents, and the full primal-dual step can be recovered locally by (5.8):

$$\begin{aligned}\begin{pmatrix}\Delta x_\ell \\ \Delta\gamma_\ell\end{pmatrix} &= -\bar{H}_\ell^{-1}\bar{A}_\ell^\top \Delta\lambda - \bar{g}_\ell,\\ \Delta s_\ell &= -c_\ell^{\mathrm{I}}(x_\ell) - s - R_\ell\,\Delta x_\ell,\\ \Delta\kappa_\ell &= -\kappa_\ell + \mathcal{S}_\ell^{-1}(\mu\,\mathbb{1} - \mathcal{K}_\ell\Delta s),\end{aligned}$$

Having recovered local variables in each agent, primal and dual steplengths $\beta_\ell^{\mathrm{p}}, \beta_\ell^{\mathrm{d}} \in (0,1]$ are determined by fraction-to-the-boundary rule in each local agent $\ell \in \mathcal{R}$.

The primal-dual iterates are then updated according to (5.10), where the shortest primal step length, i.e., $\beta^{\mathrm{p}} = \min_{\ell\in\mathcal{R}} \beta_\ell^{\mathrm{p}}$, is used. If a globalization strategy is required, the parameters α_1, α_2, α_3 are set as defined by Algorithm 4. Alternatively, for full-step updates, set $\alpha_1 = \alpha_2 = \alpha_3 = 1$.

Additionally, the dual variables $(\gamma_\ell, \kappa_\ell)$ can be updated locally by

$$\begin{aligned}\kappa_\ell^+ &= \kappa_\ell + \alpha_2\,\beta_\ell^{\mathrm{d}}\Delta\kappa_\ell, && \forall \ell \in \mathcal{R}\\ \gamma_\ell^+ &= \gamma_\ell + \alpha_2\,\beta^{\mathrm{p}}\Delta\gamma_\ell, && \forall \ell \in \mathcal{R}\end{aligned}$$

These updates help refine the initial guess for the decoupled NLPs (5.5) in subsequent iterations with additional communication overhead.

5.2.2 Distributed Inertia Correction

From Theorem 2.4, we know that the perturbed Newton-step (5.17) is a strict descent direction of the corresponding QP subproblems (5.18) if

$$\text{inertia}(K) = (N^x,\ N^{\mathrm{E}} + N^{\lambda},\ 0) \tag{5.23}$$

with $N^x = \sum_{\ell \in \mathcal{R}} N_\ell^x$ and $N^{\mathrm{E}} = \sum_{\ell \in \mathcal{R}} N_\ell^{\mathrm{E}}$, where K denote the corresponding perturbed KKT matrix. However, under the proposed distributed framework (Algorithm 10), it is hard to determine the inertia of the perturbed KKT matrix.

To extend the condition (5.23) to condensed linear system (5.7), we rely on the following property of inertia:

Lemma 5.1 (Sylvester's Law of inertia [129]). *Let $K \in \mathbb{R}^{n \times n}$ be a Hermitian matrix. Then there is a nonsingular matrix $G \in \mathbb{R}^{n \times n}$ such that $W = G^\top K G$ if and only if*

$$inertia(K) = inertia(W).$$

This Lemma indicates that the inertia of a Hermitian matrix remains invariant under the specified transformation. As a result, when applying the Schur complement, this property leads to the following key property, which is especially useful under the distributed framework:

Lemma 5.2 (Haynsworth inertia additivity formula [131, 129]). *Given a Hermitian matrix*

$$K = \begin{bmatrix} K_{11} & K_{21}^\top \\ K_{21} & K_{22} \end{bmatrix}$$

and let K_{11} be the nonsingular submatrix of K and let

$$\Lambda = K_{22} - K_{21} K_{11}^{-1} K_{21}^\top$$

be the Schur complement of K_{11} in K. Then

$$inertia(K) = inertia(K_{11}) + inertia(\Lambda).$$

Proof. Lemma 5.2 can be viewed as a direct extension of Lemma 5.1 to Schur complement. To demonstrate this, we assume that

$$G = \begin{bmatrix} I & -K_{11}^{-1}K_{21}^{\top} \\ 0 & I \end{bmatrix},$$

Then, we have

$$G^{\top}KG = \begin{bmatrix} K_{11} & 0 \\ 0 & \Lambda \end{bmatrix}.$$

By Sylvester's Law of inertia (Lemma 5.1), we know that the inertia of the original matrix K remains unchanged after the transformation, i.e.,

$$\text{inertia}(K) = \text{inertia}(G^{\top}KG).$$

Here, $\sigma(A)$ denotes the set of eigenvalues of a given matrix A. Since $\sigma(G^{\top}KG) = \sigma(K_{11}) \bigcup \sigma(\Lambda)$, this implies that

$$\text{inertia}(G^{\top}KG) = \text{inertia}(K_{11}) + \text{inertia}(\Lambda),$$

which concludes the proof. □

Theorem 5.1. *Let $K \in \mathbb{R}^{(N^x+N^E+N^\lambda)\times(N^x+N^E+N^\lambda)}$ be the perturbed KKT matrix in (5.17), i.e.,*

$$K = \begin{bmatrix} H & J^{\top} & A^{\top} \\ J & & \\ A & & \end{bmatrix}$$

and let the Jacobian $\bar{J} = \begin{bmatrix} J^{\top}, A^{\top} \end{bmatrix}^{\top}$ be of full-row rank. Then, if dual Hessian W satisfies the inertia condition

$$\textit{inertia}(W) = (N^x,\ N^E + N^\lambda,\ 0) - \sum_{\ell \in \mathcal{R}} \textit{inertia}(\bar{H}_\ell), \tag{5.24}$$

H is positive definite on the nullspace of the Jacobian $\bar{J}$ and the QP subproblem (5.18) has a strict local minimizer.

Proof. As a a direct result of Lemma 5.2, we have

$$\begin{aligned}\text{inertia}(K) &= \text{inertia}(W) + \text{inertia}(\bar{H}) \\ &\quad \Downarrow \text{ block diagonal structure of the KKT matrix } \bar{H} \text{ in (5.19)} \\ &= \text{inertia}(W) + \sum_{\ell \in \mathcal{R}} \text{inertia}(\bar{H}_\ell).\end{aligned}$$

Then, the conclusion follows from the condition (5.23). □

Algorithm 11 Distributed inertia Correction

Input: δ_x^{last}, inertia(W), inertia$(\bar{H})$, $\forall \ell \in \mathcal{R}$

if the inertia condition (5.24) satisfied **then**
  $\delta_x^{\text{last}} = 0$
else
  if $\delta_x^{\text{last}} = 0$ **then**
    $\eta^{\text{inc}} = \eta^{\text{fast}}$ and $\delta^x = \delta_0^x$
  else
    $\eta^{\text{inc}} = \eta^{\text{slow}}$ and $\delta^x = \max\left(\eta^{\text{red}}\ \delta_0^x,\ \bar{\delta}^x\right)$
  end if
  if $n^0 = 0$ **then**
    $\delta^\gamma = 0$
  else
    $\delta^\gamma = \delta_0^\gamma$
  end if
  while the inertia condition (5.24) not satisfied **do**
    inertia$(\bar{H}_\ell) =$ inertia $\left(\begin{bmatrix} H_\ell + \delta^x I & J_\ell^\top \\ J_\ell & -\delta^\gamma I \end{bmatrix}\right)$
    inertia$(W) =$ inertia $\left(-\sum_{\ell \in \mathcal{R}} \bar{A}_\ell\ \bar{H}_\ell^{-1}\bar{A}_\ell^\top\right)$
    $\delta_x^{\text{last}} = \delta^x$
    $\delta^x = \eta^+ \delta^x$
  end while
end if

Return $\delta^{\text{last}} = \delta^x$

The condition (5.24) can be viewed as a distributed variant of the conventional inertia condition (2.27), where the inertia of local matrix $\bar{H}$ can be evaluated in parallel by each agent when condensing the derivatives (5.6). Based on the centralized version (Algorithm 1), we propose a distributed inertia correction algorithm to validate inertia and modify Hessian when necessary. The distributed inertia correction process is detailed in Algorithm 11, which operates under the assumption that the consensus matrix A is nonsingular. This assumption typically holds for distributed problems, and the problem can be reformulated if it does not.

Similar to Algorithm 1, the Algorithm 11 starts if the distributed inertia conditions (5.24) are not satisfied. If the inertia check fails, the regularization parameter δ^x is initialized based on the previous Newton step, and progressively larger values of the regularization parameter δ^x are applied until the distributed inertia conditions (5.24) are met. This incremental adjustment ensures that the algorithm finds the smallest necessary perturbation to correct the inertia while maintaining the distributed structure of the optimization process.

Remark 5.6. *Although the evaluation of the inertia of local matrix $\bar{H}_\ell$ can be evaluated in parallel by local agents during the condensation of the derivatives (5.6), the condensation process itself is computationally expensive. It is particularly expensive if the local agents are densely coupled and N^λ is relatively large, as noted in Remark 2.8. Furthermore, Algorithm 1 introduces additional iterative communication, which can further increase the overall computational overhead.*

5.3 Convergence Analysis

Since the barrier problem is solved by ALADIN-type algorithm, we need some basic assumptions to ensure its successful application.

Assumption 5.1. *The primal-dual central path $(x^\star(\mu), \lambda^\star(\mu), \kappa^\star(\mu), \gamma^\star(\mu))$ consists of local solutions of the barrier problems (5.3) with respect to barrier parameter $\mu \geq 0$. At each point along this path, the following conditions hold:*

(a) For each region $\ell \in \mathcal{R}$, the local objective f_ℓ and constraint c_ℓ are twice differentiable in the neighbourhood of $x_\ell^\star$ with Lipchitz continuous second derivatives.

(b) The linear independence constraint qualification (Definition 2.3) holds at $x^\star$.

(c) The strict complementarity condition (Definition 2.4) holds at $(x^\star, \lambda^\star, \kappa^\star, \gamma^\star)$.

(d) The second order sufficient condition (Theorem 2.2) holds at $(x^\star, \lambda^\star, \kappa^\star, \gamma^\star)$.

We do not require Assumption 3.2 here because it can only ensure a locally quadratic convergence rate to $x^\star(\mu)$ with a given μ. However, the overall convergence rate to the local solution x^* of the original problem 5.1 is primarily influenced by the strategy used to reduce μ.

Now, we summarize the global convergence of the proposed Algorithm 10:

Theorem 5.2. *Let Assumption 5.1 hold, let penalty parameter ρ be sufficiently large such that*

$$\nabla_{xx} L_\ell + R_\ell^\top \mathcal{S}_\ell^{-1} \mathcal{K}_\ell R_\ell + \rho \Sigma_\ell$$

is positive definite on the null space of J_ℓ for all $\ell \in \mathcal{R}$, and the line-search parameters $\alpha_1,\ \alpha_2,\ \alpha_3$ are determined by Algorithm 4 then Algorithm 10 will terminate after a finite number of iterations.

Proof. The proof of Theorem 5.2 can be established in two steps. First, the barrier problem (5.3) with a fixed parameter μ is treated as a generic equality-constrained nonlinear programming (NLP) problem. By applying the ALADIN algorithm to solve these barrier problems, global convergence to $x^\star(\mu)$ for each fixed barrier parameter μ is guaranteed, as established by Theorem 3.2. Second, the global convergence of the overall method to the solution x^* of the original problem (5.1) is achieved by monotonically decreasing the barrier parameter μ. As μ approaches zeros, $x^\star(\mu)$ approaches the original solution x^*. □

5.4 Practical Considerations for Implementation

In this section, we discuss practical considerations for implementing the proposed Algorithm 10 within a distributed or parallel computing environment. To illustrate the interactions in the distributed process, the process sequence of the proposed Algorithm 10 between the coordinator and local agents is depicted in Fig. 5.1. This figure emphasizes the flow of information and synchronization steps, clearly visualizing the coordination for better understanding and practical deployment.

To compare different networks and partition sizes, we define two measures:

- **Network Density:**

$$\zeta = \frac{N^{\text{line}}}{N^{\text{bus}}} \in \mathbb{R}_{++},$$

  which indicates how densely buses are interconnected. This ratio does not depend on how the network is partitioned.

- **Coupling Density:**

$$\xi_\ell = \frac{N_\ell^{\text{cpl}}}{N_\ell^x} \in [0, 1],$$

  where N_ℓ^{cpl} is the number of coupling variables, and N_ℓ^x is the total number of state variables in that region. The average coupling density across all regions is

$$\bar{\xi} = \frac{1}{N^{\text{reg}}} \sum_{\ell \in \mathcal{R}} \xi_\ell = \frac{1}{N^{\text{reg}}} \sum_{\ell \in \mathcal{R}} \frac{N_\ell^{\text{cpl}}}{N_\ell^x}.$$

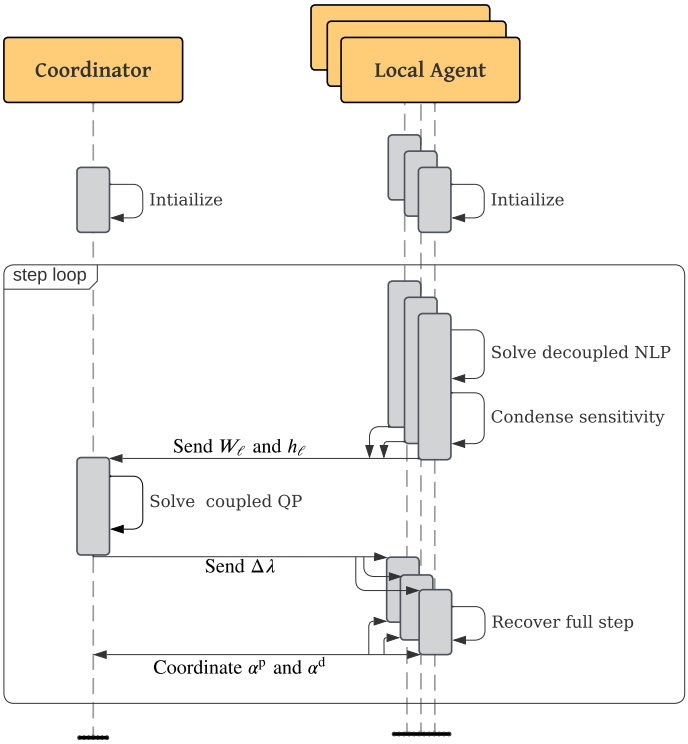


Figure 5.1: Sequence diagram of the proposed distributed optimization algorithm (Algorithm 10)

Remark 5.7. *For distributed optimization problems (5.1), N_ℓ^{cpl} equals the number of nonzero rows in A_ℓ. This indicates how many variables in the region ℓ are shared with neighboring regions.*

5.4.1 Graph-based Decomposition

How the power grid is partitioned is essential for reducing problem complexity and improving parallelization. Traditional grid partitioning methods, such as those based on different system operators, hierarchical clustering, and electrical distances, do not adequately address the needs of distributed optimization and often fail to consider computational coupling, which strongly impacts the performance of distributed optimization methods, as observed by [96].

Recent advancements have introduced more sophisticated techniques. For instance, a spectral method, first proposed in [128], improves convergence using ADMM for large-scale problems [96]. However, the algorithm is initially designed for first-order methods and requires either *a priori knowledge* of the optimal operating point or *an assumption* of minimal deviation in power flow from this point.

By contrast, the proposed Algorithm 10 applies Newton-type methods, i.e., ALADIN, to solve the barrier problems (5.3), leveraging the Schur complement to condense second-order information. Two key factors improve efficiency in a distributed or parallel environment:

- **Equitable Partitioning:** As depicted in Fig.10, the decoupled NLP subproblems (5.5) should be similar in size and computational effort. This prevents idle time and improves synchronization among local agents.
- **Reduced Coupling:** For a given network $\mathcal{S} = (\mathcal{N}, \mathcal{L}, \mathcal{R})$, small coupling density $\xi_\ell \ll 1$ means fewer nonzero rows in both A_ℓ and $\bar{A}_\ell$. This simplifies the computation of matrix products $\bar{H}_\ell^{-1}\bar{A}_\ell^\top$ during the derivative condensing step (5.6), as discussed in Remark 2.8, and leads to smaller coupled QP subproblems (5.7). As a result, both communication and computation demands are reduced.

As demonstrated later in Section 5.5.2, the condensation process and coupled coordination consume significant time for large-scale problems. Therefore, minimizing coupling has a more pronounced impact on the efficiency of the proposed Algorithm 10.

Recent research [132, 133] shows the spectral method is outperformed by Karlsruhe Fast Flow Partitioner (KaFFPa) from the Karlsruhe High Quality Partitioning (KaHIP) [134] when solving AC OPF for small- to medium-size systems using ALADIN. Unlike requiring prior knowledge or assumptions about the operating state, KaFFPa requires only a simple graph of the power system $(\mathcal{N},\ \mathcal{L})$ and produces partitions of equal size with minimal connecting lines.

These two features—fewer connections and balanced partition sizes—align well with the above-mentioned criteria. Therefore, this paper adopts KaFFPa for grid partitioning. For a comprehensive description of the KaFFPa algorithm, please refer to [134, 135].

5.4.2 Distributed Automatic Differentiation

To run optimization algorithms in parallel, some earlier works [136, 137] indicate that full parallelization generally requires parallelizing function evaluations, including the computation of sensitivities or derivatives. The challenge is to evaluate the derivatives while minimizing the communication between the different processes. This has led to the development of different

prototypes for MPI-parallel modelers [138, 139, 140]. Other research adopted approaches mainly from machine learning, using fast automatic differentiation (AD) libraries to efficiently generate derivatives on hardware accelerators such as graphics processing units (GPU) [141, 142, 82, 143].

Unlike state-of-the-art parallel optimization methods, the proposed distributed algorithm for solving NLPs, e.g., AC OPF, partitions the problem at the network level. Therefore, neither the decoupled NLPs (5.5) nor the condensation steps (5.6) require global information from neighboring regions, allowing each local agent $\ell \in \mathcal{R}$ to evaluate the objective function f_ℓ and constraints c_ℓ independently in parallel. This reduces the complexity of communication and coordination using AD, making the parallel execution of the proposed distributed algorithm more straightforward.

5.4.3 Resiliency Against Single-Point Failures

The proposed distributed algorithm offers resiliency against single-point failures. As shown in the sequence diagram (Fig. 5.1), each local agent operates independently during the decoupled step for solving NLP subproblems and condensing sensitivities. If a single local agent fails, the remaining agents can continue operating, and the centralized coordinator can proceed with a reduced dataset. Additionally, if the centralized coordinator fails, one local agent can take over the coordination tasks. This would not significantly reduce computation efficiency since local agents are idle during coordination. Moreover, this does not compromise data privacy since both the dual Hessian W_ℓ and the dual gradient g_ℓ are condensed and data-preserving (Remark 5.5).

Therefore, the proposed distributed algorithm ensures that failures do not lead to catastrophic breakdowns but instead allow for a degraded yet functional solution. By maintaining a distributed operational framework and relying on local computations, the system mitigates the risk associated with any single point of failure, thereby enhancing overall stability and reliability.

5.5 Large-scale Distributed AC OPF: A Case Study

This section demonstrates the performance of Algorithm 10 in solving large-scale AC optimal power flow (OPF) problems under different operating scenarios in a distributed computing environment. We assess the impact of coupling density and validate that this approach outperforms the state-of-the-art centralized nonlinear solver IPOPT [71].

Table 5.3: Comparing Different Region Numbers

Case	N^{bus}	N^{gen}	N^{line}	ζ	**Partition Configuration**					Iter	Init.	**Convergence Performance** (Wall Time [s])					
					$\lvert\mathcal{R}\rvert$	$\bar{\xi}$	N^{conn}	N^{λ}	N_{ℓ}^{x}			dec.	cond.	coord.	rec.	syn.	**total**
case78484	78.8k	6.8k	126.1k	1.607	10	1.65%	395	1432	18062	80	48.40	17.20	188.39	9.27	11.96	4.93	**231.75**
					15	2.45%	587	2110	12070	78	31.64	9.86	99.90	11.36	7.46	6.06	**134.64**
					20	3.28%	777	2852	9184	90	24.73	10.65	80.50	22.47	7.75	9.54	**130.90**
					25	4.11%	990	3600	7530	76	18.70	7.20	81.11	24.88	6.98	11.70	**131.88**
					30	4.17%	1002	3642	6146	87	15.99	15.54	66.08	23.45	6.17	9.92	**121.16**
					35	5.18%	1241	4548	5334	76	12.13	6.44	53.23	28.80	5.58	10.75	**104.80**
					40	5.41%	1299	4756	4734	75	12.70	6.29	37.07	26.20	3.60	10.75	***83.90***
					47	6.37%	1532	5632	4092	76	11.91	5.73	32.77	32.94	3.35	10.93	**85.73**
					50	6.73%	1624	5950	3878	79	10.69	8.60	40.06	38.16	4.90	12.25	**103.97**
					55	7.04%	1690	6232	3614	79	12.12	10.53	59.62	37.92	7.11	13.94	**129.12**
case193k	192.7k	24.6k	228.6k	1.186	10	0.10%	55	220	44610	48	49.22	24.05	25.10	0.19	0.42	0.63	**50.40**
					15	0.22%	128	492	29972	46	34.08	13.32	23.60	0.27	0.58	0.58	**38.34**
					20	0.32%	179	702	22514	46	26.72	10.45	23.02	0.56	0.56	0.81	**35.40**
					25	0.46%	257	1004	17986	34	22.14	7.82	14.07	0.37	0.42	0.67	***23.35***
					30	0.47%	272	1050	15072	35	18.72	7.25	15.13	0.83	0.45	8.80	**32.45**
					35	0.63%	351	1368	12830	44	16.85	16.27	14.00	0.68	0.64	2.58	**34.17**
					40	0.74%	414	1612	11424	49	15.28	11.86	21.81	0.89	1.24	6.28	**42.08**
					47	0.97%	549	2138	9656	50	15.66	20.21	34.31	1.54	2.07	10.46	**68.59**
					50	1.03%	575	2252	9086	48	16.05	22.13	30.38	1.72	1.71	1.71	**57.66**
					55	1.03%	585	2278	8350	47	19.01	19.02	21.44	1.81	1.09	6.51	**49.87**

5.5.1 Configuration & Setup

The framework is built on MATLAB-R2023b. The test cases are the largest test cases from the PGLib-OPF benchmark* [116] with version 23.07 and the large-scale test cases from [144]. The grid is partitioned into a different number of regions by using the KaFFPa algorithm from KaHIP project [134]. The case studies are carried out on a small workstation with two AMD® Epyc 7402 24-core processors, i.e., 48 physical cores in total, and 128 GB installed ram. The CasADi toolbox [111] is used in MATLAB. MA57 [145] is used as the linear solver. The centralized reference solution is obtained by solving the AC OPF problems with IPOPT [71].

