10-21-2020

Parallel and Asynchronous Distributed Optimization for Power Systems Operation

Ali Mohammadi

PARALLEL AND ASYNCHRONOUS DISTRIBUTED OPTIMIZATION FOR POWER SYSTEMS OPERATION

A Dissertation

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

in

The Division of Electrical and Computer Engineering

by
Ali Mohammadi
M.S., Shiraz University of Technology, Iran, 2014
B.S., Shahid Bahonar University of Kerman, Iran, 2012
December 2020

ACKNOWLEDGEMENTS

This dissertation becomes a reality with the kind support and help of many individuals. I would like to extend my sincere thanks to all of them. Foremost, I pay my deep sense of gratitude to my major advisor, Dr. Amin Kargarian, who enlightened and guided me during my Ph.D. study and this dissertation research. My dissertation would not have been achieved and completed without his aid and guidance.

Also, I would like to thank Prof. Shahab Mehraeen, Prof. Hsiao-Chun Wu, Dr. Mehdi Farasat, and Prof. Kevin McCarter for serving as my dissertation committee members and providing me with valuable advice. My appreciation also extends to all ECE staff and faculty members, especially Dr. Jerry Trahan as the chair of the Division of Electrical and Computer Engineering, Mr. Christopher Pearson and Ms. Beth Cochran for being always kind and helpful.

Last but not least, I am most grateful to my beloved parents, thoughtful brothers, and my lovely wife. They are always behind me and being my strong backing. This dissertation could be dedicated to them as a gift for their everlasting love to me.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS v

ABSTRACT v

CHAPTER 1. INTRODUCTION 1
- 1.1 Background 1
- 1.2 Motivation and Literature Review 2
- 1.3 Contribution and Organization 9

CHAPTER 2. DIAGONAL QUADRATIC APPROXIMATION FOR DECENTRALIZED COLLABORATIVE TSO+DSO OPTIMAL POWER FLOW 13
- 2.1 Introduction 13
- 2.2 Contribution 15
- 2.3 Decentralized ATC-based OPF Implementation 16
- 2.4 Diagonal Quadratic Approximation Method for Parallel Solution 24
- 2.5 Numerical Results 31
- 2.6 Conclusion 45

CHAPTER 3. ACCELERATED AND ROBUST ANALYTICAL TARGET CASCADING FOR DISTRIBUTED OPTIMAL POWER FLOW 47
- 3.1 Introduction 47
- 3.2 Contribution 49
- 3.3 OPF in the Context of ATC 49
- 3.4 Accelerated and Robust ALAD 51
- 3.5 Proposed Function to Calculate $\boldsymbol{\alpha}$ 57
- 3.6 Application of AR-ALAD to OPF 68
- 3.7 Application of Proposed Method to Other Coordination Algorithms 74
- 3.8 Conclusion 77

CHAPTER 4. MOMENTUM EXTRAPOLATION PREDICTION-BASED ASYNCHRONOUS DISTRIBUTED OPTIMIZATION FOR POWER SYSTEMS 79
- 4.1 Introduction 79
- 4.2 Contribution 79
- 4.3 Distributed OPF with Synchronous ADMM 80
- 4.4 Proposed Prediction-Correction Based Asynchronous ADMM 84
- 4.5 Numerical Results and Discussions 92
- 4.6 Conclusion 103

CHAPTER 5. LEARNING-BASED ASYNCHRONOUS DOUBLE-LOOP ADMM FOR OPTIMAL POWER FLOW 104
- 5.1 Introduction 104
- 5.2 Contribution 105
- 5.3 Asynchronous ADMM (A-ADMM) 105
- 5.4 Learning-based asynchronous ADMM 106

5.5 Numerical results 111
5.6 Conclusion 120

CHAPTER 6. CONCLUDING REMARKS AND FUTURE WORKS 121
6.1 Concluding remark 121
6.2 Future Work 122

REFERENCES 124

VITA 132

ABSTRACT

Distributed optimization approaches are gaining more attention for solving power systems energy management functions, such as optimal power flow (OPF). Preserving information privacy of autonomous control entities and being more scalable than centralized approaches are two primary reasons for developing distributed algorithms. Moreover, distributed/ decentralized algorithms potentially increase power systems reliability against failures of components or communication links.