To evaluate the performance of our proposed algorithm with minimal communication effort considered, we utilize the single program multiple data (SPMD) paradigm from the MATLAB parallel computing toolbox. This approach facilitates distributed computation by dividing work and data among workers without shared memory. Communication between workers is managed using explicit MPI-style commands, taking into account the potential delays and synchronization requirements. In our setup, one worker functions as the coordinator while the others serve as local agents.

5.5.2 Impact of Network Decomposition

We analyzed how the network decomposition affects performance by using three large power system test cases from the PGLib-OPF benchmark [116] and one additional large test case from [144]. Table 5.3 shows detailed results for case78484 and case193k. In general, changes in the number of partitions have a larger effect on overall computing time rather than on the number of iterations required for convergence.

As illustrated in Fig. 5.2, for a given network, when we increase the number of partitions $|\mathcal{R}|$, each local subproblem becomes smaller, but the average coupling density $\bar{\xi}$ increases. In all four cases, the maximum subproblem size $N^x_{\ell,\max}$ is close to its theoretical lower bound, indicating that the KaFFPa balances the subsystems effectively.

A key difference is how the average coupling density $\bar{\xi}$ changes. In case193k, $\bar{\xi}$ grows faster than in the other three networks. This may be due to the network's topology: unlike the three PGLib-OPF cases with network density $\zeta \approx 1.5$, case193k has a lower network density ($\zeta = 1.186$).

*The PGLib-OPF is built for benchmarking AC OPF algorithms under IEEE PES Task Force. The benchmark is available on GitHub: https://github.com/power-grid-lib/pglib-opf and the baseline results for v23.07 can be found here: https://github.com/power-grid-lib/pglib-opf/blob/master/BASELINE.md

When partitioned into 10 regions, case193k shows an order-of-magnitude lower connectivity than case7848, experiencing faster increases in coupling density $\bar{\xi}$ than those with a dense network.

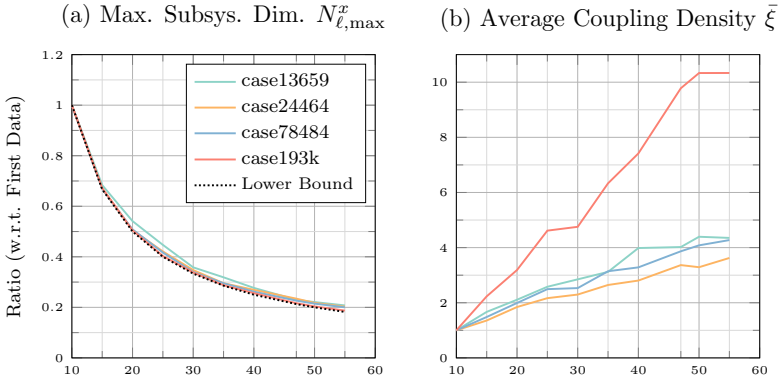


Figure 5.2: Comparison of network decomposition on different power systems, i.e., case13659 from Pan European Grid Advanced Simulation and State Estimation (PEGASE) [146] [147], case24464 from ARPA-E grid optimization competition [148], case78484 from the US Eastern Interconnection states [149] and case193k from [144].

Fig 5.3 shows how the average wall time per iteration is divided among local agents, the coordinator, and synchronization tasks, including communication overhead. Note that the sudden but simultaneous increases in coordinator (Fig. 5.3(b)) and synchronization overhead (Fig. 5.3(d)) stem from additional inertia correction (Algorithm 11), which dampens the total wall time. Except for that inertia correction, when the partition number is less or equal to 47—corresponding to one local agent per core—a higher number of partitions distributes workload to more local agents and shortens their individual computation times. However, having more partitions also makes it more expensive to condense the derivatives (5.6) and enlarge the coordination problems (5.7), leading to higher total wall times.

Remark 5.8. *For a given Network, increasing the partition number $|\mathcal{R}|$ reduces the size of each subproblem and thus lowers each local agent's computation effort. However, it also raises the coupling density ξ_ℓ in each subproblem, making the local derivative condensing—especially computing $\bar{H}_\ell^{-1}\bar{A}_\ell^\top$ in (5.6a)—more expensive, and it also increases the size of the coupled QP subproblems (5.7) at the coordinator. Hence, when hardware resources are not limited, one must balance these trade-offs to find an optimal number of partitions.*

The trade-off is also observed in Fig. 5.3: for the sparser network (case193k), where 25 partitions yield the shortest total time; beyond that, the total time grows. Although theoretically, partitioning the network further can reduce certain computational burdens, real-world performance depends

on how local tasks, network density, and coordination overhead interact. More research and testing are needed to determine the best partitioning strategy in practice.

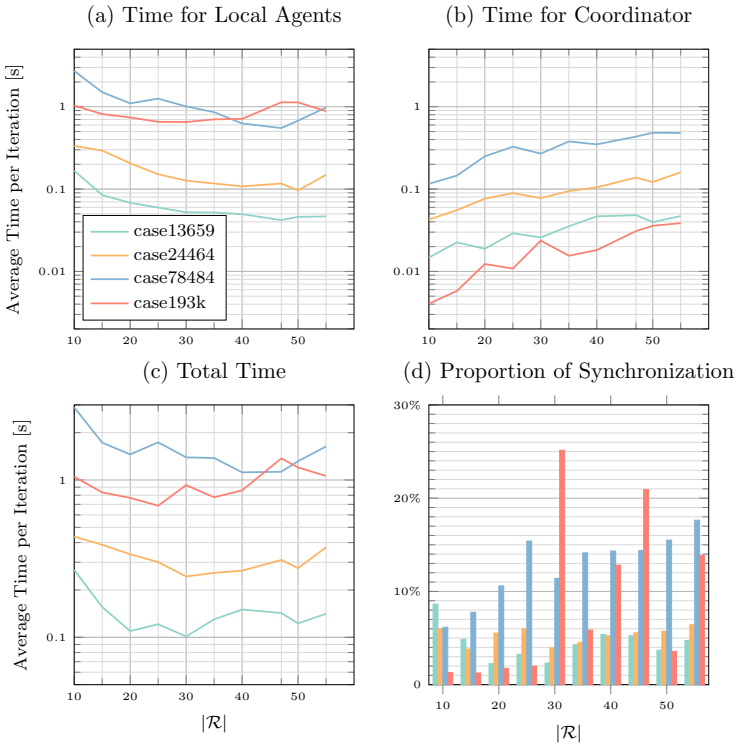


Figure 5.3: Comparison of network decomposition on performance of the proposed Algorithm 10 on large-scale benchmarks

5.5.3 Centralized vs. Distributed

This section evaluates the performance of our proposed distributed approach against the state-of-the-art centralized nonlinear solver, IPOPT [71]. Both centralized and distributed setups utilize MA57 [145] as linear solver and CasADi [111] as automatic differentiation (AD) tools to ensure a fair comparison. The evaluation involves the largest test cases from the PGLib-OPF benchmark [116], each with three operation modes: standard (std), active power increased (api) and small angle difference (sad). Additionally, the four largest test cases from recent studies [144] are included. All these power grids are divided into 40 regions for analysis.

Detailed results in Table 5.4 show that all the proposed distributed approaches converge to local optimizers. The optimality gaps between the centralized and distributed approaches are minimal, typically from 10^{-5} to 10^{-7}. Notably, the active-power increased case78484 under the centralized approach fails to converge, underscoring a potential scalability advantage of distributed approaches. Figure 5.4 compares the performance efficiency across all these large-scale test cases.

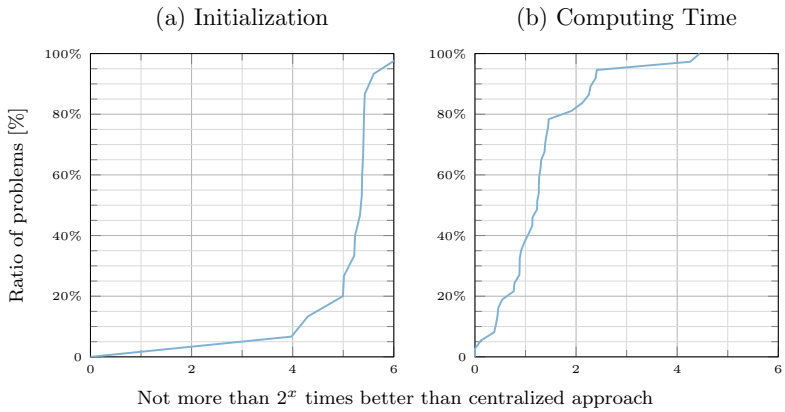


Figure 5.4: Performance profile comparing the proposed Algorithm 10 with IPOPT on large-scale AC OPF benchmarks

The distributed implementation of AD, leveraging the SPMD model, significantly enhances the initialization speed. As shown in Fig. 5.4 (a), initialization is at least 16 times faster in more than 90% of the cases compared to the centralized approach. Regarding algorithm efficiency, the distributed algorithm is at least twice as fast as the centralized method in over 60% of the cases, with around 20% of cases achieving at least four times the speed, as shown in Fig. 5.4 (b). It suggests that the larger the test case, the greater the computational benefits offered by the distributed approach, highlighting its significant advantages for handling large-scale problems.

Table 5.4: Benchmark

Case	N^x	N^c	ζ	$\bar{\xi}$	Mode	**Convergence Performance** (Wall Time [s]) BALADIN Iter	local	coord.	**total**	IPOPT Iter	deriv.	lin.	**total**	**Solution Quality** BALADIN Objective	Violation	IPOPT Objective	Violation
case9241	21.4k	75.9k	1.7367	15.91%	std.	55	2.92	1.55	***6.22***	62	4.28	4.91	**10.68**	6243150	1.5E-08	6243090	3.9E-07
					api.	49	1.61	1.77	***5.09***	101	7.27	9.97	**19.11**	7011200	2.7E-04	7011144	3.1E-07
					sad.	63	2.32	2.45	***7.07***	75	5.51	6.83	**13.94**	6318469	2.9E-06	6318469	3.9E-07
case9591	19.9k	76.5k	1.6594	26.01%	std.	46	1.71	4.54	***8.36***	52	3.60	10.48	**15.76**	1061684	3.1E-06	1061684	9.7E-08
					api.	56	2.19	5.92	***11.54***	103	7.14	20.74	**29.96**	1570274	1.4E-06	1570264	2.2E-07
					sad.	57	2.18	4.90	***9.64***	81	5.60	15.78	**23.25**	1167400	9.2E-06	1167401	9.9E-08
case10000	24k	69.6k	1.3193	16.10%	std.	59	1.91	2.66	***6.08***	78	4.62	6.65	**12.69**	1354031	8.7E-09	1354031	5.1E-07
					api.	61	1.88	2.79	***6.23***	85	5.04	7.21	**13.71**	2678660	4.5E-07	2678659	1.1E-07
					sad.	62	1.81	2.86	***6.17***	93	5.57	9.39	**16.44**	1490210	1.8E-07	1490210	4.8E-07
case10192	21.8k	81.6k	1.6695	23.68%	std.	70	3.53	6.56	***13.37***	50	4.05	8.84	**14.51**	1686923	4.7E-06	1686923	1.3E-07
					api.	57	2.36	5.40	***10.62***	74	5.32	12.56	**19.61**	1977706	1.8E-05	1977686	2.5E-07
					sad.	57	2.44	5.12	***10.09***	66	5.28	11.57	**18.63**	1720194	3.9E-05	1720194	1.3E-07
case10480	22.5k	87.1k	1.7709	26.33%	std.	67	3.87	8.22	***16.21***	63	5.24	14.77	**22.09**	2314648	2.6E-07	2375951	1.1E-07
					api.	65	3.58	9.38	***17.41***	66	5.17	15.46	**22.66**	2863484	1.9E-07	2924781	3.2E-07
					sad.	66	3.90	8.50	***16.73***	64	5.25	15.05	**22.35**	2314712	7.0E-07	2375951	1.1E-07
case13659	35.5k	102.4k	1.4984	9.65%	std.	57	2.50	2.65	***8.65***	180	15.96	26.80	**45.94**	8948202	7.5E-05	8948049	1.9E-07
					api.	72	3.99	3.12	***10.03***	94	8.34	12.97	**23.57**	9385712	4.4E-04	9385711	2.5E-07
					sad.	62	2.74	2.49	***7.88***	302	26.81	118.70	**150.55**	9042199	1.2E-05	9042198	3.2E-07
case19402	40.7k	162.3k	1.7887	19.48%	std.	43	4.67	9.06	***17.66***	69	10.28	33.83	**48.14**	1977815	1.6E-06	1977815	1.2E-07
					api.	64	6.73	15.62	***28.72***	69	10.32	34.53	**48.86**	2583662	1.2E-07	2583663	4.9E-07
					sad.	56	5.87	11.31	***21.93***	78	11.73	37.99	**53.92**	1983808	6.7E-06	1983809	1.2E-07
case20758	45.9k	162.3k	1.6063	13.35%	std.	73	6.99	10.62	**24.79**	43	6.17	12.55	***21.69***	2618662	3.1E-05	2618636	1.4E-07
					api.	63	5.45	7.65	***17.48***	69	9.84	18.90	**32.15**	3126508	5.8E-05	3126508	1.6E-07
					sad.	70	5.91	7.44	***18.20***	55	7.83	15.50	**26.48**	2638220	4.1E-06	2638220	1.4E-07

case24464	52.1k	186.7	1.5458	11.97%	std.	53	5.60	4.97	***13.05***	57	9.48	21.04	**34.17**	2629531	2.1E-05	2629531	6.9E-08
					api.	57	7.86	5.66	***17.04***	75	13.53	29.24	**46.90**	2684051	2.8E-05	2683962	3.2E-07
					sad.	64	6.39	7.22	***18.60***	68	11.40	25.56	**40.83**	2653958	9.9E-06	2653958	7.1E-08
case30000	67.1k	196.2k	1.1798	8.36%	std.	103	10.95	8.68	***27.05***	128	20.85	37.79	**63.38**	1142458	1.6E-04	1142331	4.6E-07
					api.	100	11.91	10.13	***30.80***	147	23.67	45.18	**73.96**	1778059	4.5E-04	1777931	1.8E-07
					sad.	212	21.97	29.77	***91.59***	221	36.44	83.31	**126.23**	1317386	1.3E-04	1309979	2.0E-07
case78484	170.5k	613.5k	1.6057	6.37%	std.	75	47.63	22.74	***80.35***	95	52.10	131.07	**199.05**	15316174	2.1E-05	15315886	1.3E-07
					api.	74	47.21	23.22	***83.01***	217	109.66	1688.24	**1808.58**	16140687	3.3E-07	19379770	1.3E+01
					sad.	77	49.79	24.17	***83.92***	96	53.68	132.43	**201.55**	15316174	1.7E-06	15315886	1.3E-07
case21k	47.6k	113.2k	1.1858	8.41%	std.	29	1.70	1.00	***3.30***	63	5.90	8.22	**16.08**	2592246	8.4E-06	2592098	1.2E-07
case42k	95.1k	226.5k	1.1858	4.94%	std.	42	5.14	1.39	***7.80***	66	12.55	17.53	**34.08**	2592459	2.2E-06	2592458	5.8E-07
case99k	224.2k	522.9k	1.1858	2.11%	std.	45	15.18	0.60	***16.54***	67	30.06	46.45	**86.50**	2594077	2.2E-06	2594077	5.6E-08
case194k	434.8k	1035.5k	1.1857	0.97%	std.	46	31.31	0.69	***37.35***	71	63.30	93.94	**178.33**	2595600	8.0E-07	2595599	4.1E-08

5.6 Iterative Communication Effort Analysis

This section outlines the communication requirements for the proposed Algorithm 10. Building on our earlier work [34], we extend the theoretical analysis of ADMM and ALADIN from AC OPF problems to more general NLP formulations. We also evaluate practical performance using the large-scale benchmarks described in the previous section.

5.6.1 Worst-Case Communication Effort

Under worst-case scenarios, we assume that

- The number of active constraints is equal to the number of state variables N_ℓ^x for each region $\ell \in \mathcal{R}$ such that linear independence constraint qualification (LICQ) (Definition 2.3) still holds,
- All matrices are dense, i.e., sparsity patterns are not considered, and
- Only the full-step is considered; neither the inertia correction (Algorithm 11) nor the globalization (Algorithm 4), which require additional communication, are covered in this analysis.

Communication for ADMM

In this section, we analyze the communication effort for the ADMM algorithm outlined in Algorithm 2. Local agents execute the first two steps in parallel, while the coupled averaging step (see Eq.(2.45)) requires the coordinator to collect primal and dual iterates (x^+, λ^+). The distributed formulation (Eq.(2.42)) introduces global variables z and local variable x with the consensus constraint (Eq. (2.42b)) ensuring that

$$x_\ell - z_\ell = 0 \quad | \ \lambda_\ell, \ \forall \ell \in \mathcal{R}$$

. Therefore, the forward communication, in terms of floating-point numbers, consists of:

$$\sum_{\ell \in \mathcal{R}} 2N_\ell^x.$$

Once the coordination task (Eq. (2.45)) is completed, only the primal variables are broadcasted back to the corresponding local agents. The resulting backward communication is:

$$\sum_{\ell \in \mathcal{R}} N_\ell^x$$

Communication for ALADIN

Considering that ALADIN solves a coupled linear KKT system (3.8) in coordination step:

$$\begin{bmatrix} H & A^\top & J^\top \\ A & -\frac{1}{\delta} I & 0 \\ J & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \lambda \\ \kappa^+ \end{bmatrix} = - \begin{bmatrix} g + \lambda^\top A \\ Ax - b \\ 0 \end{bmatrix}$$

where the local gradient $g_\ell \in \mathbb{R}^{N_\ell^x}$, Jacobian $J_\ell \in \mathbb{R}^{N_\ell^x \times N_\ell^x}$ and Hessian $H_\ell \in \mathbb{R}^{N_\ell^x \times N_\ell^x}$ are evaluated by each local agent $\ell \in \mathcal{R}$ and sent to the centralized coordinator later. Additionally, $A_\ell x_\ell$ is also required to evaluate the primal residual $Ax - b$. Considering the symmetricity of the Hessian H_ℓ, the forward communication in terms of floating-point numbers includes

$$\sum_{\ell \in \mathcal{R}} \underbrace{N_\ell^x}_{g_\ell} + \underbrace{(N_\ell^x)^2}_{J_\ell} + \underbrace{\frac{N_\ell^x (N_\ell^x + 1)}{2}}_{H_\ell} + \underbrace{N_\ell^{\text{cpl}}}_{A_\ell x_\ell}.$$

Once the linear system (3.8) solved, the coordinator sends back the primal-dual steps $(\Delta x, \Delta \lambda)$ to each local agent $\ell \in \mathcal{R}$, with the backward communication effort being:

$$\sum_{\ell \in \mathcal{R}} \underbrace{N_\ell^x}_{\Delta x_\ell} + \underbrace{N_\ell^{\text{cpl}}}_{\Delta \lambda}.$$

Communication for DALIPoint

For the proposed distributed Algorithm 10, the coordinator solved the condensed QP subproblem (5.7):

$$W \Delta \lambda = -h$$

with $W = \sum_{\ell \in \mathcal{R}} W_\ell$ and $h = -b + \sum_{\ell \in \mathcal{R}} h_\ell$. The communication effort here is primarily associated with transferring the dual Hessian W_ℓ and the dual gradient g_ℓ. Note that the coordinator

is also responsible for reducing the barrier parameter μ (Step 5) before solving the condensed QP. Since h_ℓ can be viewed as linear mapping of μ, i.e.,

$$h_\ell = h_{\ell,0} + \mu\, h_{\ell,\mu}$$

with

$$h_{\ell,0} = -\bar{A}_\ell \bar{H}_\ell^{-1} \begin{pmatrix} \nabla_x L_\ell + R_\ell^\top \mathcal{S}_\ell^{-1} \mathcal{K}_\ell c_\ell^{\mathrm{I}}(x_\ell) \\ c_\ell^{\mathrm{E}}(x_\ell) \end{pmatrix} + A_\ell x_\ell$$

$$h_{\ell,\mu} = -\bar{A}_\ell \bar{H}_\ell^{-1} \begin{pmatrix} \mu\, R_\ell^\top \mathcal{S}_\ell^{-1} e_\ell \\ 0 \end{pmatrix},$$

the communication effort on h_ℓ would be doubled. Similar to the discussion earlier in this section, only N_ℓ^{clp} elements from h_ℓ should be transferred to a centralized coordinator.

As a result, the forward communication in terms of floating-point numbers includes

$$\sum_{\ell \in \mathcal{R}} \underbrace{2N_\ell^{\mathrm{cpl}}}_{h_\ell} + \underbrace{\frac{N_\ell^{\mathrm{cpl}}(N_\ell^{\mathrm{cpl}}+1)}{2}}_{W_\ell}.$$

Unlike standard ALADIN, the proposed algorithm only requires the dual step $\Delta\lambda$ to be communicated during the backward phase, along with additional data for synchronizing the primal-dual steplength:

$$\underbrace{1N^{\mathrm{reg}}}_{\text{Eq. (5.10d)}} + \sum_{\ell \in \mathcal{R}} \underbrace{N_\ell^{\mathrm{cpl}}}_{\Delta\lambda}.$$

5.6.2 Theoretical and Practical Comparison

Recall that $\xi_\ell = \frac{N_\ell^{\mathrm{cpl}}}{N_\ell^x}$ measures the fraction of coupling variables in region $\ell \in \mathcal{R}$, while $\bar{\xi} = \sum_{\ell \in \mathcal{R}} \frac{\xi_\ell}{N^{\mathrm{reg}}}$ is the average coupling density across all regions.