In this dissertation, we propose multiple distributed optimization algorithms and convergence performance enhancement techniques to solve the OPF problem. We present a multi-level optimization algorithm, based on analytical target cascading, to formulate and solve a collaborative transmission and distribution OPF problem. This algorithm enables transmission and distribution system operators to solve their OPF subproblems in a parallel, yet collaborative manner, which would result in a more economical and reliable operating point for the whole power system.

Motivated by observing the sensitivity of most augmented Lagrangian (AR)-based distributed algorithms to the choice of initial values of tuning parameters and the level of importance of each term in objective functions, a technique is proposed to balance convergence speed and accuracy of these types of distributed algorithms. A balancing coefficient is determined and incorporated into AR-based distributed OPF to, potentially, accelerate its speed and enhance its robustness to the choice of initial values.

To further enhance the performance of AR-based distributed OPF algorithms, a prediction-correction based asynchronous alternating direction of multipliers (A-ADMM) is proposed. This algorithm does not need synchronization of subproblems at each iteration, which means a computationally cheap subproblem no longer needs to wait for the most updated information of its computationally expensive neighbors. A second loop, which uses anomaly detection and learning techniques, is added to the proposed A-ADMM to reduce the impact of prediction error propagation, especially if subproblems are computationally heterogeneous with a significant level of asynchrony.

CHAPTER 1. INTRODUCTION

1.1 Background

Energy management functions in power systems are undergone profound changes due to emerging new technologies, policies, and rules. The main focus of this study is on optimal power flow (OPF), which is an optimization problem whose objective function is to minimize generation cost (or other objective functions such as loss minimization) subject to system and components operational constraints. OPF is solved daily for various power system analyses, including for generation scheduling in a 5∼15-minute basis.

With the integration of distributed energy resources and the appearance of more autonomous control entities, some challenges come into play for formulating and solving the OPF problems. One of these challenges is to respect the information privacy of autonomous entities that are not willing to share their commercially sensitive information with other parties. Even is these control entities agree to share their data with an independent entity, a large-scale computationally expensive OPF should be formulated and solved.

Distributed approaches have been proposed in the literature to solve OPF distributedly instead of a centralized manner (Molzahn, Dörfler et al. 2017, Kargarian, Mohammadi et al. 2016, Wang, Wang et al. 2017). In distributed optimization, a problem is decomposed into smaller subproblems, and, usually, iterative strategies are applied to coordinate subproblems and find the optimal solution of the whole problem. While various coordination strategies are presented in the literature, the main focus of this study is on augmented Lagrangian-based algorithms. In general, coordination strategies can be categorized into sequential and parallel. In sequential strategies, subproblems are solved sequentially; while an OPF problem is being solved, the rest should stay idle. On the other hand, in the parallel strategies, subproblems can be solved simultaneously. Parallel approaches are more scalable than sequential strategies.

Although various promising distributed optimization algorithms are presented in the literature, there are still several barriers (e.g., lack of scalability, required synchrony between subproblems, sensitivity to the choice of initial conditions, and sensitivity to communication failure and missing information) to real-world applications of distributed optimization not only in power systems but also in many other disciplines. There are critical needs to develop more efficient and reliable algorithms that bear a high convergence speed, scalability, robustness against initial conditions, and resilience against communication failures.

1.2 Motivation and Literature Review

1.2.1 Distributed Optimization for Collaborative TSO+DSO Management

With the integration of distributed energy resources in distribution networks, active distribution grids have emerged with the capability of producing electricity locally for supporting loads. Active distribution grids can provide various services for a transmission system. Transmission and active distribution systems can cooperate to achieve a better grid performance in terms of, for instance, voltage stability and operational costs.

The transmission system is operated by a transmission system operator (TSO), and distribution systems are controlled by a distribution system operator (DSO). Since the transmission and distribution grids are parts of an interconnected system, any decisions made by TSO (DSOs) affect DSOs' (TSO's) operation and decisions. On the other hand, TSO and DSOs are autonomous control entities with their own rules, policies, and objectives. While one entity aims at minimizing its costs, the objective of another entity might be reliability maximization with respect to its local operational constraints. Furthermore, TSO and DSOs might compete with each other to achieve their objectives. Thus, although TSO and DSOs are parts of an interconnected system, they are unwilling to share their commercially sensitive data with other parties. A central scheduling framework, in which TSO and DSOs need to share all their information with a central control authority, may no longer be appropriate for the entire power system operation in the era of active distribution grids (Kar, Hug et al. 2014). Even if TSO and DSOs share their information with a central control authority and allow this entity to perform the decision-making, solving the resulting integrated large-scale optimization problem is challenging. In addition, failures and cyber-attacks could have a devastating impact on the functionality of a centralized control approach.

Joint operation and management of transmission and distribution systems have seen increased interest recently (Ferrante, Constantinescu et al. 2015, Kristov, De Martini et al. 2016). An iterative master-slave algorithm is presented in (Sun, Guo et al. 2015, Li, Guo et al. 2016, Li, Guo et al. 2016) to manage power transmission and distribution systems collaboratively. In (Sun, Guo et al. 2015), a heterogeneous decomposition is presented to solve an AC OPF problem for transmission and distribution systems. This decomposition approach solves first-order KKT conditions in a decentralized manner. DSOs and TSO solve their optimization problems sequentially. In (Kargarian and Fu 2014, Marvasti, Fu et al. 2014, Kargarian, Fu et al. 2016), we have presented a decentralized algorithm for collaborative day-ahead scheduling of TSO and DSOs. The coordination strategy is based on analytical target cascading (ATC), originally developed for multilevel distributed optimization of hierarchical complex engineering systems. References (Kargarian and Fu 2014, Kargarian, Fu et al. 2016) apply augmented Lagrangian block coordinate descent (AL-BCD), and (Marvasti, Fu et al. 2014) utilizes an exponential penalty function (EPF) formulation. Although AL-BCD and EPF formulations effectively coordinate TSO and DSOs, their main drawback is their sequential solution procedure. In other words, at each iteration, TSO (DSOs) needs the updated values of shared variables received from DSOs (TSO) at the same iteration. This sequential scheme degrades the computational efficiency of distributed optimization, as the overall computational time is the summation of subproblems' solution time. This computationally related issue has motivated us to develop a parallel ATC-based collaborative TSO+DSO approach in Chapter II.

1.2.2 Distributed OPF and Scaling

Various distributed optimization algorithms have been reported in the literature (Kar, Hug et al. 2014, Kargarian, Fu et al. 2015, Kargarian, Mohammadi et al. 2016, Manshadi and Khodayar 2016, Malhotra, Binetti et al. 2017, Molzahn, Dörfler et al. 2017, Nguyen, Mohsenian-Rad et al. 2017, Amini, Bahrami et al. 2018, Kargarian, Mehrtash et al. 2018, Zhang, Dehghanpour et al. 2018). We focus on algorithms that have been utilized to solve steady-state problems such as OPF. In most algorithms, interdependencies between subproblems are modeled in the form of either coupling constraints or coupling variables. Primal decomposition algorithms are usually applied to solve problems with coupling variables, and dual decomposition algorithms are used to solve problems with coupling constraints (Palomar and Chiang 2006). References (Kargarian, Mohammadi et al. 2016, Molzahn, Dörfler et al. 2017, Wang, Wang et al. 2017) review distributed algorithms and their application on power systems. Six most popular algorithms, namely, alternating direction method of multipliers (ADMM) (Boyd, Parikh et al. 2011, Erseghe 2014), ATC (Mohammadi, Mehrtash et al. 2018), auxiliary problem principle (APP) (Baldick, Kim et al. 1999), proximal message passing (EE236C), optimality condition decomposition (Hug-Glanzmann and Andersson 2009, Ahmadi-Khatir, Conejo et al. 2013), consensus+innovation (Kar, Hug et al. 2014), and their application to OPF are discussed in (Kargarian, Mohammadi et al. 2016).