In many real-world scenarios, the number of regions N^{reg} is much smaller than the total number of state variables $\sum_{\ell \in \mathcal{R}} N_\ell^x$, i.e.,

$$N^{\mathrm{reg}} \ll \sum_{\ell \in \mathcal{R}} N_\ell^x.$$

Under this condition, the proposed algorithm demonstrates a significant reduction in communication effort compared to standard ALADIN, even in tightly coupled systems.

In practical optimization problems, the Hessian H and the Jacobian J are often sparse matrices. Therefore, assuming that all matrices are dense makes theoretical analysis easier but does not provide a fair comparison for standard ALADIN, which solves a full-dimensional coupled QP subproblem in this form (5.17).

To provide a fairer comparison, we conduct large-scale tests on AC OPF problems that reflect real-world conditions. We measure communication demands in single-precision floats and summarize the results in Table 5.5.

Table 5.5: Theoretical & Practical Iterative Communication Effort

	$\bar{\xi}$	ADMM	ALADIN	BALADIN
Forward	$\sum_{\ell\in\mathcal{R}} \frac{\xi_\ell}{N^{\text{reg}}}$	$\sum_{\ell\in\mathcal{R}} 2N_\ell^x$	$\sum_{\ell\in\mathcal{R}} (N_\ell^x)^2 + \frac{3+2\xi_\ell}{2} N_\ell^x$	$\sum_{\ell\in\mathcal{R}} \frac{\xi_\ell^2}{2}(N_\ell^x)^2 + \frac{5\xi_\ell}{2} N_\ell^x$
Backward	$\sum_{\ell\in\mathcal{R}} \frac{\xi_\ell}{N^{\text{reg}}}$	$\sum_{\ell\in\mathcal{R}} N_\ell^x$	$\sum_{\ell\in\mathcal{R}} (1+\xi_\ell) N_\ell^x$	$2N^{\text{reg}} + \sum_{\ell\in\mathcal{R}} \xi_\ell N_\ell^x$
Total	$\sum_{\ell\in\mathcal{R}} \frac{\xi_\ell}{N^{\text{reg}}}$	$\sum_{\ell\in\mathcal{R}} 3N_\ell^x$	$\sum_{\ell\in\mathcal{R}} (N_\ell^x)^2 + \frac{5+4\xi_\ell}{2} N_\ell^x$	$2N^{\text{reg}} + \sum_{\ell\in\mathcal{R}} \frac{\xi_\ell^2}{2}(N_\ell^x)^2 + \frac{7\xi_\ell}{2} N_\ell^x$
case13659	9.64 %	0.427 MB	2.388 MB	0.682 MB
case24464	11.97 %	0.635 MB	4.276 MB	2.104 MB
case78484	6.37 %	2.016 MB	13.731 MB	5.791 MB
case194k	0.97%	4.999 MB	20.226 MB	0.888 MB

Note: Theoretical worst-case communication efforts are expressed in terms of floating-point numbers, while practical communication efforts for large-scale AC OPF benchmarks are measured in MegaBytes, assuming the use of single-precision floating-point

Except for the final test case, the communication effort scales linearly for ADMM and quadratically for ALADIN as system size increases. In contrast, the proposed BALADIN exhibits intermediate scaling behavior. Additionally, BALADIN communication efficiency is significantly influenced by the overall coupling density, $\bar{\xi}$, and can outperform ADMM in scenarios where each subproblem is more loosely coupled.

Our proposed distributed optimization algorithm reduces communication efforts compared to standard ALADIN and can even match ADMM under certain conditions. The communication effort and the total computational time demonstrated in Section 5.5 depend partly on how the problem is decomposed and coupled.

5.7 Summary

In this chapter, a new distributed optimization algorithm combines ALADIN with the ADMM to address large-scale power system optimization problems. The approach focuses on improving scalability and communication efficiency. It can also be applied to other optimization and control problems in power system operations, as they involve repetitive expressions for each component type. Extensive simulations, based on the most significant test cases under various operation scenarios, are used for analysis of the impact of coupling density as well as to demonstrate its performance surpasses the state-of-art centralized nonlinear solvers IPOPT [71], even on an inexpensive workstation. Additionally, iterative communication efforts are provided under the worst-case scenarios, as well as using large-scale AC OPF benchmarks for a fair comparison in practice.

6 Distributed Coordination Framework for Integrated Transmission and Distribution Systems

Unlike the previous chapters, which focused solely on iterative distributed optimization algorithms, this chapter explores a scalable coordination framework by integrating distributed optimization with system model simplifications for low-level agents. Specifically, it introduces a real-time distributed coordination framework that combines iterative distributed optimization—relying on iterative data exchanges to find minimizers—with non-iterative flexibility aggregation methods to coordinate integrated transmission-distribution (ITD) systems.

In the proposed framework, the distribution system operator (DSO) calculates and provides its own aggregated flexibility to the transmission system operator (TSO), where the flexibility is represented as power-energy envelopes. At the transmission system level, a distributed nonlinear model predictive control (NMPC) approach manages the economic dispatch of multiple TSOs, incorporating the aggregated flexibility of numerous DSOs. Through spatial and temporal decomposition, each subsystem represents a single time-period AC OPF for one TSO.

Instead of using methods such as the active-set method or the smoothing technique to solve coupled equality-constrained QP subproblems, the proposed algorithm solves coupled inequality-constrained QP subproblems under the ALADIN framework, enhancing scalability at the cost of increased computation time.

Outline. The chapter begins with an introduction to the methodology in Section 6.1. Section 6.2 presents the distribution model and flexibility aggregation method, while the nonlinear economic dispatch model at the transmission level is detailed in Section 6.3. Section 6.4 covers distributed optimization with a receding horizon for dispatch decision-making through spatial and temporal decomposition. Finally, simulations based on a summer day in Germany with

a significant prediction mismatch are presented in Section 6.5, and the chapter concludes with Section 6.6. This chapter is based on [42].

6.1 Introduction

With the rapid adoption of distributed energy resources (DERs) in distribution systems, the aggregated flexibility of all these controllable devices can play an important role in dispatch problems in transmission systems. It can improve the operational efficiency of the overall power grid and enhance reliability when integrating increased levels of renewable energy resources [150]. Hence, coordinating ITD systems becomes essential for efficiently operating future power systems [1, 36].

Multiperiod dispatch problems for ITD systems usually couple individual steady-state optimal power flow (OPF) optimization problems over multiple time periods [151, 152, 153]. The coupling constraints include the generator ramping limits, the model of distributed energy storage systems (ESS), and other time-dependent constraints to consider the controllable devices with time-variant properties. However, it is still a challenge to solve a multiperiod AC optimal power flow (MPOPF) for ITD systems. On the one hand, the AC OPF is generally NP-hard [25], and the complexity of solving an MPOPF is further magnified by the intercoupling of subsequent time periods [152]. On the other hand, collecting necessary and realistic data from multiple stakeholders (i.e., TSOs and DSOs), including grid topology, load profiles, and other sensitive information regarding consumer behaviors, is either not preferred or restricted by regulations [36]. To address these challenges and achieve an efficient operation of the overall ITD systems, recent research analyzed the determination of the aggregated flexibility of the controllable devices in distribution networks [154, 155]. The flexible dispatch region of a distribution network is summarized in a time-coupled power-energy band, taking into account the network topology [156] and operational constraints. However, the proposed ITD framework does not consider the coordination between multiple TSOs in a data-preserving manner, and the proposed inner approximation is computationally inefficient, requiring solving multiple mixed-integer linear programming (MILP) problems iteratively.

To enable privacy preservation and improve computational efficiency, distributed operation frameworks enable TSOs and DSOs to operate independently and collaborate effectively by sharing limited information with a subset of other operators [4, 96, 157, 14, 158]. These proposed distributed frameworks can maintain data privacy and decision-making independence and are based on distributed AC OPF [29, 96] and MPOPF with receding horizon [159]. In addition to the aforementioned distributed algorithms, ALADIN is proposed for generic nonconvex optimization problems with convergence guarantees in [33]. ALADIN-type algorithms have been successfully applied to solve the single period AC OPF for heterogeneous power systems by a single-machine

numerical simulation [34, 101, 158], as well as in a geographically distributed environment [40]. However, these aforementioned studies either lack a convergence guarantee or their scalability is limited by the computational complexity, which so far hinders an application to MPOPF in ITD systems.

In this chapter, we propose a novel real-time framework that combines flexibility aggregation methods and distributed optimization for coordinating the economic dispatch problem integrated transmission-distribution (ITD) systems. At the distribution level, the distribution system operator (DSO) calculates the feasible dispatch region of controllable devices using the LinDistFlow model, which is then communicated to the transmission system operator (TSO). The TSO solves the coordinated economic dispatch problem by considering the aggregated flexibility of distribution systems, utilizing a distributed approach. Unlike previous methods, we develop an ALADIN-type distributed nonlinear model predictive control (NMPC) approach to handle the multiperiod coordination of ITD systems with receding horizons, effectively distributing computational complexity while maintaining data privacy. Simulations using real-world data from a summer day in Germany highlight the approach's scalability and efficiency, demonstrating its practical relevance for TSO-DSO coordination for real-time operation.

6.2 Model of Flexibility in Distribution Systems

In this section, we consider a radial distribution system denoted by a directed tree graph $\mathcal{G}(\mathcal{N}, \mathcal{L})$, where $\mathcal{N} = \{1, ..., N_{\text{bus}}\}$ is the set of buses. The set $\mathcal{L} \subseteq \mathcal{N} \times \mathcal{N}$ collects "links" or "lines" for all $(i, j) \in \mathcal{L}$. The number of links in a distribution network is N_{line}. Bus 1 is the slack (root) bus and is assumed to have a fixed voltage. We also assume that the distribution systems have a pure tree topology, i.e., $N_{\text{bus}} = N_{\text{line}} + 1$ holds.

We leverage the definition of connectivity matrices C^g, C^s and C^{pcc} with respect to generator, ESS and the point of common coupling (PCC) between transmission and distribution, as defined in [112].

Definition 6.1 [160]. *Let $C^{inc} \in \mathbb{R}^{N_{line} \times N_{bus}}$ be the incidence matrix of a given radial network; we set $[C^{inc}]_{\alpha i} = +1$ if bus i is the head of branch α and $[C^{inc}]_{\alpha i} = -1$ if bus i is the tail of the branch α.*

For details about incidence matrices, see [112, 152].

6.2.1 Model of Distribution Systems

We use the LinDistFlow model [156] to describe the relationship between the voltages and net loads in distribution systems by the following linear power flow equation:

$$1 = e_1^\top U_k, \tag{6.1a}$$
$$0 = C^{\text{inc}} U_k - 2RP_k^l - 2XQ_k^l, \tag{6.1b}$$
$$0 = e_1 p_k^{\text{pcc}} - P_k^d - (C^{\text{inc}})^\top P_k^l - C^s P_k^s, \tag{6.1c}$$
$$0 = e_1 q_k^{\text{pcc}} - Q_k^d - (C^{\text{inc}})^\top Q_k^l, \tag{6.1d}$$
$$\underline{U} \leq U_k \leq \overline{U}, \tag{6.1e}$$
$$\underline{P}^s \leq P_k^s \leq \overline{P}^s, \tag{6.1f}$$

where $e_1 = [1, 0, \cdots, 0]^\top \in \mathbb{R}^{N_{\text{bus}}}$, $R = \textbf{diag}(\text{r})$, $X = \textbf{diag}(\text{x})$. $\mathbf{r}, \mathbf{x} \in \mathbb{R}^{N_{\text{line}}}$ denote the resistance and reactance vectors respectively. U_k denotes the vector of squared voltage magnitude at the time instant k, p_k^{pcc}, q_k^{pcc} denote active and reactive power exchanges with the transmission system at the PCC of the distribution system. We use vectors P_k^d, Q_k^d to denote the active and reactive nodal power consumptions, P_k^l, Q_k^l to denote the active and reactive branch flows, and P_k^s to denote the nodal consumptions by distributed energy storage systems (ESS) at time period k. Moreover, (6.1a) fixes the voltage magnitude at the slack bus. Equations (6.1b)-(6.1d) are the LinDistFlow constraints. Upper and lower bounds (6.1e) (6.1f) restrict the voltage magnitude at each bus and the charging/discharging power of ESSs. We rewrite the above power flow equations (6.1a)-(6.1d) in a compact form:

$$M\chi_k + BP_k^s + D_k = 0, \tag{6.2}$$

where

$$M = \begin{bmatrix} e_1^\top & 0 & 0 & 0 & 0 \\ C^{\text{inc}} & -2R & -2X & 0 & 0 \\ 0 & -(C^{\text{inc}})^\top & 0 & e_1 & 0 \\ 0 & 0 & -(C^{\text{inc}})^\top & 0 & e_1 \end{bmatrix},$$

$$\chi_k = \begin{bmatrix} U \\ P^l \\ Q^l \\ p^{\text{pcc}} \\ q^{\text{pcc}} \end{bmatrix}, \; B = -\begin{bmatrix} 0 \\ 0 \\ C^s \\ 0 \end{bmatrix}, \text{ and } D_k = \begin{bmatrix} e_1 \\ 0 \\ P_k^d \\ Q_k^d \end{bmatrix}.$$

Note that $M \in \mathbb{R}^{(2N_{\text{bus}}+N_{\text{line}}+1)\times(N_{\text{bus}}+2N_{\text{line}}+2)}$ is a square matrix since for radial distribution grids, $N_{\text{bus}} = N_{\text{line}} + 1$. In (6.2), M and B remain time-invariant. All dependent variables χ_k are influenced by controllable power injections P_k^s from ESSs, as well as the load demands P_k^d and Q_k^d at each time period k. Therefore, in this chapter, the flexibility in distribution systems primarily arises from the integration of ESSs.

In [44], it is shown that the squared voltage magnitude U can be explicitly expressed by the active and reactive power injections. However, positive definiteness of resistance and reactance for all the branches is required, a condition not universally met in practical power system datasets, as discussed in [45]. To extend the applicability of the proposed coordination framework to a broader range of power systems, we generalize the result from [44]. With the assistance of graph theory, we rigorously demonstrate the invertibility of matrix M, affirming that all state variables, including squared voltage magnitude U, can be explicitly expressed in terms of controllable power injections for all radial networks. This expansion significantly enhances the robustness and versatility of the proposed framework for practical power systems.

Lemma 6.1. *For a given radial network denoted by $\mathcal{G}(\mathcal{N}, \mathcal{L})$, let bus i be a leaf of graph $\mathcal{G}$, let branch α be an edge connected to leaf bus β, then there is only one nonzero element in the β-th column of incidence matrix $C^{inc}(\mathcal{G})$, and it is located in the α-th row.*

This lemma follows directly from the fact that a leaf has only one parent in a radial network.

Lemma 6.2 [160]**.** *A radial network with at least two buses has at least two leaves. Deleting a leaf from a radial network with N buses produces a radial network with $N-1$ buses.*

Proposition 6.1. *Given a radial network $\mathcal{G}$, matrix M is invertible.*

Proof. Considering a non-slack leaf bus β, (α, β) is the only nonzero element in β-th column in matrix C^{inc} due to the incidence matrix property in Lemma 6.1. Hence, $(\alpha + 1, \beta)$ is the only nonzero element in β-th column in matrix M. Similarly, $(\beta + N^{\text{bus}}, \beta + N^{\text{bus}})$ and $(\beta + 2N^{\text{bus}}, \beta + 2N^{\text{bus}})$ are the only nonzero elements in the $(\beta + N^{\text{bus}})$-th and the $(\beta + 2N^{\text{bus}})$-th row respectively.

By eliminating the leaf bus β of the network $\mathcal{G}$, we obtain a reduced radial network $\mathcal{G}^{(1)}$. The resulting matrix $M^{(1)}$ can be viewed as a submatrix of M by removing the set of row $\{\alpha + 1, \beta + N^{\text{bus}}, \beta + 2N^{\text{bus}}\}$ and the set of column $\{\beta, \alpha + N^{\text{bus}}, \alpha + 2N^{\text{bus}}\}$.

Since the nonzero elements in the incidence matrix $C^{\text{inc}}(\mathcal{G})$ is $\{-1,\ 1\}$, the determinant of matrix M can be written as

$$|\det(M)| = \left|\det(M^{(1)})\right| \tag{6.3}$$

with the assistance of cofactor expansions.

By further removing non-slack leaves of the resulting reduced radial networks, we have

$$\begin{aligned} |\det(M)| &= \left|\det(M^{(1)})\right| = \cdots = \left|\det(M^{(N^{\text{bus}}-1)})\right| \\ &= \det\left(\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\right) = 1. \end{aligned} \tag{6.4}$$

Therefore, M is invertible for the given radial network. □

As a result of the generalized proposition 6.1, for a given distribution grid, all the dependent variables in χ_k can be expressed explicitly in terms of the controllable power injections P_k^s, and thus, the exact feasible set is convex and can be written in a convex polytope with respect to P_k^s,

$$\mathcal{P}_k^s = \{P_k^s \in \mathbb{R}^e | A^s P_k^s \leq b^s\}, \tag{6.5}$$

where e denotes the number of ESSs. In the example illustrated in Fig. 6.1, the blue polytope represents an exact feasible set constrained by upper and lower voltage bounds along with power limits of ESSs (6.1f).

6.2.2 Flexibility Aggregation with Tutorial Example

In this chapter, the flexibility of distribution systems primarily arises from the integration of ESS in the LinDistFlow model (6.1). Instead of applying the exact feasible set (6.5), we replace the complex polytope with a strictly inner hyperbox approximation, enhancing computational efficiency while maintaining safe operation guarantees within the system, i.e.,

$$\mathcal{B}_k^s \subseteq \mathcal{P}_k^s, \quad \forall k \in \{1, 2, \cdots, N_k\}, \tag{6.6}$$

where the hyperbox $\mathcal{B}_k$ is defined as

$$\mathcal{B}_k^s(\underline{P}_k^{\text{appr}}, \overline{P}_k^{\text{appr}}) = \{P_k^s \in \mathbb{R}^e | \underline{P}_k^{\text{appr}} \leq P_k^s \leq \overline{P}_k^{\text{appr}}\}, \tag{6.7}$$

Note that $\overline{P}_k^{\text{appr}}$ and $\underline{P}_k^{\text{appr}}$ are upper and lower bounds of the inner hyperbox approximation.

In the tutorial example illustrated in Fig. 6.1, the 2-dimensional green box in Fig. 6.1 represents an inner hyperbox approximation to the exact feasible set (blue polytope). To maximize the performance of the resulting ESS system, we adopt the so-called *maximum volume inner hyperbox* [161]. The hyperbox (6.7) can be written as $\mathcal{B}_k^s(\zeta, \zeta + \xi)$ and the inner approximation can be obtained by solving the following optimization problem:

$$\max_{\xi,\zeta} \quad \sum_{i \in \mathcal{E}} \ln \zeta_i, \tag{6.8a}$$

$$\text{s.t.} \quad A^s \xi + A^{s+} \zeta \leq b^s, \tag{6.8b}$$

where A^{s+} is the positive part of A^s and $\mathcal{E} = \{1, \cdots, e\}$ is the set of ESSs. However, in practice, it can occur that the standby mode of a ESS is excluded by the inner approximation, i.e.,

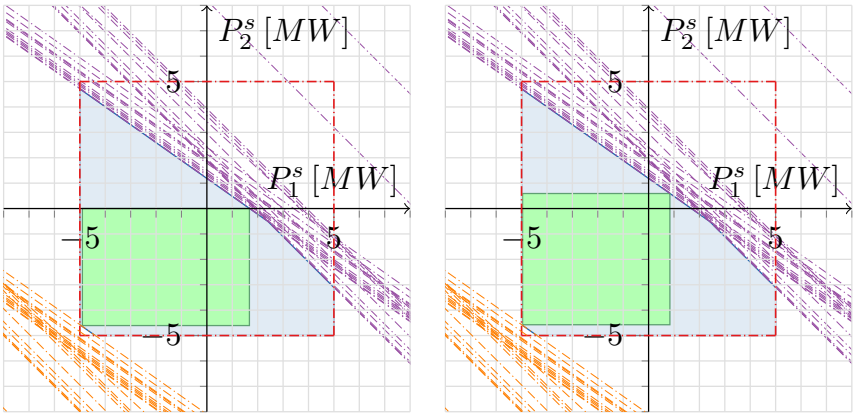
$$\exists i \in \mathcal{E}, \quad [\overline{P}_k^{\text{appr}}]_i < 0 \quad \text{or} \quad [\underline{P}_k^{\text{appr}}]_i > 0,$$

i.e., the origin is not included in the resulting hyperbox (green), as shown in Fig. 6.1 (a).

To address this issue, instead of focusing on maximizing the volume in $\mathbb{R}^e$ space, i.e., finding an equilibrium where ESSs have wide ranges of permissible *power output intervals*, we propose to maximize the volume within the $\mathbb{R}^{2d}$ space, thereby expanding both *charging and discharging power limits* of the ESSs according to

$$\max_{\xi,\zeta} \quad \sum_{i \in \mathcal{E}} \ln(\xi_i + \zeta_i) + \ln(-\xi_i), \tag{6.9a}$$

$$\text{s.t.} \quad A^s \xi + A^{s+} \zeta \leq b^s, \tag{6.9b}$$



(a) $\mathcal{B}_k^s$ calculated by (6.8)

(b) $\mathcal{B}_k^s$ calculated by (6.9)

Figure 6.1: Comparison of inner approximation methods with 2 ESSs located in the heavily loaded IEEE 33-bus system. The orange and the purple lines show the upper and lower bounds on the squares of voltage magnitudes (6.1e). The red lines show the limits on ESSs' power output (6.1f). The blue polytope denotes the exact feasible set (6.5) and the green rectangle denotes the inner hyperbox approximation (6.7).

enabling scenarios where, for instance, both ESSs can charge even during periods of high system load, as illustrated in Fig. 6.1(b), but more importantly, the origin is included in the hyperbox (green).

Since LinDistFlow (6.1) ignores the power losses along branches, power exchanged with the transmission can be expressed with the assistance of conservation of power for all time periods $k \in \mathcal{K}$,

$$p_k^{\text{pcc}} = \mathbb{1}^\top P_k^d + \mathbb{1}^\top P_k^s, \quad P_k^s \in \mathcal{B}_k^s, \tag{6.10a}$$

$$q_k^{\text{pcc}} = \mathbb{1}^\top Q_k^d, \tag{6.10b}$$

where $\mathcal{B}_k^s$ is calculated by applying (6.9). As a result, at the transmission level, a distribution system can be modeled as a load and multiple ESSs at the PCC.