In (Wang, Wu et al. 2017), a consensus-based ADMM is proposed to solve OPF with demand response. An asynchronous ADMM is developed in (Guo, Hug et al. 2017) for AC OPF. Reference (Mhanna, Verbič et al. 2018) has proposed two accelerated subgradient methods and an adaptive penalty parameter to speed up the convergence of ADMM for application to OPF. Reference (Liu, Benosman et al. 2015) has presented a distributed OPF that uses a consensus algorithm to estimate optimization variables between neighboring nodes in a given network. ADMM and ATC are applied, respectively, in (Dall'Anese, Zhu et al. 2013) and (Malekpour and Pahwa 2017) to solve OPF in distribution systems. A multi-area OPF algorithm is introduced in (Lu, Liu et al. 2018) based on the distributed interior point method. Regional correction equations are converted into solving a parametric quadratic programming problem during each Newton-Raphson iteration. In (Costley and Grijalva 2012), the feasibility of fully-distributed OPF architectures for the power industry is explored. In (Huang, Chen et al. 2008), a distributed OPF algorithm is proposed based on the decomposition of KKT conditions of the centralized OPF. Distributed large-scale OPF with geographical network decomposition is discussed in (Guo, Hug et al. 2017). Augmented Lagrangian alternating direction inexact Newton method is applied in (Engelmann, Jiang et al. 2019) to solve OPF. Initialization free distributed coordination is proposed in (Cherukuri and Cortes 2016, Yi, Hong et al. 2016) for economic dispatch.

In augmented Lagrangian based distributed algorithms, such as ADMM and ATC, coupling constraints are penalized in the objective function and subgradient methods are used to update Lagrange multipliers (Hestenes 1969, Boyd, Parikh et al. 2011). Penalty parameters, step sizes, and initial values for variables play significant roles in convergence behavior and accuracy of results. If one selects these values improperly, ADMM and ATC may take many iterations to converge or even diverge. Appropriate step sizes and penalty parameters must be selected to ensure that feasible and optimal results will be obtained. Although large step sizes and penalty parameters might increase the speed at the first few iterations, they may hinder convergence and cause losing optimality, oscillating around the optimal point, or divergence at the end. On the other hand, small step sizes and penalty parameters increase the chance of finding the optimal solution; however, they increase the number of iterations (Tosserams, Etman et al. 2006, Bertsekas 2014). Definition of small and large step sizes and penalty parameters is problem-dependent. A step size/penalty parameter might be small for a problem while being too large for another problem.

Furthermore, appropriateness of step sizes and penalty parameters is correlated with the choice of initial values for variables. The combination of these three factors determines the level of importance of each term of the optimization objective function and affects the solution procedure. In distributed optimization, if penalty functions are dominant as compared to other cost functions, optimization pays more attention to reducing inconsistency of coupling constraints rather than the solution optimality. If one selects inconsistency of coupling constraints as stopping criteria, the coupling constraints will be satisfied after a couple of iterations (reach a feasible point) while the result is not optimal. In contrast, if other cost function terms dominate, optimization pays more attention to optimize each term of subproblems' objective function locally and solely while the inconsistency of coupling constraints may vanish slowly. In other words, optimization finds the optimal point for each subproblem while they don't have shared variables.

Hence, the scaling and level of importance of each term in subproblems' objective functions need to be carefully determined. It is highly desirable to design a distributed algorithm that is fast, accurate, and robust (i.e., less sensitivity to the choice of step sizes, penalty parameters, and initial values of variables) and can tolerate improper initialization without divergence. This is the motivation of our studies in Chapter III.

1.2.3 Synchronous and Asynchronous Distributed Optimization for Power Systems

Various distributed optimization algorithms have been presented in the literature for power systems management. These algorithms are based on geographical, contingency scenario, uncertainty scenario, and temporal decomposition strategies (Conejo, Castillo et al. 2006, Malekpour, Pahwa et al. 2016, D. K. Molzahn, F. Dorfler et al. 2017, Amini, Bahrami et al. 2018, Kargarian, Mohammadi et al. 2018). Three main categories of algorithms exist in the context of distributed computing, namely, sequential, synchronous parallel, and asynchronous parallel. Most of the existing algorithms, including the classical ADMM, the classical ATC, and several variants of augmented Lagrangian relaxation are sequential in which optimization subproblems are solved sequentially (Chang, Xu et al. , Hur, Park et al. 2002, Boyd, Parikh et al. 2011, Dall'Anese, Zhu et al. 2013, DorMohammadi and Rais-Rohani 2013, Erseghe 2014, Kargarian, Fu et al. 2015, Kargarian, Mohammadi et al. 2016, Li, Guo et al. , D. K. Molzahn, F. Dorfler et al. 2017, Abraham and Kulkarni 2018, Bahrami and Amini 2018, Kargarian, Mohammadi et al. 2018, Mbuwir, Spiessens et al. 2020). These algorithms are applied to OPF, economic dispatch, and unit commitment problems (Conejo, Castillo et al. 2006, Kar, Hug et al. 2014, Li, Guo et al. 2016, D. K. Molzahn, F. Dorfler et al. 2017, Kargarian, Mohammadi et al. 2018). The main drawback of such approaches is their ample solution time and scalability.