6.3 Economic Dispatch in Transmission Systems

In this chapter, the economic dispatch problems among TSOs are formulated as multiperiod AC optimal power flow (OPF) in the bus injection model [110] with complex voltages expressed in polar coordinates. Here, Θ_k, V_k stack nodal voltage angles $\theta_{k,i}$ and magnitudes $v_{k,i}$ for each bus i at time period k respectively. P_k^{pcc}, Q_k^{pcc} stack active and reactive power exchanges (6.10) for all distribution systems, respectively. $Y = G + \mathbf{j}B$ denote the complex bus admittance matrix, where

$\mathbf{j} = \sqrt{-1}$ and $G, B \in \mathbb{R}^{N_{\text{bus}} \times N_{\text{bus}}}$. The resulting multiperiod AC optimal power flow (MPOPF) for coordinating different TSOs can be written as

$$\min \quad \sum_{k=1}^{N_k} (P_k^g)^\top \mathbf{diag}(a_2) P_k^g + a_1^\top P_k^g + a_0^\top \mathbb{1}, \tag{6.11a}$$

subject to $\forall k \in \mathcal{K} := \{1, 2, \cdots, N_k\}$

$$P_k^b(\Theta_k, V_k) = C^g P_k^g - P_k^d - C^{\text{pcc}} P_k^{\text{pcc}} - C^s P_k^s, \tag{6.11b}$$
$$Q_k^b(\Theta_k, V_k) = C^g Q_k^g - Q_k^d - C^{\text{pcc}} Q_k^{\text{pcc}}, \tag{6.11c}$$
$$\left| S_k^l(\Theta_k, V_k) \right| \le \overline{S}^l, \tag{6.11d}$$
$$\underline{V} \le V_k \le \overline{V}, \ \underline{P}^g \le P_k^g \le \overline{P}^g, \ \underline{Q}^g \le Q_k^g \le \overline{Q}^g, \tag{6.11e}$$
$$\mathrm{e}_k = \mathrm{e}_{k-1} + \Delta t \cdot P_k^s \ \text{ with initial state } E_0 = E(t), \tag{6.11f}$$
$$P_k^g = P_{k-1}^g + \Delta P_k^g \ \text{ with initial state } P_0^g = P_0^g(t), \tag{6.11g}$$
$$\underline{E} \le E_k \le \overline{E}, \ \underline{P}_k^s \le P_k^s \le \overline{P}_k^s, \ \underline{R} \le \Delta P_k^g \le \overline{R}, \tag{6.11h}$$

where $P_k^b, Q_k^b : \mathbb{R}^{N_{\text{bus}}} \times \mathbb{R}^{N_{\text{bus}}} \mapsto \mathbb{R}^{N_{\text{bus}}}$ represent the vector functions of active and reactive power injections for all buses at time period k, and the corresponding i-th element can be expressed as

$$\left[P_k^b\right]_i = v_{k,i} \sum_{j \in \mathcal{N}} v_{k,j} \left(G_{ij} \cos \theta_{k,ij} + B_{ij} \sin \theta_{k,ik} \right),$$
$$\left[Q_k^b\right]_i = v_{k,i} \sum_{j \in \mathcal{N}} v_{k,j} \left(G_{ij} \sin \theta_{k,ij} - B_{ij} \cos \theta_{k,ij} \right),$$

with angle difference $\theta_{k,ij} = \theta_{k,i} - \theta_{k,j}$. Similarly, S_k^l are nonlinear mappings $\mathbb{R}^{N_{\text{bus}}} \times \mathbb{R}^{N_{\text{bus}}} \mapsto \mathbb{C}^{N_{\text{line}}}$ representing apparent branch power flows for all branches at time period k; for the detailed formulation of branch power flows, we refer the reader to [110]. Evidently, MPOPF (6.11) constructs a simultaneous formulation of N_k AC OPF problems with standard power flow constraints (6.11b)-(6.11e), coupled by intertemporal interactions (6.11f) (6.11g) and the corresponding upper and lower bounds (6.11h). Notice that ESSs possess time-variant power limits in (6.11h), due to the inner hyperbox approximation (6.7) utilized for aggregating distribution systems.

Rather than devising intricate mathematical models to precisely represent distribution systems, the flexibility aggregation method offers a substantial reduction in computational complexity of the MPOPF in (6.11). It enhances the scalability of the proposed framework without sacrificing the active involvement of distribution systems in the dispatch problems.

6.4 Distributed Optimization with Receding Horizon

This section presents the proposed distributed real-time coordination framework of ITD systems using a receding horizon scheme while considering day-ahead forecast and actual values. Compared to the classical distributed MPC scheme, only solving the structured optimal control problem either in a spatially distributed manner or in a temporally distributed manner [162], our approach decouples the optimization problems across both different system operators and periods, with each subproblem representing an individual single-period AC OPF of a single transmission system.

6.4.1 Spatial and Temporal Decomposition

We describe a coordination problem of ITD systems by a tuple $\mathcal{C} = (\mathcal{N},\, \mathcal{L},\, \mathcal{K},\, \mathcal{R})$ over N_k time periods. Thereby, $\mathcal{N}$ denotes the set of all buses, $\mathcal{L}$ the set of all branches, and $\mathcal{R} = \{T_1,\, T_2,\, \cdots\}$ denotes the set of coordinated transmission systems.

The objective function (6.11a) represents the sum of quadratic generation cost from all regions $\ell \in \mathcal{R}$ over all time periods $k \in \mathcal{K}$. This enables a straightforward separation of the objective function across different system operators and time periods:

$$f(x) = \sum_{k\in\mathcal{K}} \sum_{\ell\in\mathcal{R}} f_{k,\ell}(z_{k,\ell}),$$

where $z_{k,\ell}$ represents state variables in the transmission system ℓ at the time period k and x is a vector stacking all the subvectors $z_{k,\ell}$.

The constraints of MPOPF (6.11) can be decoupled across time periods, where each of the temporal coupling constraints (6.11f) (6.11g) is associated with only one specific transmission system. Thereby, these temporal coupling constraints can be written in the following standard affinely coupled form

$$\sum_{k\in\mathcal{K}} \Lambda_{k,\ell} z_{k,\ell} = 0,\ \ell \in \mathcal{R},$$

where the sparse matrices $\Lambda_{k,\ell}$ contain non-zero elements of $\{-1, 1, \Delta t\}$, connecting the state variables current $E_{k,\ell}$ and $P^g_{k,\ell}$ with neighboring time periods $\{k-1, k+1\}$ for each transmission system ℓ.

Regarding spatial coupling among different TSOs, we follow the idea of sharing components [6], i.e., sharing nodal voltage angles and magnitudes at both sides of connecting tie-lines between

neighboring transmission systems. The resulting spatial coupling constraints are linear and can be written in the following affinely coupled form

$$\sum_{\ell \in \mathcal{R}} \Gamma_{k,\ell} z_{k,\ell} = 0, \ k \in \mathcal{K},$$

where the sparse matrices $\Gamma_{k,\ell}$ contain non-zeros elements of $\{-1, 1\}$, connecting the coupling voltage angles and magnitudes between neighboring transmission systems for each time period k.

Thereby, the MPOPF in the transmission level can be decoupled across different system operators and time periods and reformulated in standard affinely distributed form

$$\min \sum_{k \in \mathcal{K}} \sum_{\ell \in \mathcal{R}} f_{k,\ell}(z_{k,\ell}) \qquad \text{Decoupled Objective} \tag{6.12a}$$

$$\text{s.t. } \forall k \in \mathcal{K}, \ \sum_{\ell \in \mathcal{R}} \Gamma_{k,\ell} z_{k,\ell} = 0 \quad | \ \lambda_k \qquad \text{Spatial Couplings} \tag{6.12b}$$

$$\text{s.t. } \forall \ell \in \mathcal{R}, \ \sum_{k \in \mathcal{K}} \Lambda_{k,\ell} z_{k,\ell} = 0 \quad | \ \kappa_\ell \qquad \text{Temporal Couplings} \tag{6.12c}$$

$$\text{s.t. } \begin{cases} \forall \, k \in \mathcal{K}, \ \forall \ell \in \mathcal{R} & \\ c_{k,\ell}(z_{k,\ell}) \leq 0 & | \ \nu_{k,\ell} \end{cases} \qquad \text{Decoupled Constraints} \tag{6.12d}$$

where λ_k, κ_ℓ and $\nu_{k,\ell}$ denote Lagrangian multipliers associated with the corresponding constraints. Constraints (6.12d) correspond to the standard AC OPF constraints (6.11b)-(6.11e) with power and energy limits on the ESSs (6.11h) for each transmission system $\ell \in \mathcal{R}$ over all time periods $k \in \mathcal{K}$.

6.4.2 Distributed Real-Time Iteration

By reformulating (6.11) into the affinely coupled distributed form (6.12), it can be solved efficiently by using distributed optimization algorithms. In this chapter, we tailor the ALADIN algorithm [33] to deal with (6.12) in a closed loop.

As discussed in Section 3.4, the standard ALADIN algorithm encounters scalability challenges due to the combinatorial difficulty when inequality constraints exist. Instead of smoothing the original problems, as explored in Chapter 5, we solve coupled inequality-constrained QP subproblems directly to avoid active-set methods at the cost of computation effort, a strategy

previously described in Section 2.3.1. The resulting distributed coordination scheme in receding horizon fashion is outlined in Algorithm 12.

Algorithm 12 Distributed Real-Time Coordination of ITD Systems

Offline:

– Choose initial guess $(x^0, \lambda^0, \kappa^0)$ for closed loop

Repeat:

1 The local operator of the regional transmission systems measures the current states $(E_{0,\ell}(t), P^g_{0,\ell}(t))$ for all $\ell \in \mathcal{R}$.

2 Solve (6.12) cooperatively to obtain solution $(x^*, \lambda^*, \kappa^*)$ by repeating

a) Solve decoupled NLPs for all $k \in \mathcal{K}$ and $\ell \in \mathcal{R}$

$$\begin{aligned} \min_{x_{k,\ell}} \quad & f_{k,\ell}(x_{k,\ell}) + [\lambda_k^\top, \kappa_\ell^\top][\Gamma_{k,\ell}^\top, \Lambda_{k,\ell}^\top]^\top x_{k,\ell} + \frac{\rho}{2} \left\| x_{k,\ell} - z_{k,\ell} \right\|_2^2 \\ \text{s.t.} \quad & c_{k,\ell}(x_{k,\ell}) \leq 0 \qquad \mid \nu_{k,\ell}. \end{aligned} \tag{6.13}$$

b) Compute the Jacobian matrix $J_{k,\ell}$ of constraints $c_{k,\ell}$ at the local solution $x_{k,\ell}$ by

$$J_{k,\ell} = \nabla c_{k,\ell}(x_{k,\ell}) \tag{6.14}$$

with $[\cdot]_i$ denotes the i-th row, and gradient

$$g_{k,\ell} = \nabla f_{k,\ell}(x_{k,\ell}),$$

and choose Hessian approximation

$$0 \prec H_{k,\ell} \approx \nabla^2 \left\{ f_{k,\ell}(x_{k,\ell}) + \nu_{k,\ell}^\top ck, \ell(x_{k,\ell}) \right\}. \tag{6.15}$$

c) Update $(x \leftarrow y + \Delta y, \lambda \leftarrow \lambda^{\mathrm{QP}}, \kappa \leftarrow \kappa^{\mathrm{QP}})$ by solving

$$\begin{aligned} \min_{\Delta y, s} \quad & \sum_{k \in \mathcal{K}} \left\{ \lambda_k^\top \; s_{1,k} + \frac{\mu_1}{2} \left\| s_{1,k} \right\|_2^2 \right\} + \sum_{\ell \in \mathcal{R}} \left\{ \kappa_\ell^\top \; s_{2,\ell} + \frac{\mu_2}{2} \left\| s_{2,\ell} \right\|_2^2 \right\} \\ & + \sum_{k \in \mathcal{K}} \sum_{\ell \in \mathcal{R}} \left\{ \frac{1}{2} \Delta x_{k,\ell} \top H_{k,\ell} \Delta x_{k,\ell} + g_{k,\ell}^\top \Delta x_{k,\ell} \right\} && \text{(6.16a)} \\ \text{s.t.} \quad & \sum_{\ell \in \mathcal{R}} \Gamma_{k,\ell}(y_\ell + \Delta y_\ell) = s_{1,k} \quad \mid \lambda_k^{\mathrm{QP}}, \; k \in \mathcal{K}, && \text{(6.16b)} \\ & \sum_{k \in \mathcal{K}} \Lambda_{k,\ell}(y_\ell + \Delta y_\ell) = s_{2,\ell} \quad \mid \kappa_\ell^{\mathrm{QP}}, \; \ell \in \mathcal{R}, && \text{(6.16c)} \\ & c_{k,\ell}(x_{k,\ell}) + J_{k,\ell} \, \Delta x_{k,\ell} \leq 0, \quad \ell \in \mathcal{R}, \; k \in \mathcal{K}. && \text{(6.16d)} \end{aligned}$$

3 The local ℓ-th TSO for all $\ell \in \mathcal{R}$ deploys their first inputs $(P^{s,*}_{0,\ell}(t), \Delta P^{g,*}_{0,\ell}(t))$ to the real process and sends the solution to connected DSOs.

4 Reinitialize for all $\ell \in \mathcal{R}$

$$x_\ell^0 \leftarrow (x^*_{2,\ell}, ..., x^*_{N_k,\ell}, x^*_{N_k,\ell}), \; \kappa_\ell^0 \leftarrow ([\kappa_\ell^*]_2, ..., [\kappa_\ell^*]_{N_k}, 0)$$

with $[\cdot]_k$ the elements w.r.t k-th time coupling, and

$$\lambda^0 \leftarrow (\lambda^*_2, ..., \lambda^*_{N_k}, 0).$$

Then, set $t \leftarrow t + 1$ and go to Step 1).

Based on the local measurements collected in Step 1), Step 2) of Algorithm 12 outlines the tailored ALADIN algorithm to solve (6.12). Step 2.a) solves $N_k \cdot |\mathcal{R}|$ subproblems, in which the regional TSO deals with the N_k temporal subproblems in parallel locally. These problems are constructed using the Lagrangian of (6.12) by dualizing the spatial coupling (6.12b) and temporal coupling (6.12c). Based on the decoupled solutions $x_{k,\ell}$, Step 2.b) computes sensitivities of objective and constraints with respect to the current iteration of ALADIN. Here, in order to improve the numerical robustness of the algorithm, a small perturbation is added to the second-order derivatives (6.15) approximated by positive definite $H_{k,\ell}$ Notice that under a mild assumption for the perturbation as outlined in [36, Theorem 2], the local quadratic convergence can be guaranteed. Step 2.c) solves the coupled QP (6.16) with only equality constraints. Taking the temporal coupling (6.16c) as local equality constraints for the ℓ-th region, one can solve (6.16) in a decentralized manner that only requires neighbor-to-neighbor communications. For more details, the reader is referred to [103]. Algorithm 12 terminates if the primal conditions

$$\max_{k\in\mathcal{K}} \left\| \sum_{\ell\in\mathcal{R}} \Gamma_{k,\ell} y_{k,\ell} \right\| \leq \epsilon, \max_{\ell\in\mathcal{R}} \left\| \sum_{k\in\mathcal{K}} \Lambda_{k,\ell} y_{k,\ell} \right\| \leq \epsilon, \tag{6.17a}$$

and dual condition

$$\max_{\substack{k\in\mathcal{K}\\ \ell\in\mathcal{R}}} \|y_{k,\ell} - x_{k,\ell}\| \leq \epsilon \tag{6.17b}$$

hold. When employing Algorithm 12 as an online solver, Algorithm 12 presents a receding horizon scheme to coordinate the ITD system in the closed loop. During the online process, each local TSO measures the states $(E_{0,\ell}, P^g_{0,\ell})$ and then, Algorithm 12 solves (6.12) in a distributed manner. After local solutions are determined at the transmission level, the determined inputs $(P^{s,*}_{1,\ell}, \Delta P^{g,*}_{1,\ell})$ are applied to the respective generators and storages. Notably, step 4 in Algorithm 12 serves as an initialization phase for step 2 in the ensuing online cycle, adhering to the methodology outlined in [162].

6.5 A Case Study of a Summer Day in Germany

This section presents a comprehensive evaluation of the proposed coordination strategy by examining its performance through operational scenarios on a summer day in Germany, characterized by considerable prediction mismatches due to severe weather conditions.

6.5.1 Setting

To model an operational scenario within the German electrical grid, we utilize four 118-bus systems from the PGLib-OPF dataset [116], representing the transmission systems. These are interconnected through multiple tie-lines, reflecting the configuration of the four TSOs in Germany, as depicted in Fig. 1.1(b). Additionally, each transmission system is connected to 10 distribution systems in a star configuration, employing the IEEE 33-bus system with multiple DERs for these distribution networks. As a result, the ITD system encompasses a total of 1792 buses with 472 buses at the transmission level and 1320 buses at the distribution level.

To capture modern and contemporary power system dynamics under the impact of severe weather, we utilize measurement data from the ENTSO-E Transparency Platform* [41] dated July 24, 2023. As depicted in Fig. 6.2, the utilized data includes day-ahead predictions (dotted lines) and actual values (solid lines) for load demand, solar generation, and wind generation in each TSO in Germany. This day was marked by adverse weather events, including heavy rainfall, leading to noticeable prediction mismatches, particularly in solar generation. This is visually represented in Fig. 6.2, highlighting a substantial mismatch during the noon hours.

The simulations cover a 24-hour period with a prediction horizon with $N_k = 96$ and a time interval of $\Delta t = 15$ min. By aggregating flexibility from DSOs to the transmission level, we significantly reduce the complexity of the optimization problems by not delving into the detailed network topologies but rather by considering the power-energy envelope of the distribution systems at PCC. Consequently, The optimization tasks at the transmission level involve $187,776$ state variables divided into 4 transmission systems, each across 96 time periods, resulting in a total of 384 subproblems.

Note that, in Fig. 6.2 to Fig. 6.5, the data are arranged in multiple columns to enable a detailed comparative analysis. Specifically, the first four columns in each figure correspond to data from four distinct control areas, i.e., four TSOs and their respective DSOs, respectively. The final column integrates this data, offering a synthesized overview of these four control areas. This configuration facilitates a straightforward comparison across the spatial decomposition to ensure a structured and clear presentation of the simulation results.

*The data utilized in this chapter is available online at the ENTSO-E Transparency Platform: https://transparency.entsoe.eu

6.5.2 Isolated vs. Coordinated Operation Mode

Three distinct operating strategies are explored in the case study: isolated operation, centralized coordination, and distributed coordination. In all these strategies, the flexibility of distribution systems is aggregated to the transmission level as proposed in Sec. 6.2, and the dispatch problems at the transmission level are optimized with a receding horizon. The primary differences between these strategies lie in their operational methodologies and how they address the economic dispatch problems at the transmission level.

In isolated operation mode, each TSO operates in an islanded manner without any communication or power exchange with other transmission systems. The results of the economic dispatches per time period are comprehensively visualized in Fig. 6.3. The net power generations—calculated as the positive stacked bars minus the negative stacked bars—marginally exceed the actual demands (red lines) over a 24-hour period, across all instances in Fig. 6.3. This indicates that the balance between supply and demand is maintained, with minimal power line losses.

Contrary to isolated operation, both the centralized and the distributed coordinations utilize the combined system model (6.11) to facilitate autonomous power exchange (purple bars) between TSOs, aiming to minimize overall generation costs, as depicted in Fig. 6.4. Fig. 6.5 demonstrates the state of charge (Soc) of ESSs, highlighting the effective autonomous management in supporting dispatch tasks while adhering to the energy constraints of the ESSs.

A noteworthy instance occurs at $13:00$, highlighted as vertical dotted lines, where transmission system T_1 encounters a significant prediction mismatch. In this time period, T_1 experiences higher actual demand and reduced solar generation, as shown in Fig. 6.2, coinciding with lower SOC of ESSs in T_1, as shown in Fig. 6.5. In response to this prediction error, power export to other systems (purple bar) is intentionally curtailed as a compensatory measure, demonstrating the system's capacity to adapt to unexpected operational dynamics.

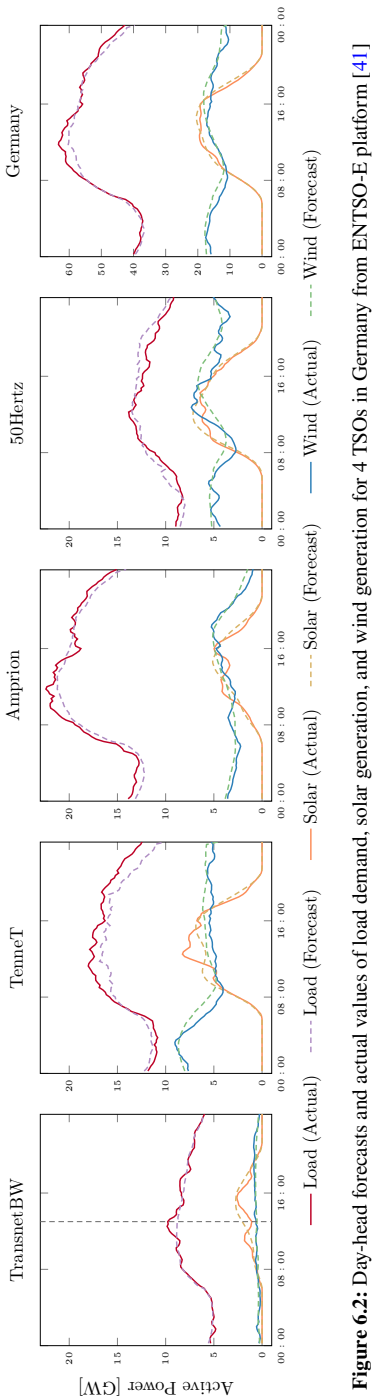


Figure 6.2: Day-head forecasts and actual values of load demand, solar generation, and wind generation for 4 TSOs in Germany from ENTSO-E platform [41]

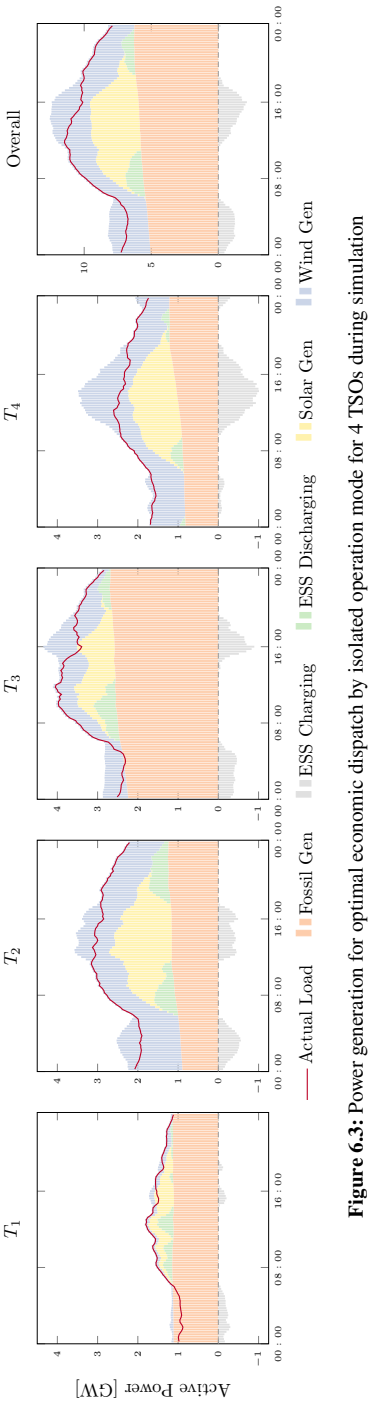


Figure 6.3: Power generation for optimal economic dispatch by isolated operation mode for 4 TSOs during simulation

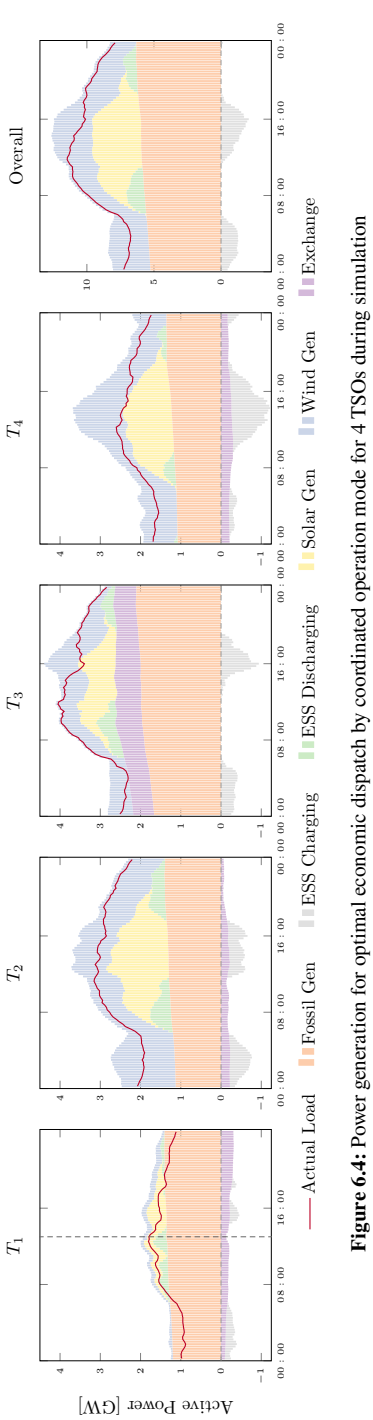


Figure 6.4: Power generation for optimal economic dispatch by coordinated operation mode for 4 TSOs during simulation

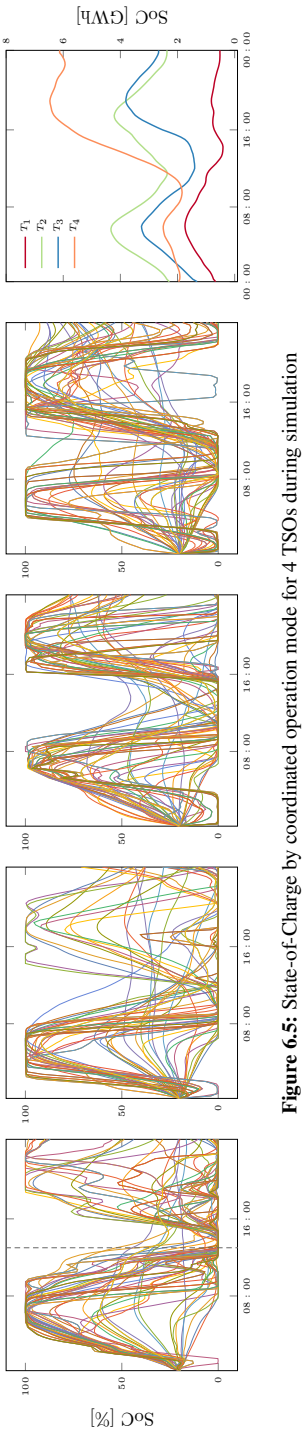


Figure 6.5: State-of-Charge by coordinated operation mode for 4 TSOs during simulation

6.5.3 Centralized vs. Distributed Coordination Approaches

The key difference between these two coordination strategies lies in the optimization approaches. Centralized coordination communicates all private data to a centralized entity and employs a centralized algorithm to solve the optimization problem. In contrast, distributed coordination solves the optimization problem based on the proposed algorithm in a distributed fashion with limited information exchanged between TSOs.

Given that both the centralized and the distributed coordination adopt the same system model (6.11) with $187,776$ state variables divided into 384 subproblems, we use centralized solutions as reference solutions to evaluate the effectiveness of the proposed distributed approach in solving the economic dispatch problems at the transmission level. The convergence performance of the proposed distributed approach across 24 hours is demonstrated in Fig. 6.6, representing a number of iterations to converge, total computing time for solving one economic dispatch problem, primal and dual residual (6.17), deviations of controllable power injections and optimality gap for each TSOs, expressed as $\left|\frac{f(x_\ell)-f(x_\ell^*)}{f(x_\ell^*)}\right|$. Notably, all the 96 optimization tasks during the daily operation demonstrate fast convergence in a dozen iterations, under 500 seconds, with both the primal and dual residuals reaching tolerable values. Compared with centralized coordination, the proposed distributed approach showcases remarkable accuracy in terms of controllable power injections and total optimality gap over all 96 time periods. These results highlight the scalability and numerical robustness for real-world applications in large-scale ITD systems.

Table 6.1: Generation Costs [€] with aggregated Flexibility of DSOs

	Isolated	Centralized Coordination	Distributed Coordination
T_1	2 034 052	2 499 736	2 499 785
T_2	2 006 145	2 396 549	2 396 573
T_3	5 597 566	4 058 846	4 058 842
T_4	1 778 179	2 241 376	2 241 368
Total	11 415 942	11 196 505	11 196 568

The economic efficiency comparison among the three operational strategies, as shown in Table 6.1 indicates that operating in isolation leads to the highest total costs, whereas centralized coordination results in the lowest. Distributed coordination presents a viable alternative, balancing data privacy and competitive costs, approximately 0.0006% higher than centralized methods.

Both coordination strategies effectively find local minimizers of the system model (6.11), with negligible differences in total costs.

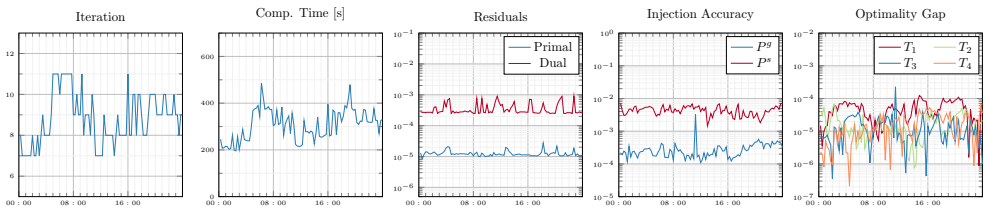


Figure 6.6: Convergence Performance of distributed NMPC

6.6 Summary

This chapter proposes a novel real-time distributed operational framework for efficient coordination of ITD systems. It employs a flexibility aggregation method at the distribution level, leveraging controllable devices through power-energy envelopes provided by DSOs, thereby avoiding additional computational complexity of economic dispatch problems at the transmission level. Furthermore, the framework's receding horizon strategy enhances its robustness against prediction mismatches, especially under severe weather conditions, highlighted by a case study of a summer day in Germany. By utilizing real operational data with significant prediction mismatches, this study confirms the framework's practical relevance and applicability in real-world scenarios.

7 Conclusion & Future Work

This final chapter provides an overview of the dissertation and proposes avenues for future research to build upon the foundations laid by this work.

7.1 Conclusion

Addressing the challenges mentioned in Section 1.1, this dissertation advances distributed optimization in power systems by focusing on scalability, computation and communication efficiency, and practical deployment in real-world scenarios. We summarize some of the main insights and novel ideas introduced in this work. A detailed overview of the structure and the specific contributions is provided in Section 1.3.

Chapter 3 presents the augmented Lagrangian based alternating direction inexact Newton (ALADIN), designed for solving generic nonlinear program (NLP) in a distributed fashion, showing greater potential than classic first-order distributed optimization, particularly for nonconvex problems. This dissertation enhances ALADIN algorithms for power system applications in Chapter 4. Specifically, a Gauss-Newton-based ALADIN framework is proposed to solve AC power flow (PF) problems, modeled as nonlinear equation systems, detailed in Section 4.3. Further improvements involve refining the globalization strategy to improve ALADIN's robustness, enabling reliable convergence from a broader range of initial conditions while preserving the algorithm's efficiency and performance, as detailed in Section 4.4. Additionally, the standard ALADIN is deployed in a distributed computation environment to solve AC optimal power flow (OPF) problems. The cooperation between multiple system operators is simulated in a geographically distributed environment, with observed performance trade-offs outlined in Section 4.5.

While fast convergence speed can be obtained by large-scale problems in the presence of inequality constraints, as demonstrated in Section 4.3, scalability of ALADIN is limited with inequality constraints due to the combinatorial difficulty, as discussed in Section 2.3.3. Additionally, the standard ALADIN requires solving full-dimensional QP subproblems centrally, which increases communication demands and risks exposing topology information contained in the sparsity pattern of Jacobian and Hessian matrices.

To solve these issues, Chapter 5 proposes a novel distributed optimization algorithm that combines ALADIN with the interior point method (IPM), details outlined in Section 5.2. This integrated approach enhances scalability compared to the standard ALADIN algorithm by circumventing combinatorial difficulties while preserving convergence guarantees as analysis in Section 5.3. Communication demands and the risk of exposing topology information are reduced by condensing the derivatives locally in parallel, improving overall algorithm efficiency. The proposed distributed algorithm is thoroughly tested on extensive benchmarks under various operational scenarios in Section 5.5, even outperforming state-of-the-art centralized nonlinear solvers on an inexpensive workstation. Additionally, the communication requirements of the algorithm are evaluated under both worst-case and practical conditions in Section 5.6, providing valuable insights into its applicability in real-world distributed systems.

Another important contribution is the development of a real-time distributed coordination framework for integrated transmission and distribution systems, as presented in Chapter 6. This framework aggregates the flexibility of distributed energy resources (DERs) within distribution systems, as described in Section 6.2, enhancing power grids' overall operational efficiency and reliability. By employing nonlinear model predictive control (NMPC), the framework enables proactive and adaptive economic dispatch, accommodating the dynamic nature of renewable energy sources, as outlined in Section 6.4. The framework's effectiveness is validated through simulations conducted over a summer day in Germany, characterized by significant prediction mismatch in energy production and consumption. These simulations demonstrate the framework's ability to manage complex power systems in real-time while ensuring data privacy and scalability. Detailed results and analyses of these simulations are presented in Section 6.5.

In conclusion, this dissertation has laid a strong foundation for more efficient and scalable distributed optimization in large-scale power systems. It demonstrates the feasibility and potential of distributed approaches by addressing both theoretical and practical challenges. The proposed methods improve scalability, computational, and communication efficiency, providing a pathway toward more resilient and adaptable energy management solutions. Future research will build on these findings, pushing the boundaries of distributed energy management and ensuring that power systems are ready to meet future challenges.

7.2 Future Work

While this dissertation makes significant progress in advancing distributed optimization for power systems, several opportunities for further research remain. One important direction is enhancing the convergence speed of the proposed algorithms. Adaptive strategies for better tuning algorithm

parameters could facilitate faster and more reliable convergence, making the optimization process more efficient for real-time applications, especially in scenarios requiring rapid decision-making.

Further simulations and case studies on larger power system benchmarks are needed to evaluate the algorithms' performance in more complex and diverse operational scenarios, such as security-constrained AC optimal power flow (SC-AC-OPF). These studies would provide deeper insights into the robustness and efficiency of distributed optimization methods when applied to larger and more intricate power systems, thereby validating their scalability and practicality in real-world settings.

Expanding the real-time coordination framework between transmission and distribution systems is another promising area. Refining the coordination algorithms to reduce latency and enhance stability during real-time operations is critical for reliable system performance. Effective real-time coordination among multiple system operators will improve the scalability and adaptability of distributed energy systems, especially in integrated transmission and distribution networks.

Finally, exploring hardware acceleration techniques, such as using graphics processing units (GPUs), also offers a promising avenue for advancing distributed optimization. Hardware acceleration could significantly boost computational efficiency, allowing for faster processing and enabling optimization of increasingly large-scale systems. Addressing the computational demands of large-scale optimization problems is crucial as power systems grow in complexity. Leveraging modern hardware capabilities could make distributed optimization algorithms more practical and scalable, supporting their broader adoption in real-world power system applications.

A Mathematical Fundamentals

A.1 Background Material

Theorem A.1 (Taylor's Theorem [46, Thm 2.1]). *Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable and that $p \in \mathbb{R}^n$. Then we have that*

$$f(x+p) = f(x) + \nabla f(x+tp)^\top p,$$

for some $t \in (0,1)$

Theorem A.2 (Mean Value Theorem[46]). *Given a vector-valued function $f : \mathbb{R}^n \to \mathbb{R}^m$ and its Jacobian matrix $J = \frac{\partial f}{\partial x} \in \mathbb{R}^{m \times n}$. Given x, $p \in \mathbb{R}^n$, we have*

$$f(x+p) - f(x) = \int_0^1 J(x+\tau\, p)\, p\, d\tau \tag{A.1}$$

Definition A.1 (Lipschitz Continuity [46]). *Considering a function $f : \mathbb{R}^n \to \mathbb{R}^m$. The function f is said to be Lipschitz continuous on some set $\mathcal{X} \in \mathbb{R}^n$ if there is a constant $L > 0$ such that*

$$\|f(x_1) - f(x_0)\| \leq L \|x_1 - x_0\|, \quad \forall x_0, x_1 \in \mathcal{X}. \tag{A.2}$$

Here, L is called the *Lipschitz constant*.

A.2 Proof of Lemma 3.1

The proof follows the globalization anaylsis of ALADIN in [33, Sec. 6]. It is provided here to ensure a complete and self-contained presentation.

Proof. Assumption 3.3 implies that without update of the primal variable z, step c) of Algorithm 4 ensures that the dual variable λ can conrgece to a limit point

$$\lambda^\star = \arg\max_{\lambda} V_\rho(z, \lambda),$$

which is equalient to the optimal dual variables to a auxiliny optimziationn problem

$$\min_{x} \quad \sum_{\ell \in \mathcal{R}} \left\{ f_\ell(x_\ell) + \frac{\rho}{2} \left\| x_\ell - z_\ell \right\|_{\Sigma_\ell}^2 \right\} \tag{A.3a}$$

$$\text{s.t.} \quad \sum_{\ell \in \mathcal{R}} A_\ell x_\ell - b = 0 \qquad \mid \lambda \tag{A.3b}$$

$$c_\ell(x_\ell) \leq 0, \quad \ell \in \mathcal{R}. \tag{A.3c}$$

A exact merit function for the auxiliny problem A.3 can be set based on (3.11)

$$\begin{aligned}\bar{\Phi}(x) =& \Phi(x) + \sum_{\ell \in \mathcal{R}} \frac{\rho}{2} \left\| x_\ell - z_\ell \right\|_{\Sigma_\ell}^2 \\ =& \sum_{\ell \in \mathcal{R}} \left\{ f_\ell(x_\ell) + \sum_{\ell \in \mathcal{R}} \frac{\rho}{2} \left\| x_\ell - z_\ell \right\|_{\Sigma_\ell}^2 \right\} + \bar{\lambda} \left\| \sum_{\ell \in \mathcal{R}} A_\ell x_\ell - b \right\|_1 + \bar{\kappa} \sum_{\substack{\ell \in \mathcal{R} \\ i \in \mathcal{C}_\ell}} \max\{0, [c_\ell(x_\ell)]_i\}\end{aligned}$$

With the optimal dual λ^*, the local solution to the decoupled NLP problems converge to the optimal primal variables to the auxiliny problem A.3, i.e.,

$$x^\star = \arg\min_{x} \sum_{\ell \in \mathcal{R}} \left\{ f_\ell(x_\ell) + \frac{\rho}{2} \left\| x_\ell - z_\ell \right\|_{\Sigma_\ell}^2 \right\}, \text{ s.t. (A.3b) (A.3c).}$$

$$
\begin{aligned}
\Phi(z) &= \sum_{\ell\in\mathcal{R}} f_\ell(z_\ell) + \bar{\lambda}\left\|\sum_{\ell\in\mathcal{R}} A_\ell z_\ell - b\right\|_1 + \bar{\kappa}\sum_{\substack{\ell\in\mathcal{R}\\ i\in\mathcal{C}_\ell}} \max\{0, [c_\ell(z_\ell)]_i\} \\
&= \sum_{\ell\in\mathcal{R}} \left\{ f_\ell(z_\ell) + \frac{\rho}{2}\left\|z_\ell - z_\ell\right\|_{\Sigma_\ell}^2 \right\} + \bar{\lambda}\left\|\sum_{\ell\in\mathcal{R}} A_\ell z_\ell - b\right\|_1 + \bar{\kappa}\sum_{\substack{\ell\in\mathcal{R}\\ i\in\mathcal{C}_\ell}} \max\{0, [c_\ell(z_\ell)]_i\} \\
&\quad \Downarrow\ \bar{\Phi}(z) \geq \bar{\Phi}(x^\star) \\
&\geq \sum_{\ell\in\mathcal{R}} \left\{ f_\ell(x_\ell^\star) + \frac{\rho}{2}\left\|x_\ell^\star - z_\ell\right\|_{\Sigma_\ell}^2 \right\} + \bar{\lambda}\left\|\sum_{\ell\in\mathcal{R}} A_\ell x_\ell^\star - b\right\|_1 + \bar{\kappa}\sum_{\substack{\ell\in\mathcal{R}\\ i\in\mathcal{C}_\ell}} \max\{0, [c_\ell(x_\ell^\star)]_i\} \\
&\quad \Downarrow\ x^\star \text{ satisfies the constraints (A.3b) (A.3c)} \\
&= \sum_{\ell\in\mathcal{R}} \left\{ f_\ell(x_\ell^\star) + \frac{\rho}{2}\left\|x_\ell^\star - z_\ell\right\|_{\Sigma_\ell}^2 \right\} \\
&= \Phi(x^\star) + \frac{\rho}{2}\sum_{\ell\in\mathcal{R}} \left\|x^\star - z\right\|_{\Sigma_\ell}^2
\end{aligned}
$$

which implies

$$
\Phi(z) - \Phi(x^\star) \geq \sum_{\ell\in\mathcal{R}} \frac{\rho}{2}\left\|x^\star - z\right\|_{\Sigma_\ell}^2 + \bar{\lambda}\left\|\sum_{\ell\in\mathcal{R}} A_\ell x_\ell^\star - b\right\|_1 \tag{A.4}
$$

This means that th $x^\star$ is a strict desecnt direction. More detailed analysis refer to [33, Sec. 6] □

A.3 Proof of Lemma 4.1

Proof. To show the equivalence, we write down the analytical solution of (4.27) by working out its KKT system,

$$\begin{bmatrix} H_1 & & & & A_1^\top \\ & H_2 & & & A_2^\top \\ & & \ddots & & \vdots \\ & & & H_N & A_3^\top \\ A_1 & A_2 & \dots & A_N & \end{bmatrix} \begin{pmatrix} x_1^{\mathrm{QP}} - y_1 \\ x_2^{\mathrm{QP}} - y_2 \\ \vdots \\ x_N^{\mathrm{QP}} - y_N \\ \lambda^{\mathrm{QP}} \end{pmatrix} = \begin{pmatrix} -g_1 \\ -g_2 \\ \vdots \\ -g_N \\ b - \sum_{i=1}^N A_\ell x_\ell \end{pmatrix} \tag{A.5}$$

$$\Longrightarrow \begin{cases} \lambda^{\mathrm{QP}} = M^\dagger \left(\sum_{i=1}^N A_\ell (x_\ell - H_\ell^{-1} g_\ell) - b \right) , \\ \zeta_\ell^{\mathrm{QP}} = x_\ell - H_\ell^{-1}(A_\ell^\top \lambda^{\mathrm{QP}} + g_\ell) , \quad i = 1, ..., N . \end{cases} \tag{A.6}$$

Here, the first equation in (A.6) uses the equality $\sum_{i=1}^N A_\ell \zeta_\ell = b$. Then, we write down the coupled QP if we use standard ALADIN to solve 4.32,

$$\tilde{x}^{\mathrm{QP}} := \arg\min_x \quad \sum_{i=1}^N \left\{ \frac{1}{2} \zeta_\ell^\top H_\ell \zeta_\ell + (g_\ell + \rho \Sigma_\ell (x_\ell - z_\ell) - H_\ell x_\ell)^\top \zeta_\ell \right\} \tag{A.7a}$$

$$\text{s.t.} \quad \sum_{i=1}^N A_\ell \zeta_\ell = b \qquad \mid \tilde{\lambda}^{\mathrm{QP}} . \tag{A.7b}$$

As the KKT system is linear, the solution $\tilde{x}^{\mathrm{QP}}$ and $\tilde{\lambda}^{\mathrm{QP}}$ can be represented by

$$\tilde{x}^{\mathrm{QP}} = x^{\mathrm{QP}} + \Delta x^{\mathrm{QP}} , \quad \tilde{\lambda}^{\mathrm{QP}} = \lambda^{\mathrm{QP}} + \Delta \lambda^{\mathrm{QP}} ,$$

where $(\Delta x^{\mathrm{QP}}, \Delta \lambda^{\mathrm{QP}})$ is the solution of linear equations

$$\begin{bmatrix} H_1 & & & & A_1^\top \\ & H_2 & & & A_2^\top \\ & & \ddots & & \vdots \\ & & & H_N & A_N^\top \\ A_1 & A_2 & \dots & A_N & \end{bmatrix} \begin{pmatrix} \Delta x_1^{\mathrm{QP}} \\ \Delta x_2^{\mathrm{QP}} \\ \vdots \\ \Delta x_N^{\mathrm{QP}} \\ \Delta \lambda^{\mathrm{QP}} \end{pmatrix} = \rho \begin{pmatrix} \Sigma_1 (z_1 - y_1) \\ \Sigma_2 (z_2 - y_2) \\ \vdots \\ \Sigma_N (z_N - y_N) \\ 0 \end{pmatrix} .$$

The solution of the linear equations above is given by

$$\Delta\lambda^{\text{QP}} = \rho M^{\dagger} \left(\sum_{i=1}^{N} A_\ell H_\ell^{-1} \Sigma_\ell (z_\ell - x_\ell) \right) , \tag{A.8a}$$

$$\Delta\zeta_\ell^{\text{QP}} = H_\ell^{-1} (\rho \Sigma_\ell (z_\ell - x_\ell) - A_\ell^{\top} \Delta\lambda^{\text{QP}}) , \quad i = 1, ..., N, \tag{A.8b}$$

which concludes the proof. □

B Optimization Problems in Power System

Accurate grid modeling is crucial for reliable power system analysis and decision-making over short to medium timeframes, typically from seconds to hours. This section introduces two key steady-state problems in power system engineering. The first is the AC power flow (PF) problem, a set of nonlinear equations used to determine the steady-state (complex) quantities in an AC electrical network. The second is the AC optimal power flow (OPF) problem, a constrained nonlinear optimization problem designed to minimize electricity generation costs while ensuring the system operates safely within specified limits.