Parallel synchronous strategies, such as parallel variants of ADMM, parallel ATC, consensus+innovations (Kar, Hug et al. 2014), and APP (Kim and Baldick 2000), coordinate subproblems in a parallel manner and reduce the under-utilization of computation resources and decrease the solution time of each iteration as compared to that of sequential approaches (Cohen 1980, Kim and Baldick 2000, Ahmadi-Khatir, Conejo et al. 2013, Dall'Anese, Zhu et al. 2013, Erseghe 2014, Kar, Hug et al. 2014, Kargarian, Fu et al. 2015, Malekpour, Pahwa et al. 2016, Wang, Wu et al. 2016, Baroche, Pinson et al. 2019). Perfect synchrony among subproblems, however, is still a bottleneck. The synchrony requirement means that all subproblems must be solved once at each iteration, and information of shared variables must be exchanged between neighbors before starting the next iteration. While slower subproblems are still running, faster subproblems should stay idle until all subproblems are solved. Slower subproblem dictates the solution time of iterations. This limits the solution time and scalability enhancement, particularly if subproblems are computationally heterogeneous with considerably different computational costs. Power systems operation and management problems, such as geographically decomposed OPF, include considerably heterogenous subproblems.

Some efforts have attempted to overcome shortcomings of synchronous parallel approaches by developing asynchronous parallel distributed optimization (Dwork, Lynch et al. 1988, Zhang and Kwok 2014, Peng, Xu et al. 2016). The concept of asynchronous parallel optimization is used in several disciplines (Chang, Hong et al. 2016, Kumar, Jain et al. 2016), but it is relatively new to the power system community (Aravena and Papavasiliou 2015, Guo, Hug et al. 2017, Ramanan, Yildirim et al. 2019). Asynchronous ADMM is becoming popular for power system optimization, e.g., OPF (Guo, Hug et al. 2017). Although asynchronous ADMM outperforms synchronous ADMM, it still has several limitations that degrade its scalability. The latest values of shared variables received from slower subproblems are used to continuously solve faster subproblems for several iterations instead of keeping fast subproblems in an idle mode. While it reduces unproductive time and under-utilization of computation resources, not much significant progress toward the optimal solution is observed after carrying out each iteration due to using un-updated values of share variables for several iterations. If updated information is not received from slower

OPF subproblems after several iterations, faster OPF subproblems should go to an idle mode until updated information is received from other subproblems. This is to avoid moving toward a suboptimal point or divergence.

Mathematical models and optimization techniques may be developed to solve these problems. Although such techniques can be useful, their development is inherently challenging and complex. Instead, researchers can take advantage of developed approaches in other fields to devise hybrid algorithms for addressing asynchronous ADMM limitations. Machine learning is a research direction gaining tremendous attention in many disciplines, including power systems (Ardakani and Bouffard 2018, Mohri, Rostamizadeh et al. 2018, Baker and Bernstein 2019, Karagiannopoulos, Aristidou et al. 2019). By using historical or simulated datasets, machine learning tools can be applied to project behaviors of different phenomena that cannot easily be analyzed or solved within an acceptable timeframe by pure mathematical models. These features make regression and classification methods promising tools to solve the limitations of asynchronous ADMM. The main challenge is that solving distributed optimization is a sort of online data streaming in which data samples and their trend are obtained iteration by iteration until convergence. This makes applications of regression and classification methods to distributed optimization challenging.

With the given motivation, in Chapter IV, we propose an asynchronous form of ADMM using a prediction-correction technique. In Chapter V, we will enhance the performance of this algorithm by adding an anomaly detection step to reduce the impact of prediction errors.