B.1 System Models

This section reviews the key components of power systems, outlines mathematical models for describing AC power flows, and introduces the notations used throughout this dissertation. A basic understanding of electrical circuit theory, the frequency domain (phasor & angle) representation of electrical quantities, and the concept of complex electric power is assumed. Readers unfamiliar with these topics are encouraged to refer to [110, Appendix B] for a brief overview.

B.1.1 Components of the System Network

Power systems consist of buses (nodes where components such as generators and loads are connected) and branches (transmission lines and transformers that link the buses). Each component has associated variables and constraints, such as power outputs and voltage magnitudes, which are subject to operational limits. The network must also maintain a power balance at each bus, introducing constraints that couple components across the system. This section provides an overview of the core elements of a power system, focusing on power and voltage variables and their associated constraints.

We represent a power system as a network $\mathcal{S} = (\mathcal{N},\ \mathcal{L})$, where $\mathcal{N}$ represents the set of buses, $\mathcal{L}$ represents the set of branches.

B.1.1.1 Buses

A bus represents a node in an electrical network where generators, loads, and branches connect. The complex voltage at bus i, denoted by V_i, can be represented in polar coordinates and in rectangular coordinates.

In polar coordinates, the complex volage is expressed as:

$$V_i = v_i e^{\theta_i}, \quad \forall i \in \mathcal{N}, \tag{B.1}$$

where v_i and θ_i are the magnitude and angle of the complex voltage V_i. The voltage magnitude is constrained by:

$$\underline{v}_i \leq v_i \leq \overline{v}_i, \quad \forall i \in \mathcal{N}, \tag{B.2}$$

where $\underline{v}_i \leq \overline{v}_i$ for all $i \in \mathcal{N}$ are lower and upper voltage bounds. Since voltage angles are relative, a reference bus is selected with a fixed angle, typically:

$$v_{\text{ref}} = 0. \tag{B.3}$$

As an alternative, in rectangular coordinates, V_i is written as:

$$V_i = u_i + \mathrm{j} w_i, \quad \forall i \in \mathcal{N}, \tag{B.4}$$

where u_i and w_i denote the real and imaginary components of the complex voltage V_i. The corresponding voltage magnitude limits in this representation become:

$$\underline{v}_i^2 \leq u_i^2 + w_i^2 \leq \overline{v}_i^2, \quad \forall i \in \mathcal{N}, \tag{B.5}$$

The reference angle constraint is expressed as:

$$w_{\text{ref}} = 0. \tag{B.6}$$

B.1.1.2 Generators

Generators transform primary energy sources into electrical power. At bus $i \in \mathcal{N}$, a generator can supply active power p_i^g and reactive power q_i^g subject to operational limits:

$$\underline{p}_i^g \leq p_i^g \leq \overline{p}_i^g, \quad \underline{q}_i^g \leq q_i^g \leq \overline{q}_i^g, \quad \forall i \in \mathcal{N}. \tag{B.7}$$

If no generator is connected to bus i, the generator outputs are set to zero as $p_i^g = q_i^g = 0$.

B.1.1.3 Branches

Branches allow for the flow of electricity from generators to loads. The most common type is the AC transmission line, as discussed in Appendix B.1.2. For other kinds of branches, including transformers, phase shifters, and DC transmission lines, the reader is referred to [112, 163] for further information.

Each transmission line (i, j) connects buses from bus i (from-side) to bus j (to-side). AC power lines have thermal limits to prevent lines from sagging and automatic protection devices from activating [164]. These limits are typically given as apparent power limits on the lines, i.e.,

$$|s_{ij}| \leq s_{ij}^{\max}, \quad \forall (i,j) \in \mathcal{L} \cup \mathcal{L}^R, \tag{B.8a}$$

with

$$s_{ij} = \sqrt{p_{ij}^2 + q_{ij}^2}, \quad \forall (i,j) \in \mathcal{L} \cup \mathcal{L}^R, \tag{B.8b}$$

where s_{ij}, p_{ij} and q_{ij} denote the active, reactive and apparent power flow bus i to bus j. Note that $\mathcal{L}$ represents the set of the from-side branches, while $\mathcal{L}$ represents the to-side branches. The apparent power flow limits are added to both sets of the branches due to the potential bidirectional power flow between bus i and bus j.

Small phase angle differences between from-side and to-side are also a design imperative in AC power systems [164]. These constraints are not typically incorporated in AC transmission test cases in Matpower [112]. However, recent work [165, 166] has observed that incorporating phase angle difference (PAD) constraints, i.e.,

$$-\theta_{ij}^{\max} \leq \theta_{ij} \leq \theta_{ij}^{\max}, \quad \forall (i,j) \in \mathcal{L}, \tag{B.9}$$

where $\theta_{ij} = \theta_i - \theta_j$ is the PAD between bus i and bus j, and $\theta_{ij}^{\max}$ denote the maximal PAD along the branch $(i, j) \in \mathcal{L}$. For systems using rectangular coordinates (B.4), these PAD constraints can

be reformulated as a linear relationship between the real and imaginary components of $V_i V_j^*$ [167]:

$$\tan(-\theta_{ij}^{\max})\, \mathfrak{Re}(V_i V_j^*) \leq \mathfrak{Im}(V_i V_j^*) \leq \tan(\theta_{ij}^{\max})\, \mathfrak{Re}(V_i V_j^*), \quad \forall (i,j) \in \mathcal{L}, \tag{B.10a}$$

where

$$\mathfrak{Re}(V_i V_j^*) = u_i u_j + w_i w_j \text{ and } \mathfrak{Im}(V_i V_j^*) = u_j w_i - u_i w_j. \tag{B.10b}$$

B.1.1.4 Loads

Loads are modeled as constant withdrawals of active and reactive power from the system. The active and reactive power demands at bus $i \in \mathcal{N}$ are denoted as p_i^d and q_i^d, respectively.

B.1.2 AC Power Flow Models

The AC power flow equations describe the relationships between complex quantities such as current I, voltage V, admittance Y, and power S. The following principles govern these equations:

1 **Kirchhoff's Current Law (KCL):** The sum of currents flowing into and out of a bus must be balanced:

$$I_i^g - I_i^d = \sum_{(i,j) \in \mathcal{L} \cup \mathcal{L}^R} I_{ij} \tag{B.11}$$

where I_i^g and I_i^d denote the generator and load currents at bus i, respectively, and I_{ij} is the current flowing along the branch from bus i to j.

2 **Ohm's Law:** The current along a branch is proportional to the voltage difference between its terminals:

$$I_{ij} = Y_{ij}(V_i - V_j), \tag{B.12}$$

where Y_{ij} is the admittance of the branch connecting buses i and j. For a detailed introduction to the admittance matrix Y, refer to [110, Sec. 4].

3 **Definition of AC Power:** The complex power flow S_{ij} on a branch is defined as:

$$S_{ij} = Y_i I_{ij}^*. \tag{B.13}$$

where V_i is the voltage at bus i, and I_{ij}^* is the complex conjugate of the current flow I_{ij}.

Combining these three properties yields the AC power flow model, i.e.,

$$S_i^g - S_i^d = \sum_{(i,j)\in\mathcal{L}\cup\mathcal{L}^R} S_{ij}, \qquad \forall i \in \mathcal{N} \tag{B.14a}$$

$$S_{ij} = Y_{ij}^* V_i V_i^* - Y_{ij}^* V_i V_j^*, \qquad \forall (i,j) \in \mathcal{L}\cup\mathcal{L}^R. \tag{B.14b}$$

Equation (B.14a) describes the power nodal balance at bus $i \in \mathcal{N}$, while equation (B.14b) defines power flows along branch $(i,j) \in \mathcal{L}\cup\mathcal{L}^R$. These nonlinear equations form the foundation of many power system applications.

For numerical analysis and optimization, the complex AC power flow equations (B.14) can be decomposed into equivalent real-valued nonlinear equations by separating their real and imaginary components. These equations can be expressed in various forms depending on whether power S, voltage V, and admittance Y are represented in polar or rectangular coordinates.

In this dissertation, we use rectangular coordinates for the admittance matrix admittance $Y = G + \mathbf{j}B$ with $G, B \in \mathbb{R}^{|\mathcal{N}_\ell|\times|\mathcal{N}_\ell|}$, and for the power $S = p + \mathbf{j}q$.

When using polar coordinates for voltage (B.1), the AC power flow equations are expressed as:

$$p_i^g - p_i^l = v_i \sum_{j\in\mathcal{N}_T} v_j \left(G_{ij}\cos\theta_{ij} + B_{ij}\sin\theta_{ij}\right), \qquad \forall i \in \mathcal{N}, \tag{B.15a}$$

$$q_i^g - q_i^l = v_i \sum_{j\in\mathcal{N}_T} v_j \left(G_{ij}\sin\theta_{ij} - B_{ij}\cos\theta_{ij}\right), \qquad \forall i \in \mathcal{N}, \tag{B.15b}$$

and the power flows along the branches are expressed as:

$$p_{ij} = v_i^2 G_{ij} - v_i v_j \left(G_{ij}\cos\theta_{ij} + B_{ij}\sin\theta_{ij}\right), \qquad \forall (i,j) \in \mathcal{L}\cup\mathcal{L}^R, \tag{B.15c}$$

$$q_{ij} = -v_i^2 B_{ij} - v_i v_j \left(G_{ij}\sin\theta_{ij} - B_{ij}\cos\theta_{ij}\right), \qquad \forall (i,j) \in \mathcal{L}\cup\mathcal{L}^R. \tag{B.15d}$$

When using rectangular coordinates for voltage (B.4), the AC power flow equations are expressed as:

$$p_i^g - p_i^d = \sum_{k\in\mathcal{N}} G_{ik}(u_i u_k + w_i w_k) + B_{ik}(w_i u_k - u_i w_k), \qquad \forall i \in \mathcal{N}, \tag{B.16a}$$

$$q_i^g - q_i^d = \sum_{k\in\mathcal{N}_\ell} G_{ik}(w_i u_k - u_i w_k) - B_{ik}(u_i u_k + w_i w_k), \qquad \forall i \in \mathcal{N}, \tag{B.16b}$$

and the power flows along the branches are expressed as:

$$p_{ij} = G_{ij}\{u_i(u_i - u_j) + w_i(w_i - w_j)\} + B_{ij}(u_i w_j - w_i u_j), \quad \forall (i,j) \in \mathcal{L} \cup \mathcal{L}^R, \quad \text{(B.16c)}$$

$$q_{ij} = B_{ij}\{u_i(u_j - u_i) + w_i(w_j - w_i)\} + G_{ij}(u_i w_j - w_i u_j), \quad \forall (i,j) \in \mathcal{L} \cup \mathcal{L}^R. \quad \text{(B.16d)}$$

Alternative models for describing AC power flows, including polar coordinates for admittance and the branch flow model (BFM) for radial networks, are discussed in [110].

B.2 Steady-State Problem Formulations

This section introduces two steady-state optimization problems based on the AC power flow models for power system applications.

B.2.1 AC Power Flow Problems

Unlike classic optimization problems, the conventional power flow problem does not include an objective function. Instead, it involves solving a set of equations to compute all complex voltages and power injections in the power system. These equations are categorized into bus specifications and power flow equations.

Bus Specification

In a conventional power flow problem, buses are classified into three types: slack, PV (voltage-controlled), and PQ (load) buses. Table B.1 defines the symbols used for each type.

Symbol	Meaning
θ	Voltage angle
v	Voltage magnitude
p	Real power injection
q	Reactive power injection
n_{pv}	Number of pv buses
n_{pq}	Number of pq buses
n_{bus}	Number of all buses

Table B.1: List of symbols for conventional power flow.

- **Slack Bus:** The slack bus serves as the reference bus and is typically connected to at least one generator in the power system model. Its voltage magnitude and angle are fixed, usually set to $v = 1$ p.u. and $\theta = 0$, while the complex power injection is allowed to vary. The slack bus ensures a unique and feasible solution for the power flow problem [110, Appendix C]. In a conventional power flow problem, there is only one slack bus [110].
- **PV (Voltage-Controlled) Bus:** Generator buses, excluding the slack bus, are classified as PV buses. At these buses, the real power injection and voltage magnitude are fixed, while the reactive power injection and voltage angle are determined by solving the power flow equations.
- **PQ (Load) Bus:** PQ buses are non-generator buses. Unlike the slack bus, the real and reactive power injections are fixed at PQ buses, while the voltage magnitude and angle are free variables.

Bus Type	Slack	pv	pq
Know quantities	θ, v	p, v	p, q
Unknown quantities	p, q	θ, q	θ, v
Number of buses in system	1	n_{pv}	$n_{\text{pq}} = n_{\text{bus}} - n_{\text{pv}} - 1$
Number of bus specifications	2	$2n_{\text{pv}}$	$2n_{\text{pq}}$

Table B.2: Bus types for power flow.

Table B.2 summarizes the fixed and free quantities for different bus types. Each bus has two known variables, resulting in $2n_{\text{bus}}$ bus specifications in a conventional power flow problem. These specifications can be expressed as a vector function:

$$g^{\text{bus}}(x) = 0 \tag{B.17}$$

where the state of the power system is represented as:

$$x = \begin{pmatrix} \theta & v & p & q \end{pmatrix} \tag{B.18}$$

with $\theta,\ v,\ p,\ q \in \mathbb{R}^{n_{\text{bus}}}$. Here, power injection is defined as:

$$p_i = p_i^g - p_i^d, \quad \forall i \in \mathcal{N}, \tag{B.19}$$

$$q_i = q_i^g - q_i^d, \quad \forall i \in \mathcal{N}. \tag{B.20}$$

Each bus is associated with two power flow equations—real and reactive power flow equations. Therefore, a power system with n_{bus} buses has $4n_{\text{bus}}$ power flow equations. These equations, expressed in polar coordinates as (B.15a) (B.15b) or in rectangular coordinates as (B.16a) (B.16b), can be compactly represented as a vector function:

$$g^{\text{pf}}(x) = 0 \tag{B.21}$$

Together, the bus specifications (B.17) and power flow equations form the conventional power flow problem:

$$g^{\text{pf}}(x) = 0 \tag{B.22a}$$
$$g^{\text{bus}}(x) = 0; \tag{B.22b}$$

with problem-specific:

$$g^{\text{pf}} : \mathbb{R}^{4n_{\text{bus}}} \to \mathbb{R}^{2n_{\text{bus}}}$$
$$g^{\text{bus}} : \mathbb{R}^{4n_{\text{bus}}} \to \mathbb{R}^{2n_{\text{bus}}}$$

There are $4n_{\text{bus}}$ variables in this conventional power flow problem, as stated in (B.18). Whereas, the total number of equations of the conventional power flow is also $4n_{\text{bus}}$. Thereby, it can be concluded that, if feasible solution exists, there is only one unique solution of the conventional power flow problem.

Remark B.1. *In general, while a power flow problem may have multiple mathematically valid solutions [168, 169], only one of these solutions corresponds to the physically meaningful operating state of the power system [110]. This uniqueness arises due to the periodic nature of the voltage angle θ and its associated trigonometric functions, which can introduce ambiguity in the mathematical solutions.*

B.2.2 AC Optimal Power Flow Problems

Optimal Power Flow (OPF) combines an objective function with the power flow equations—expressed in polar coordinates as (B.15a) (B.15b) or in rectangular coordinates as (B.16a) (B.16b)—to form an optimization problem. The inclusion of power flow equations distinguishes OPF from other power system problems, such as classic Economic Dispatch (ED), Unit Commitment (UC), and market-clearing problems.

The classic OPF formulation extends the objectives of ED by incorporating the physical constraints of the power system. Its primary objective is to minimize the total cost of electricity generation while ensuring the power system operates within safe limits.

Recalling that the power network in this Appendix is modeled $\mathcal{S} = (\mathcal{N},\ \mathcal{L})$, where $\mathcal{N}$ represents the set of buses, $\mathcal{L}$ represents the set of branches. Each generator incurs an operating cost, typically modeled as a quadratic function of its real power output p_i^g. The objective is to minimize the total generation cost, expressed as:

$$f(p^g) = \sum_{i\in\mathcal{N}} \left\{ a_{2i} \left(p_i^g\right)^2 + a_{1i}\, p_i^g + a_{0i} \right\} \tag{B.24}$$

The classic formulation of AC OPF can be expressed in polar coordinates:

$$\min_{v,\theta,p^g,q^g} \quad \sum_{i\in\mathcal{N}} \left\{ a_{2i} \left(p_i^g\right)^2 + a_{1i}\, p_i^g + a_{0i} \right\} \tag{B.25a}$$

$$\text{s.t.} \quad \theta_{\text{ref}} = 0 \tag{B.25b}$$

$$p_i^g - p_i^l = v_i \sum_{j\in\mathcal{N}_T} v_j \left(G_{ij}\cos\theta_{ij} + B_{ij}\sin\theta_{ij}\right), \qquad \forall i \in \mathcal{N}, \tag{B.25c}$$

$$q_i^g - q_i^l = v_i \sum_{j\in\mathcal{N}_T} v_j \left(G_{ij}\sin\theta_{ij} - B_{ij}\cos\theta_{ij}\right), \qquad \forall i \in \mathcal{N}, \tag{B.25d}$$

$$p_{ij}^2 + q_{ij}^2 \le \overline{s}_{ij}^2, \qquad \forall (i,j) \in \mathcal{L}\cup\mathcal{L}^R, \tag{B.25e}$$

$$-\theta_{ij}^{\max} \le \theta_{ij} \le \theta_{ij}^{\max}, \qquad \forall (i,j) \in \mathcal{L}, \tag{B.25f}$$

$$\underline{v}_i \le v_i \le \overline{v}_i, \underline{p}_i^g \le p_i^g \le \overline{p}_i^g, \underline{q}_i^g \le q_i^g \le \overline{q}_i^g, \qquad \forall i \in \mathcal{N}, \tag{B.25g}$$

with power flow in branch $(i,j) \in \mathcal{L}$

$$p_{ij} = \ v_i^2 G_{ij} - v_i v_j \left(G_{ij}\cos\theta_{ij} + B_{ij}\sin\theta_{ij}\right), \qquad \forall (i,j) \in \mathcal{L}\cup\mathcal{L}^R, \tag{B.25h}$$

$$q_{ij} = -v_i^2 B_{ij} - v_i v_j \left(G_{ij}\sin\theta_{ij} - B_{ij}\cos\theta_{ij}\right), \qquad \forall (i,j) \in \mathcal{L}\cup\mathcal{L}^R, \tag{B.25i}$$

Alternatively, the formulation in rectangular coordinates is:

$$\min_{v,\theta,p^g,q^g} \quad \sum_{i\in\mathcal{N}} \left\{ a_{2i}\,(p_i^g)^2 + a_{1i}\,p_i^g + a_{0i} \right\} \tag{B.26a}$$

$$\text{s.t.} \quad w_{\text{ref}} = 0 \tag{B.26b}$$

$$p_i^g - p_i^d = \sum_{k\in\mathcal{N}} G_{ik}(u_i u_k + w_i w_k) + B_{ik}(w_i u_k - u_i w_k), \qquad \forall i \in \mathcal{N}, \tag{B.26c}$$

$$q_i^g - q_i^d = \sum_{k\in\mathcal{N}_\ell} G_{ik}(w_i u_k - u_i w_k) - B_{ik}(u_i u_k + w_i w_k), \qquad \forall i \in \mathcal{N}, \tag{B.26d}$$

$$p_{ij}^2 + q_{ij}^2 \le \overline{s}_{ij}^2, \qquad \forall (i,j) \in \mathcal{L} \cup \mathcal{L}^R, \tag{B.26e}$$

$$-\theta_{ij}^{\max} \le \theta_{ij} \le \theta_{ij}^{\max}, \qquad \forall (i,j) \in \mathcal{L}, \tag{B.26f}$$

$$\underline{v}_i \le v_i \le \overline{v}_i, \underline{p}_i^g \le p_i^g \le \overline{p}_i^g, \underline{q}_i^g \le q_i^g \le \overline{q}_i^g, \qquad \forall i \in \mathcal{N}, \tag{B.26g}$$

with power flow in branch $(i,j) \in \mathcal{L}$

$$p_{ij} = G_{ij}\{u_i(u_i - u_j) + w_i(w_i - w_j)\} + B_{ij}(u_i w_j - w_i u_j), \quad \forall (i,j) \in \mathcal{L} \cup \mathcal{L}^R, \tag{B.26h}$$

$$q_{ij} = B_{ij}\{u_i(u_j - u_i) + w_i(w_j - w_i)\} + G_{ij}(u_i w_j - w_i u_j), \quad \forall (i,j) \in \mathcal{L} \cup \mathcal{L}^R. \tag{B.26i}$$

Both polar (B.25) and rectangular (B.26) formulations can be expressed as a generic nonlinear programming (NLP) problem:

$$\min_{x\in\mathbb{R}^{n_x}} \quad f(x) \tag{B.27a}$$

$$\text{s.t.} \quad c^{\text{E}}(x) = 0 \tag{B.27b}$$

$$c^{\text{I}}(x) \le 0 \tag{B.27c}$$

or equivalently:

$$\min_{x\in\mathbb{R}^{n_x}} \quad f(x) \tag{B.28a}$$

$$c(x) \le 0 \tag{B.28b}$$

The second form (B.28) is often preferred in active-set methods, as it treats equality and inequality constraints equivalently during active-set detection, as discussed in Section 2.3.1.

List of Figures

List of Tables

References

[1] A. G. Givisiez, K. Petrou, and L. F. Ochoa, "A review on TSO-DSO coordination models and solution techniques," *Electric Power Systems Research*, vol. 189, p. 106659, 2020.

[2] H. Gerard, E. I. R. Puente, and D. Six, "Coordination between transmission and distribution system operators in the electricity sector: A conceptual framework," *Utilities Policy*, vol. 50, pp. 40–48, 2018.

[3] P. De Martini, L. Kristov, and L. Schwartz, "Distribution systems in a high distributed energy resources future: planning, market design, operation and oversight," *Future Electric Utility Regulation series. Lawrence Berkeley National Laboratory*, 2015.

[4] D. K. Molzahn, F. Dörfler, H. Sandberg, S. H. Low, S. Chakrabarti, R. Baldick, and J. Lavaei, "A survey of distributed optimization and control algorithms for electric power systems," *IEEE Transactions on Smart Grid*, vol. 8, no. 6, pp. 2941–2962, 2017.

[5] N. Patari, V. Venkataramanan, A. Srivastava, D. K. Molzahn, N. Li, and A. Annaswamy, "Distributed optimization in distribution systems: Use cases, limitations, and research needs," *IEEE Transactions on Power Systems*, vol. 37, no. 5, pp. 3469–3481, 2021.

[6] T. Mühlpfordt, **X. Dai**, A. Engelmann, and V. Hagenmeyer, "Distributed power flow and distributed optimization—formulation, solution, and open source implementation," *Sustainable Energy, Grids and Networks*, vol. 26, p. 100471, 2021.

[7] D. Bertsekas and J. Tsitsiklis, *Parallel and distributed computation: numerical methods*. Athena Scientific, 2015.

[8] J. N. Tsitsiklis, *Problems in decentralized decision making and computation*. PhD thesis, Massachusetts Institute of Technology, 1984.

[9] J. Tsitsiklis, D. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Transactions on Automatic Control*, vol. 31, no. 9, pp. 803–812, 1986.

[10] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[11] A. Nedić and J. Liu, "Distributed optimization for control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, no. 1, pp. 77–103, 2018.

[12] T. Yang, X. Yi, J. Wu, Y. Yuan, D. Wu, Z. Meng, Y. Hong, H. Wang, Z. Lin, and K. H. Johansson, "A survey of distributed optimization," *Annual Reviews in Control*, vol. 47, pp. 278–305, 2019.

[13] K. Tang, S. Dong, Y. Liu, L. Wang, and Y. Song, "Asynchronous distributed global power flow method for transmission–distribution coordinated analysis considering communication conditions," *Electric Power Systems Research*, vol. 182, p. 106256, 2020.

[14] **X. Dai**, Y. Lian, Y. Jiang, C. N. Jones, and V. Hagenmeyer, "Hypergraph-based fast distributed AC power flow optimization," in *62rd IEEE Conference on Decision and Control (CDC)*, 2023.

[15] J. Zhao, Z. Zhang, J. Yao, S. Yang, and K. Wang, "A distributed optimal reactive power flow for global transmission and distribution network," *International Journal of Electrical Power & Energy Systems*, vol. 104, no. JAN., pp. 524–536, 2019.

[16] Z. Li, Q. Guo, H. Sun, and J. Wang, "Coordinated transmission and distribution AC optimal power flow," *IEEE Transactions on Smart Grid*, vol. 9, no. 2, pp. 1228–1240, 2018.

[17] A. Mohammadi, M. Mehrtash, and A. Kargarian, "Diagonal quadratic approximation for decentralized collaborative TSO+DSO optimal power flow," *IEEE Transactions on Smart Grid*, vol. 10, no. 3, pp. 2358–2370, 2019.

[18] C. Lin, W. Wu, and M. Shahidehpour, "Decentralized AC optimal power flow for integrated transmission and distribution grids," *IEEE Transactions on Smart Grid*, vol. 11, no. 3, pp. 2531–2540, 2020.

[19] J. Zhai, Y. Jiang, Y. Shi, C. Jones, and X. Zhang, "Distributionally robust joint chance-constrained dispatch for integrated transmission-distribution systems via distributed optimization," *IEEE Transactions on Smart Grid*, 2022.

[20] Z. Chen, Z. Li, C. Guo, J. Wang, and Y. Ding, "Fully distributed robust reserve scheduling for coupled transmission and distribution systems," *IEEE Transactions on Power Systems*, vol. 36, no. 1, pp. 169–182, 2021.

[21] R. Roofegari nejad, W. Sun, and A. Golshani, "Distributed restoration for integrated transmission and distribution systems with DERs," *IEEE Transactions on Power Systems*, vol. 34, no. 6, pp. 4964–4973, 2019.

[22] J. Zhao, H. Wang, Y. Liu, Q. Wu, Z. Wang, and Y. Liu, "Coordinated restoration of transmission and distribution system using decentralized scheme," *IEEE Transactions on Power Systems*, vol. 34, no. 5, pp. 3428–3442, 2019.

[23] A. Al-Tawaha, E. Cibaku, S. Park, J. Lavaei, and M. Jin, "Distributed optimization and learning: A paradigm shift for power systems," 2023.

[24] K. Lehmann, A. Grastien, and P. Van Hentenryck, "AC-feasibility on tree networks is NP-Hard," *IEEE Transactions on Power Systems*, vol. 31, no. 1, pp. 798–801, 2015.

[25] D. Bienstock and A. Verma, "Strong NP-Hardness of AC power flows feasibility," *Operations Research Letters*, vol. 47, no. 6, pp. 494–501, 2019.

[26] G. Hug-Glanzmann and G. Andersson, "Decentralized optimal power flow control for overlapping areas in power systems," *IEEE Transactions on Power Systems*, vol. 24, no. 1, pp. 327–336, 2009.

[27] R. Baldick, B. Kim, C. Chase, and Y. Luo, "A fast distributed implementation of optimal power flow," *IEEE Transactions on Power Systems*, vol. 14, no. 3, pp. 858–864, 1999.

[28] X. He, Y. Zhao, and T. Huang, "Optimizing the dynamic economic dispatch problem by the distributed consensus-based ADMM approach," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 5, pp. 3210–3221, 2020.

[29] T. Erseghe, "Distributed optimal power flow using ADMM," *IEEE Transactions on Power Systems*, vol. 29, no. 5, pp. 2370–2380, 2014.

[30] B. Oh, D.-H. Lee, W.-C. Jeong, and D. Lee, "Distributed optimal power flow for distribution system using second order cone programming and consensus alternating direction method of multipliers," *Journal of Electrical Engineering & Technology*, vol. 17, no. 2, pp. 999–1008, 2022.

[31] B. Huang, Y. Li, F. Zhan, Q. Sun, and H. Zhang, "A distributed robust economic dispatch strategy for integrated energy system considering cyber-attacks," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 2, pp. 880–890, 2022.

[32] K. Sun and X. A. Sun, "A two-level ADMM algorithm for AC OPF with global convergence guarantees," *IEEE Transactions on Power Systems*, vol. 36, no. 6, pp. 5271–5281, 2021.

[33] B. Houska, J. V. Frasch, and M. Diehl, "An augmented lagrangian based algorithm for distributed nonconvex optimization," *SIAM Journal on Optimization*, vol. 26, no. 2, pp. 1101–1127, 2016.

[34] A. Engelmann, Y. Jiang, T. Mühlpfordt, B. Houska, T. Faulwasser, "Toward Distributed OPF Using ALADIN," *IEEE Transactions on Power Systems*, vol. 34, no. 1, pp. 584–594, 2019.

[35] N. Meyer-Huebner, M. Suriyah, and T. Leibfried, "Distributed optimal power flow in hybrid AC–DC grids," *IEEE Transactions on Power Systems*, vol. 34, no. 4, pp. 2937–2946, 2019.

[36] **X. Dai**, J. Zhai, Y. Jiang, Y. Guo, C. N. Jones, and V. Hagenmeyer, "Advancing distributed AC optimal power flow for integrated transmission-distribution systems," *IEEE Transactions on Network Science and Engineering*, vol. 12, no. 1, pp. 1210 – 1223, 2025.

[37] Y. Jiang*, **X. Dai***, P. Nimmegeers, B. Houska, and C. N. Jones, "ALADIN with enhanced globalization strategy for distributed nonconvex and nonsmooth optimization," 2025, under review.

[38] **X. Dai**, Y. Jiang, Y. Guo, C. N. Jones, M. Diehl, and V. Hagenmeyer, "Distributed AC optimal power flow: A scalable solution for large-scale problems," 2025, under review.

[39] **X. Dai**, Y. Cai, Y. Jiang, and V. Hagenmeyer, "Rapid scalable distributed power flow with open-source implementation," in *9th IFAC Conference on Networked Systems (NECSYS)*, 2022.

[40] **X. Dai***, A. Kocher*, J. Kovačević, B. Dindar, Y. Jiang, C. Jones, H. K. Çakmak, and V. Hagenmeyer, "Ensuring data privacy in AC optimal power flow with a distributed co-simulation framework," *Electric Power Systems Research,* Sepecial issue from Power System Computation Conference (PSCC), vol. 235, p. 110710, 2024.

[41] L. Hirth, J. Mühlenpfordt, and M. Bulkeley, "The ENTSO-E transparency platform–a review of Europe's most ambitious electricity data platform," *Applied Energy*, vol. 225, pp. 1054–1067, 2018.

[42] **X. Dai**, Y. Guo, Y. Jiang, C. N. Jones, G. Hug, and V. Hagenmeyer, "Real-time coordination of integrated transmission and distribution systems: Flexibility modeling and distributed NMPC scheduling," *Electric Power Systems Research,* Sepecial issue from Power System Computation Conference (PSCC), vol. 234, p. 110627, 2024.

[43] M. Baran and F. F. Wu, "Optimal sizing of capacitors placed on a radial distribution system," *IEEE Transactions on Power Delivery*, vol. 4, no. 1, pp. 735–743, 1989.

[44] M. Farivar, L. Chen, and S. Low, "Equilibrium and dynamics of local voltage control in distribution systems," in *52nd IEEE Conference on Decision and Control (CDC)*, pp. 4329–4334, 2013.

[45] D. Turizo and D. K. Molzahn, "Invertibility conditions for the admittance matrices of balanced power systems," *IEEE Transactions on Power Systems*, 2022.

[46] J. Nocedal and S. Wright, *Numerical optimization*. Springer Science & Business Media, 2006.

[47] B. Chachuat, "Nonlinear and dynamic optimization: From theory to practice," 2007.

[48] O. L. Mangasarian and S. Fromovitz, "The fritz john necessary optimality conditions in the presence of equality and inequality constraints," *Journal of Mathematical Analysis and applications*, vol. 17, no. 1, pp. 37–47, 1967.

[49] W. H. Press, *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge university press, 2007.

[50] P. E. Gill and E. Wong, "Sequential quadratic programming methods," in *Mixed integer nonlinear programming*, pp. 147–224, Springer, 2011.

[51] R. B. Wilson, "A simplicial algorithm for concave programming," *Ph. D. Dissertation, Graduate School of Bussiness Administration*, 1963.

[52] J. T. Betts and P. D. Frank, "A sparse nonlinear optimization algorithm," *Journal of Optimization Theory and Applications*, vol. 82, pp. 519–541, 1994.

[53] L. T. Biegler, J. Nocedal, C. Schmid, and D. Ternet, "Numerical experience with a reduced hessian method for large scale constrained optimization," *Computational Optimization and Applications*, vol. 15, pp. 45–67, 2000.

[54] P. T. Boggs, A. J. Kearsley, and J. W. Tolle, "A practical algorithm for general large scale nonlinear optimization problems," *SIAM Journal on Optimization*, vol. 9, no. 3, pp. 755–778, 1999.

[55] R. Fletcher and S. Leyffer, "Nonlinear programming without a penalty function," *Mathematical programming*, vol. 91, pp. 239–269, 2002.

[56] P. E. Gill, W. Murray, and M. A. Saunders, "Snopt: An SQP algorithm for large-scale constrained optimization," *SIAM review*, vol. 47, no. 1, pp. 99–131, 2005.

[57] F. E. Curtis, T. Mitchell, and M. L. Overton, "A BFGS-SQP method for nonsmooth, nonconvex, constrained optimization and its evaluation using relative minimization profiles," *Optimization Methods and Software*, vol. 32, no. 1, pp. 148–181, 2017.

[58] M. Diehl, A. Walther, H. G. Bock, and E. Kostina, "An adjoint-based SQP algorithm with quasi-newton jacobian updates for inequality constrained optimization," *Optimization Methods & Software*, vol. 25, no. 4, pp. 531–552, 2010.

[59] M. Diehl, "Lecture notes on numerical optimization," *University of Freiburg, Freiburg*, 2017.

[60] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, "qpoases: A parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, vol. 6, pp. 327–363, 2014.

[61] R. Schwan, Y. Jiang, D. Kuhn, and C. N. Jones, "Piqp: A proximal interior-point quadratic programming solver," in *62nd IEEE Conference on Decision and Control (CDC)*, pp. 1088–1093, IEEE, 2023.

[62] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: An operator splitting solver for quadratic programs," *Mathematical Programming Computation*, vol. 12, no. 4, pp. 637–672, 2020.

[63] B. O'donoghue, E. Chu, N. Parikh, and S. Boyd, "Conic optimization via operator splitting and homogeneous self-dual embedding," *Journal of Optimization Theory and Applications*, vol. 169, pp. 1042–1068, 2016.

[64] A. Bambade, S. El-Kazdadi, A. Taylor, and J. Carpentier, "Prox-QP: Yet another quadratic programming solver for robotics and beyond," in *RSS 2022-Robotics: Science and Systems*, 2022.

[65] R. H. Byrd, N. I. Gould, J. Nocedal, and R. A. Waltz, "An algorithm for nonlinear optimization using linear programming and equality constrained subproblems," *Mathematical Programming*, vol. 100, no. 1, pp. 27–48, 2003.

[66] A. Wächter and L. T. Biegler, "Line search filter methods for nonlinear programming: Local convergence," in *Technical Report*, Technical Report RC23033 (W0312-090), TJ Watson Research Center, Yorktown, 2003.

[67] M. Ulbrich, S. Ulbrich, and L. N. Vicente, "A globally convergent primal-dual interior-point filter method for nonlinear programming," *Mathematical Programming*, vol. 100, pp. 379–410, 2004.

[68] A. Wächter and L. T. Biegler, "Line search filter methods for nonlinear programming: Motivation and global convergence," *SIAM Journal on Optimization*, vol. 16, no. 1, pp. 1–31, 2005.

[69] C. Shen, S. Leyffer, and R. Fletcher, "A nonmonotone filter method for nonlinear optimization," *Computational Optimization and Applications*, vol. 52, pp. 583–607, 2012.

[70] D. Bertsekas, *Nonlinear Programming*. Athena Scientific, 1999.

[71] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.

[72] S. Mehrotra, "On the implementation of a primal-dual interior point method," *SIAM Journal on optimization*, vol. 2, no. 4, pp. 575–601, 1992.

[73] J. Nocedal, A. Wächter, and R. A. Waltz, "Adaptive barrier update strategies for nonlinear interior methods," *SIAM Journal on Optimization*, vol. 19, no. 4, pp. 1674–1693, 2009.

[74] J. L. Morales, J. Nocedal, R. A. Waltz, G. Liu, and J.-P. Goux, "Assessing the potential of interior methods for nonlinear optimization," in *Large-scale PDE-constrained Optimization*, pp. 167–183, Springer, 2003.

[75] F. E. Curtis, J. Huber, O. Schenk, and A. Wächter, "A note on the implementation of an interior-point algorithm for nonlinear optimization with inexact step computations," *Mathematical programming*, vol. 136, no. 1, pp. 209–227, 2012.

[76] A. Forsgren, P. E. Gill, and M. H. Wright, "Interior methods for nonlinear optimization," *SIAM review*, vol. 44, no. 4, pp. 525–597, 2002.

[77] N. Gould, D. Orban, and P. Toint, "Numerical methods for large-scale nonlinear optimization," *Acta Numerica*, vol. 14, pp. 299–361, 2005.

[78] R. J. Vanderbei, "Loqo: An interior point code for quadratic programming," *Optimization methods and software*, vol. 11, no. 1-4, pp. 451–484, 1999.

[79] R. H. Byrd, J. Nocedal, and R. A. Waltz, "Knitro: An integrated package for nonlinear optimization," *Large-scale nonlinear optimization*, pp. 35–59, 2006.

[80] K. Świrydowicz, E. Darve, W. Jones, J. Maack, S. Regev, M. A. Saunders, S. J. Thomas, and S. Peleš, "Linear solvers for power grid optimization problems: a review of GPU-accelerated linear solvers," *Parallel Computing*, vol. 111, p. 102870, 2022.

[81] S. Regev, N.-Y. Chiang, E. Darve, C. G. Petra, M. A. Saunders, K. Świrydowicz, and S. Peleš, "Hykkt: a hybrid direct-iterative method for solving KKT linear systems," *Optimization Methods and Software*, vol. 38, no. 2, pp. 332–355, 2023.

[82] S. Shin, M. Anitescu, and F. Pacaud, "Accelerating optimal power flow with GPUs: Simd abstraction of nonlinear programs and condensed-space interior-point methods," *Electric Power Systems Research*, vol. 236, p. 110651, 2024.

[83] N. I. Gould, "On practical conditions for the existence and uniqueness of solutions to the general equality quadratic programming problem," *Mathematical Programming*, vol. 32, no. 1, pp. 90–99, 1985.

[84] A. Forsgren, "Inertia-controlling factorizations for optimization algorithms," *Applied Numerical Mathematics*, vol. 43, no. 1-2, pp. 91–107, 2002.

[85] N.-Y. Chiang and V. M. Zavala, "An inertia-free filter line-search algorithm for large-scale nonlinear programming," *Computational Optimization and Applications*, vol. 64, pp. 327–354, 2016.

[86] G. B. Dantzig and P. Wolfe, "Decomposition principle for linear programs," *Operations research*, vol. 8, no. 1, pp. 101–111, 1960.

[87] J. BnnoBRs, "Partitioning procedures for solving mixed-variables programming problems," *Numer. Math*, vol. 4, no. 1, pp. 238–252, 1962.

[88] H. Everett III, "Generalized lagrange multiplier method for solving problems of optimum allocation of resources," *Operations research*, vol. 11, no. 3, pp. 399–417, 1963.

[89] M. R. Hestenes, "Multiplier and gradient methods," *Journal of optimization theory and applications*, vol. 4, no. 5, pp. 303–320, 1969.

[90] R. T. Rockafellar, "The multiplier method of hestenes and powell applied to convex programming," *Journal of Optimization Theory and applications*, vol. 12, no. 6, pp. 555–562, 1973.

[91] W. Deng, M.-J. Lai, Z. Peng, and W. Yin, "Parallel multi-block ADMM with o (1/k) convergence," *Journal of Scientific Computing*, vol. 71, pp. 712–736, 2017.

[92] C. Chen, B. He, Y. Ye, and X. Yuan, "The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent," *Mathematical Programming*, vol. 155, no. 1-2, pp. 57–79, 2016.

[93] J. Ma, X.-Y. Liu, Z. Shou, and X. Yuan, "Deep tensor admm-net for snapshot compressive imaging," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10223–10232, 2019.

[94] A. Ramdas and R. J. Tibshirani, "Fast and flexible admm algorithms for trend filtering," *Journal of Computational and Graphical Statistics*, vol. 25, no. 3, pp. 839–858, 2016.

[95] J. Sun, H. Li, Z. Xu, *et al.*, "Deep admm-net for compressive sensing mri," *Advances in neural information processing systems*, vol. 29, 2016.

[96] J. Guo, G. Hug, and O. K. Tonguz, "A case for nonconvex distributed optimization in large-scale power systems," *IEEE Transactions on Power Systems*, vol. 32, no. 5, pp. 3842–3851, 2017.

[97] R. H. Byrd, J. Nocedal, and R. B. Schnabel, "Representations of quasi-newton matrices and their use in limited memory methods," *Mathematical Programming*, vol. 63, no. 1, pp. 129–156, 1994.

[98] A. R. Conn, N. I. Gould, and P. L. Toint, *Trust region methods*. SIAM, 2000.

[99] M. Hong, Z.-Q. Luo, and M. Razaviyayn, "Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems," *SIAM Journal on Optimization*, vol. 26, no. 1, pp. 337–364, 2016.

[100] A. Engelmann, T. Mühlpfordt, Y. Jiang, B. Houska, and T. Faulwasser, "Distributed AC optimal power flow using ALADIN," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 5536–5541, 2017.

[101] J. Zhai*, **X. Dai***, Y. Jiang, Y. Xue, V. Hagenmeyer, C. Jones, and X.-P. Zhang, "Distributed optimal power flow for VSC-MTDC meshed AC/DC grids using ALADIN," *IEEE Transactions on Power Systems*, pp. 1–1, 2022.

[102] E. Anderson, Z. Bai, C. Bischof, L. S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, *et al.*, *LAPACK users' guide*. SIAM, 1999.