1.3 Contribution and Organization

In Chapter II, the power system is modeled as a system of systems in which TSO and DSOs are autonomous entities with their local policies and rules. A collaborative two-level OPF is presented with respect to a) interdependencies of transmission and distribution systems and b) the information privacy of TSO and DSO. Interdependencies between TSO and DSOs are modeled by a set of hard constraints. Quadratic penalty terms are utilized to relax the hard constraints in the local objective of each entity. A technique is presented to make non-separable quadratic terms of augmented Lagrangian penalty functions separable. A parallel solution algorithm is presented with two loops: an inner loop to enhance the accuracy of the solution and an outer loop to force the algorithm to converge. At each iteration of the proposed parallel procedure, TSO (DSOs) needs the updated values of the shared variables received from DSOs (TSO) obtained at the previous iteration. Hence, compared with the sequential algorithm, the computational time of each iteration decreases. This can significantly improve the convergence speed as the number of levels increases.

In Chapter III, we focus on ATC and its application for solving the OPF problem in a distributed manner. A technique is proposed to make a tradeoff between accuracy and speed of the ATC-based distributed OPF. A function is designed to determine a balancing coefficient depending on the choice of initial values for the coupling variables and penalty parameters. This coefficient is incorporated into the solution algorithm to automatically make a tradeoff between the convergence speed and solution accuracy by adjusting penalty terms with respect to cost functions. The proposed algorithm avoids a premature convergence if the initial conditions are selected inappropriately, and in particular, if penalty parameters are initialized to large values. The proposed function adjusts penalty parameters to enhance the convergence speed if they are initialized to small values. We name the proposed algorithm an accelerated, robust ATC (AR-ATC) since it enhances the solution speed and makes it more robust against the choice of initial values. We analyze the performance of the proposed algorithm through mathematical justifications and rigorous analytical and numerical justifications. Many cases are simulated to show the performance and effectiveness of the proposed AR-ATC to solve OPF in a distributed manner.

In Chapter IV, we propose a prediction-correction based asynchronous ADMM (A-ADMM) to solve the OPF problem in a distributed manner. The objective is to reduce the unproductive time and the under-utilization of computation resources if OPF subproblems are computationally heterogeneous, which is the case in power systems. The proposed A-ADMM does not need synchronization of subproblems at each iteration, which means a computationally cheap subproblem no longer needs to wait for the most updated information of its computationally expensive neighbors. A momentum-based extrapolation method is proposed to predict missing information required by each subproblem based on the values of shared variables obtained over previous iterations. The extrapolation term projects the values of missing information at next iterations while the momentum term, i.e., the correction step, prevents the predicted values to become far from the possible solution and hence avoids divergence. In addition to predicting missing information, the values of shared variables at next iterations can be estimated before solving subproblems. This enhances the solution speed even if subproblems are homogeneous. Numerical results on various test systems show that the proposed A-ADMM considerably outperforms the classical synchronous parallel ADMM for solving the OPF problem.

In Chapter V, we propose a learning-based double-loop asynchronous ADMM (LA-ADMM) to solve the OPF problem in a distributed manner. This algorithm is an extension of A-ADMM proposed in Chapter IV. In the case of high heterogeneity of subproblems or a considerable number of consecutive iterations with missing information, predictions of missing information may degrade LA-ADMM convergence performance because of prediction error propagation from one iteration to the next iteration. We develop an online steaming-based anomaly detection method using the k-means classifier and add it in the algorithm to measure anomalies in predicted values of shared variables. This method works as a switch controller that activates an inner loop to reduce the propagating impact of shared variables prediction error on Lagrange multipliers, which have a critical impact on distributed optimization convergence behavior.

In Chapter VI, the concluding remarks and future research directions are discussed.

CHAPTER 2. DIAGONAL QUADRATIC APPROXIMATION FOR DECENTRALIZED COLLABORATIVE TSO+DSO OPTIMAL POWER FLOW

2.1 Introduction

Collaborative operation of electricity transmission and distribution networks improves power system economics and reliability. However, this is a challenging problem given that transmission system operators (TSOs) and distribution system operators (DSOs) are autonomous entities that are unwilling to reveal their commercially sensitive information with other parties. This chapter presents a decentralized decision-making algorithm for collaborative TSO+DSO optimal power flow implementation. The proposed algorithm is based on analytical target cascading for multilevel hierarchical optimization in complex engineering systems. A local OPF is formulated for each TSO/DSO taking into consideration interactions between the transmission and distribution systems while respecting autonomy and information privacy of TSO and DSOs. The local OPF of TSO is solved at the upper-level of the hierarchy, and the local OPFs of DSOs are handled at the lower-level. A diagonal quadratic approximation (DQA) and a truncated diagonal quadratic approximation (TDQA) are presented to coordinate local OPF problems in a parallel manner. The proposed collaborative TSO+DSO OPF is evaluated using a 6-bus system and the IEEE 118-bus test system, and promising results are obtained.