[103] A. Engelmann, Y. Jiang, B. Houska, and T. Faulwassser, "Decomposition of non-convex optimization via bi-level distributed ALADIN," *IEEE Control Netw. Syst.*, vol. 7, p. 1848–1858, 2020.

[104] B. Houska and Y. Jiang, "Distributed optimization and control with ALADIN," *Recent Advances in Model Predictive Control: Theory, Algorithms, and Applications*, p. 135–163, 2021.

[105] J. J. Grainger, *Power system analysis*. McGraw-Hill, 1999.

[106] A. X. Sun, D. T. Phan, and S. Ghosh, "Fully decentralized AC optimal power flow algorithms," in *2013 IEEE Power & Energy Society General Meeting*, pp. 1–5, IEEE, 2013.

[107] H. K. Çakmak and V. Hagenmeyer, "Using open data for modeling and simulation of the all electrical society in eASiMOV," in *2022 Open Source Modelling and Simulation of Energy Systems (OSMSES)*, pp. 1–6, IEEE, 2022.

[108] T. Erseghe, "A distributed approach to the OPF problem," *EURASIP J. Adv. Signal Process*, vol. 2015, no. 1, pp. 1–13, 2015.

[109] A. Engelmann, Y. Jiang, H. Benner, R. Ou, B. Houska, and T. Faulwasser, "ALADIN-α – an open-source MATLAB toolbox for distributed non-convex optimization," *Optimal Control Applications and Methods*, vol. 43, no. 1, pp. 4–22, 2022.

[110] S. Frank and S. Rebennack, "An introduction to optimal power flow: Theory, formulation, and examples," *IIE Transactions*, vol. 48, no. 12, pp. 1172–1197, 2016.

[111] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "Casadi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Comput."*, vol. 11, no. 1, pp. 1–36, 2019.

[112] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "Matpower: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 12–19, 2010.

[113] C. Gomes, C. Thule, D. Broman, P. G. Larsen, and H. Vangheluwe, "Co-simulation: a survey," *ACM Computing Surveys (CSUR)*, vol. 51, no. 3, pp. 1–33, 2018.

[114] A. Erdmann, H. K. Çakmak, U. Kühnapfel, and V. Hagenmeyer, "A new communication concept for efficient configuration of energy systems integration co-simulation," in *2019 IEEE/ACM 23rd International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*, pp. 1–8, IEEE, 2019.

[115] A. Erdmann, A. Marcellan, J. Stock, D. Neuroth, C. Utama, M. Suriyah, S. Steinle, F. Müller, D. Hering, H. Francke, S. Gritzbach, M. Henke, N. Pflugradt, H. Çakmak, L. Kotzur, D. Stolten, T. Leibfried, D. Müller, R. Schlatmann, A. Xhonneux, V. Hagenmeyer, and C. Ulbrich, "A comparative co-simulation analysis to improve the sustainability of cogeneration-based district multi-energy systems using photovoltaics, power-to-heat, and heat storage," *Energy Conversion and Management*, vol. 291, p. 117226, 2023.

[116] S. Babaeinejadsarookolaee, A. Birchfield, and et al, "The power grid library for benchmarking AC optimal power flow algorithms," *arXiv preprint arXiv:1908.02788*, 2019.

[117] A. Parisio, J. Schiffer, and C. A. Hans, *System Level Control and Optimisation of Microgrids*. The Institution of Engineering and Technology, 2024.

[118] Q. Wang, C. Lin, W. Wu, B. Wang, G. Wang, H. Liu, H. Zhang, and J. Zhang, "A nested decomposition method for the AC optimal power flow of hierarchical electrical power grids," *IEEE Transactions on Power Systems*, 2022.

[119] W. Zheng, W. Wu, B. Zhang, H. Sun, and Y. Liu, "A fully distributed reactive power optimization and control method for active distribution networks," *IEEE Transactions on Smart Grid*, vol. 7, no. 2, pp. 1021–1033, 2015.

[120] Q. Peng and S. H. Low, "Distributed optimal power flow algorithm for radial networks, i: Balanced single phase case," *IEEE Transactions on Smart Grid*, vol. 9, no. 1, pp. 111–121, 2016.

[121] A. Rajaei, S. Fattaheian-Dehkordi, M. Fotuhi-Firuzabad, and M. Moeini-Aghtaie, "Decentralized transactive energy management of multi-microgrid distribution systems based on admm," *International Journal of Electrical Power & Energy Systems*, vol. 132, p. 107126, 2021.

[122] H. Sun, Q. Guo, B. Zhang, Y. Guo, Z. Li, and J. Wang, "Master–slave-splitting based distributed global power flow method for integrated transmission and distribution analysis," *IEEE Transactions on Smart Grid*, vol. 6, no. 3, pp. 1484–1492, 2014.

[123] S. Tu, A. Wächter, and E. Wei, "A two-stage decomposition approach for AC optimal power flow," *IEEE Transactions on Power Systems*, vol. 36, no. 1, pp. 303–312, 2020.

[124] A. Engelmann, G. Stomberg, and T. Faulwasser, "An essentially decentralized interior point method for control," in *60th IEEE Conference on Decision and Control (CDC)*, pp. 2414–2420, IEEE, 2021.

[125] S. Mhanna, G. Verbič, and A. C. Chapman, "Adaptive ADMM for distributed AC optimal power flow," *IEEE Transactions on Power Systems*, vol. 34, no. 3, pp. 2025–2035, 2018.

[126] K. Sun and X. A. Sun, "A two-level distributed algorithm for nonconvex constrained optimization," *Computational Optimization and Applications*, vol. 84, no. 2, pp. 609–649, 2023.

[127] A. Engelmann, Y. Jiang, B. Houska, and T. Faulwasser, "Decomposition of nonconvex optimization via bi-level distributed ALADIN," *IEEE Transactions on Control of Network Systems*, vol. 7, no. 4, pp. 1848–1858, 2020.

[128] J. Guo, G. Hug, and O. K. Tonguz, "Intelligent partitioning in distributed optimization of electric power systems," *IEEE Transactions on Smart Grid*, vol. 7, no. 3, pp. 1249–1258, 2015.

[129] F. Zhang, *The Schur complement and its applications*, vol. 4. Springer Science & Business Media, 2006.

[130] R. H. Byrd, G. Liu, and J. Nocedal, "On the local behavior of an interior point method for nonlinear programming," *Numerical analysis*, vol. 1997, pp. 37–56, 1997.

[131] E. V. Haynsworth, "Determination of the inertia of a partitioned hermitian matrix," *Linear algebra and its applications*, vol. 1, no. 1, pp. 73–81, 1968.

[132] A. Murray, M. Kyesswa, P. Schmurr, H. Çakmak, and V. Hagenmeyer, "On grid partitioning in AC optimal power flow," in *2020 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe)*, (The Hague, Netherlands), pp. 524–528, IEEE, 2020.

[133] A. Murray, M. Kyesswa, H. Çakmak, and V. Hagenmeyer, "On optimal grid partitioning for distributed optimization of reactive power dispatch," in *2021 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, (Washington, DC, USA), pp. 1–5, IEEE, 2021.

[134] P. Sanders and C. Schulz, "Engineering multilevel graph partitioning algorithms," in *Algorithms–ESA 2011. 19th Annual European Symposium, Saarbrücken, Germany, September 5-9, 2011. Proceedings. Ed.: C. Demetrescu*, (Saarbrücken, Germany), p. 469, Springer Berlin Heidelberg, 2011.

[135] P. Sanders and C. Schulz, "Think Locally, Act Globally: Highly Balanced Graph Partitioning," in *Proceedings of the 12th International Symposium on Experimental Algorithms (SEA'13)*, vol. 7933 of *LNCS*, pp. 164–175, Springer, 2013.

[136] A. Migdalas, P. M. Pardalos, and S. Storøy, *Parallel computing in optimization*, vol. 7. Springer Science & Business Media, 2013.

[137] R. B. Schnabel, "A view of the limitations, opportunities, and challenges in parallel nonlinear optimization," *Parallel computing*, vol. 21, no. 6, pp. 875–905, 1995.

[138] M. Colombo, A. Grothey, J. Hogg, K. Woodsend, and J. Gondzio, "A structure-conveying modelling language for mathematical and stochastic programming," *Mathematical Programming Computation*, vol. 1, pp. 223–247, 2009.

[139] J.-P. Watson, D. L. Woodruff, and W. E. Hart, "Pysp: modeling and solving stochastic programs in python," *Mathematical Programming Computation*, vol. 4, pp. 109–149, 2012.

[140] J. S. Rodriguez, R. B. Parker, C. D. Laird, B. L. Nicholson, J. D. Siirola, and M. L. Bynum, "Scalable parallel nonlinear optimization with pynumero and parapint," *INFORMS Journal on Computing*, vol. 35, no. 2, pp. 509–517, 2023.

[141] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, *et al.*, "Jax: composable transformations of python+ numpy programs," 2018.

[142] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[143] F. Pacaud, M. Schanen, S. Shin, D. A. Maldonado, and M. Anitescu, "Parallel interior-point solver for block-structured nonlinear programs on SIMD/GPU architectures," *Optimization Methods and Software*, pp. 1–24, 2024.

[144] J. Kardoš, D. Kourounis, O. Schenk, and R. Zimmerman, "BELTISTOS: A robust interior point method for large-scale optimal power flow problems," *Electric power systems research*, vol. 212, p. 108613, 2022.

[145] I. S. Duff, "Ma57—a code for the solution of sparse symmetric definite and indefinite systems," *ACM Transactions on Mathematical Software (TOMS)*, vol. 30, no. 2, pp. 118–144, 2004.

[146] S. Fliscounakis, P. Panciatici, F. Capitanescu, and L. Wehenkel, "Contingency ranking with respect to overloads in very large power systems taking into account uncertainty, preventive, and corrective actions," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 4909–4917, 2013.

[147] C. Josz, S. Fliscounakis, J. Maeght, and P. Panciatici, "AC power flow data in matpower and qcqp format: itesla, rte snapshots, and pegase," *arXiv preprint arXiv:1603.01533*, 2016.

[148] I. Aravena, D. K. Molzahn, S. Zhang, C. G. Petra, F. E. Curtis, S. Tu, A. Wächter, E. Wei, E. Wong, A. Gholami, *et al.*, "Recent developments in security-constrained AC optimal power flow: Overview of challenge 1 in the arpa-e grid optimization competition," *Operations research*, vol. 71, no. 6, pp. 1997–2014, 2023.

[149] J. M. Snodgrass, *Tractable algorithms for constructing electric power network models*. The University of Wisconsin-Madison, 2021.

[150] Y. Wen, Z. Hu, and L. Liu, "Aggregate temporally coupled power flexibility of ders considering distribution system security constraints," *IEEE Transactions on Power Systems*, 2022.

[151] A. Lorca and X. A. Sun, "The adaptive robust multi-period alternating current optimal power flow problem," *IEEE Transactions on Power Systems*, vol. 33, no. 2, pp. 1993–2003, 2017.

[152] D. Kourounis, A. Fuchs, and O. Schenk, "Toward the next generation of multiperiod optimal power flow solvers," *IEEE Transactions on Power Systems*, vol. 33, no. 4, pp. 4005–4014, 2018.

[153] A. Agarwal and L. Pileggi, "Large scale multi-period optimal power flow with energy storage systems using differential dynamic programming," *IEEE Transactions on Power Systems*, vol. 37, no. 3, pp. 1750–1759, 2021.

[154] Y. Wen, Z. Hu, J. He, and Y. Guo, "Improved inner approximation for aggregating power flexibility in active distribution networks and its applications," 2023.

[155] M. Zafeiropoulou, N. Sijakovic, M. Zarkovic, V. Ristic, A. Terzic, D. Makrygiorgou, E. Zoulias, V. Vita, T. I. Maris, and G. Fotis, "Development and implementation of a flexibility platform for active system management at both transmission and distribution level in greece," *Applied Sciences*, vol. 13, no. 20, p. 11248, 2023.

[156] M. E. Baran and F. F. Wu, "Network reconfiguration in distribution systems for loss reduction and load balancing," *IEEE Transactions on Power Delivery*, vol. 4, no. 2, pp. 1401–1407, 1989.

[157] Y. Jiang, D. Kouzoupis, H. Yin, M. Diehl, and B. Houska, "Decentralized optimization over tree graphs," *Journal of Optimization Theory and Applications*, vol. 189, pp. 384–407, 2021.

[158] R. Bauer, **X. Dai**, and V. Hagenmeyer, "A shapley value-based distributed AC OPF approach for redispatch congestion cost allocation," in *Proceedings of the 14th ACM International Conference on Future Energy Systems (e-Energy)*, pp. 109–113, 2023.

[159] K. Baker, J. Guo, G. Hug, and X. Li, "Distributed MPC for efficient coordination of storage and renewable energy sources across control areas," *IEEE Transactions on Smart Grid*, vol. 7, no. 2, pp. 992–1001, 2016.

[160] D. B. West *et al.*, *Introduction to graph theory*, vol. 2. Prentice hall Upper Saddle River, 2001.

[161] A. Bemporad, C. Filippi, and F. D. Torrisi, "Inner and outer approximations of polytopes using boxes," *Computational Geometry*, vol. 27, no. 2, pp. 151–178, 2004.

[162] Y. Jiang, P. Sauerteig, B. Houska, and K. Worthmann, "Distributed optimization using ALADIN for MPC in smart grids," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 5, pp. 2142–2152, 2020.

[163] T. Gonen, *Modern power system analysis*. CRC Press, 2016.

[164] P. Kundur, "Power system stability," *Power system stability and control*, vol. 10, pp. 7–1, 2007.

[165] C. Coffrin and P. Van Hentenryck, "A linear-programming approximation of AC power flows," *INFORMS Journal on Computing*, vol. 26, no. 4, pp. 718–734, 2014.

[166] H. Hijazi, C. Coffrin, and P. V. Hentenryck, "Convex quadratic relaxations for mixed-integer nonlinear programs in power systems," *Mathematical Programming Computation*, vol. 9, pp. 321–367, 2017.

[167] R. Madani, S. Sojoudi, and J. Lavaei, "Convex relaxation for optimal power flow problem: Mesh networks," *IEEE Transactions on Power Systems*, vol. 30, no. 1, pp. 199–211, 2014.

[168] A. J. Korsak, "On the question of uniqueness of stable load-flow solutions," *IEEE Transactions on Power Apparatus and Systems*, no. 3, pp. 1093–1100, 1972.

[169] W. A. Bukhsh, A. Grothey, K. I. McKinnon, and P. A. Trodden, "Local solutions of the optimal power flow problem," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 4780–4788, 2013.

[170] L. Li, T. Kappler, B. Schwarz, N. Munzke, **X. Dai**[§], V. Hagenmeyer, and M. Hiller, "Optimal BESS management for peak load shaving and battery health under prediction uncertainty," *IEEE Transactions on Sustainable Energy*, 2025.

[171] D. Song, L. Yan, **X. Dai**, X. Zhu, V. Hagenmeyer, and J. Zhai, "Low-carbon energy management for networked multi-energy microgrids using multi-agent soft actor-critic algorithm," *Sustainable Energy, Grids and Networks*, p. 101821, 2025.

[172] **X. Dai**, Y. Jiang, F. Zahn, Y. Guo, and V. Hagenmeyer, "Distributed coordination for transmission-distribution systems with nonlinear flexibility aggregation," 2025, under review.

[173] **X. Dai**, Y. Jiang, J. Zhai, J. Liu, T. Schlachter, C. Jones, X.-P. Zhang, and V. Hagenmeyer, "Distributed convex second-order optimization for AC/DC hybrid distribution systems within PROcess Orchestration Framework," 2025, under review.

[174] Y. Jiang, **X. Dai**[§], F. Zahn, Y. Guo, and V. Hagenmeyer, "Error accumulation using linearized models for aggregating flexibility in distribution systems," in *IEEE Power & Energy Society International Meeting (PESIM)*, 2026, accepted.

[175] Y. Jiang, **X. Dai**[§], F. Zahn, and V. Hagenmeyer, "Enhanced flexibility aggregation using lindistflow model with loss compensation," in *IEEE PowerTech*, 2025.

[176] R. Bauer, **X. Dai**, F. Zahn, and V. Hagenmeyer, "Shapley value-based cost allocation for battery energy storage systems in power grids with a high share of renewables," in *Proceedings of the 16th ACM International Conference on Future and Sustainable Energy Systems*, pp. 650–655, 2025.

[177] J. You, **X. Dai**, Y. Jiang, H. Yin, Y. Shi, and C. N. Jones, "Real-time pricing mechanism for V2G using distributed bilevel optimization," in *63rd IEEE Conference on Decision and Control (CDC)*, 2024.

[178] R. Bauer, **X. Dai**, and V. Hagenmeyer, "Industrial application of the shapley value-based redispatch cost allocation to large-scale power grids requires AC optimal power flow," in *IEEE Power & Energy Society General Meeting (PESGM)*, 2024.

Publications

Journal Articles

Peer-Reviewed Articles:

[36] **X. Dai**, J. Zhai, Y. Jiang, Y. Guo, C. N. Jones, and V. Hagenmeyer, "Advancing distributed AC optimal power flow for integrated transmission-distribution systems," *IEEE Transactions on Network Science and Engineering*, vol. 12, no. 1, pp. 1210 – 1223, 2025

[170] L. Li, T. Kappler, B. Schwarz, N. Munzke, **X. Dai**§, V. Hagenmeyer, and M. Hiller, "Optimal BESS management for peak load shaving and battery health under prediction uncertainty," *IEEE Transactions on Sustainable Energy*, 2025

[171] D. Song, L. Yan, **X. Dai**, X. Zhu, V. Hagenmeyer, and J. Zhai, "Low-carbon energy management for networked multi-energy microgrids using multi-agent soft actor-critic algorithm," *Sustainable Energy, Grids and Networks*, p. 101821, 2025

[42] **X. Dai**, Y. Guo, Y. Jiang, C. N. Jones, G. Hug, and V. Hagenmeyer, "Real-time coordination of integrated transmission and distribution systems: Flexibility modeling and distributed NMPC scheduling," *Electric Power Systems Research,* Sepecial issue from Power System Computation Conference (PSCC), vol. 234, p. 110627, 2024

[40] **X. Dai**⋆, A. Kocher⋆, J. Kovačević, B. Dindar, Y. Jiang, C. Jones, H. K. Çakmak, and V. Hagenmeyer, "Ensuring data privacy in AC optimal power flow with a distributed co-simulation framework," *Electric Power Systems Research,* Sepecial issue from Power System Computation Conference (PSCC), vol. 235, p. 110710, 2024

[101] J. Zhai⋆, **X. Dai**⋆, Y. Jiang, Y. Xue, V. Hagenmeyer, C. Jones, and X.-P. Zhang, "Distributed optimal power flow for VSC-MTDC meshed AC/DC grids using ALADIN," *IEEE Transactions on Power Systems*, pp. 1–1, 2022

Note: ⋆ denotes equal contribution; § denotes the corresponding author.

[6] T. Mühlpfordt, **X. Dai**, A. Engelmann, and V. Hagenmeyer, "Distributed power flow and distributed optimization—formulation, solution, and open source implementation," *Sustainable Energy, Grids and Networks*, vol. 26, p. 100471, 2021

Articles Under Review:

[37] **X. Dai**, Y. Jiang, Y. Guo, C. N. Jones, M. Diehl, and V. Hagenmeyer, "Distributed AC optimal power flow: A scalable solution for large-scale problems," 2025, under review

[172] **X. Dai**, Y. Jiang, F. Zahn, Y. Guo, and V. Hagenmeyer, "Distributed coordination for transmission-distribution systems with nonlinear flexibility aggregation," 2025, under review

[173] **X. Dai**, Y. Jiang, J. Zhai, J. Liu, T. Schlachter, C. Jones, X.-P. Zhang, and V. Hagenmeyer, "Distributed convex second-order optimization for AC/DC hybrid distribution systems within PROcess Orchestration Framework," 2025, under review

[38] Y. Jiang*, **X. Dai***, P. Nimmegeers, B. Houska, and C. N. Jones, "ALADIN with enhanced globalization strategy for distributed nonconvex and nonsmooth optimization," 2025, under review

Conference Papers

Peer-Reviewed Articles:

[174] Y. Jiang, **X. Dai**§, F. Zahn, Y. Guo, and V. Hagenmeyer, "Error accumulation using linearized models for aggregating flexibility in distribution systems," in *IEEE Power & Energy Society International Meeting (PESIM)*, 2026, accepted

[175] Y. Jiang, **X. Dai**§, F. Zahn, and V. Hagenmeyer, "Enhanced flexibility aggregation using lindistflow model with loss compensation," in *IEEE PowerTech*, 2025

[176] R. Bauer, **X. Dai**, F. Zahn, and V. Hagenmeyer, "Shapley value-based cost allocation for battery energy storage systems in power grids with a high share of renewables," in *Proceedings of the 16th ACM International Conference on Future and Sustainable Energy Systems*, pp. 650–655, 2025

Note: * denotes equal contribution; § denotes the corresponding author.

[177] J. You, **X. Dai**, Y. Jiang, H. Yin, Y. Shi, and C. N. Jones, "Real-time pricing mechanism for V2G using distributed bilevel optimization," in *63rd IEEE Conference on Decision and Control (CDC)*, 2024

[178] R. Bauer, **X. Dai**, and V. Hagenmeyer, "Industrial application of the shapley value-based redispatch cost allocation to large-scale power grids requires AC optimal power flow," in *IEEE Power & Energy Society General Meeting (PESGM)*, 2024

[14] **X. Dai**, Y. Lian, Y. Jiang, C. N. Jones, and V. Hagenmeyer, "Hypergraph-based fast distributed AC power flow optimization," in *62rd IEEE Conference on Decision and Control (CDC)*, 2023

[158] R. Bauer, **X. Dai**, and V. Hagenmeyer, "A shapley value-based distributed AC OPF approach for redispatch congestion cost allocation," in *Proceedings of the 14th ACM International Conference on Future Energy Systems (e-Energy)*, pp. 109–113, 2023

[39] **X. Dai**, Y. Cai, Y. Jiang, and V. Hagenmeyer, "Rapid scalable distributed power flow with open-source implementation," in *9th IFAC Conference on Networked Systems (NECSYS)*, 2022