Nomenclature

A. Indices, Sets, and Parameters

a, b	Index for border buses in TSO side.
a', b'	Index for border buses in DSOs side.
i, j	Index for subproblem j in level i.
k	Outer loop iteration index.
l	Inner loop iteration index.

f^*	Operating cost determined by the centralized algorithm.
f^{ATC}	Operating cost determined by the decentralized algorithm.
f_{ij}	Operating cost function of subproblem j located in level i.
g_{ij}	Set of inequality constraints of subproblem j located in level i.
h_{ij}	Set of equality constraints of subproblem j located in level i.
N_d	Number of DSO in level two.
P_l^*	Power mismatch in tie-line l determined by the centralized algorithm.
P_l^{ATC}	Power mismatch in tie-line l determined by the decentralized algorithm.
X	Set of variables.
$\pi(.)$	Penalty function.
P_{mis}	Relative power mismatch in a tie-line.
rel	Relative distance of the operating cost.

B. Variables

r_{ij}	Response of subproblem$j$ in level i.
t_{ij}	Targets of subproblem$j$ in level i.
v_a	Voltage magnitude at bus a.
δ_a	Voltage angle at bus a.
$\tilde{v}\angle\tilde{\delta}$	Voltage phasor corresponding to response variables.
$v\angle\delta$	Voltage phasor corresponding to target variables.
α, β	EPF penalty multipliers.
$\delta^{k,l}$	Voltage angle in outer loop k and in inner loop l.
Γ	Step size.
λ, w	AL_BCD’s penalty multipliers.
$\lambda_{ij,\delta}^T, w_{ij,\delta}$	DQA penalty multipliers related to voltage angle of subproblem$j$ in level i.
$\lambda_{2j,v}^T, w_{ij,v}$	DQA penalty multipliers related to voltage magnitude of subproblem j in level i.

τ Tuning parameter.

2.2 Contribution

The contributions of this chapter are summarized as follows:

- The power system is modeled as a system of systems (SoS) in which TSO and DSOs are autonomous entities with their local policies and rules. A collaborative two-level OPF is presented with respect to a) interdependencies of transmission and distribution systems and b) the information privacy of TSO and DSO.

- Interdependencies between TSO and DSOs are modeled by a set of hard constraints. Quadratic penalty terms are utilized to relax the hard constraints in the local objective of each entity. A technique is presented to make non-separable quadratic terms of augmented Lagrangian penalty functions separable.

- A fully parallel solution algorithm is presented, which has two loops: an inner loop to enhance the accuracy of the solution and an outer loop to force the algorithm to converge. At each iteration of the proposed parallel procedure, TSO (DSOs) needs the updated values of the shared variables received from DSOs (TSO) obtained at the previous iteration. Hence, compared with the sequential algorithm, the computational time of each iteration decreases. This can significantly improve the convergence speed as the number of levels increases.

2.3 Decentralized ATC-based OPF Implementation

2.3.1 Dependency of TSO and DSO

Assume that the transmission network is not connected to the active distribution grids. In this case (isolated mode), TSO and DSOs are capable of solving their local OPF problems completely independent from one another. However, this is not the case in reality since distribution grids are interconnected to transmission networks via one or more connection points. Consider the system shown in Fig. 1, which includes one TSO and two DSOs. The system has two levels. TSO is at the first level (upper-level), and DSOs are at the second level (lower-level). Control variables of buses a and a' (i.e., voltage magnitudes and angles) couple TSO and DSO1. Both TSO and DSO1 are interested in controlling these coupling variables to improve their grid performance. Likewise, TSO and DSO2 are coupled via control variables of buses b and b'. The coupling variables, i.e., $\{v_a \angle \delta_a, v_{a'} \angle \delta_{a'}, v_b \angle \delta_b, v_{b'} \angle \delta_{b'}\}$, make decisions of TSO, DSO1, and DSO2 interdependent (note that active and reactive power flows in a tie-line are by-products of the voltage magnitudes and angles of ending terminals of the tie-line). Thus, coordination of the coupling variables is in great interest of TSO and DSOs.

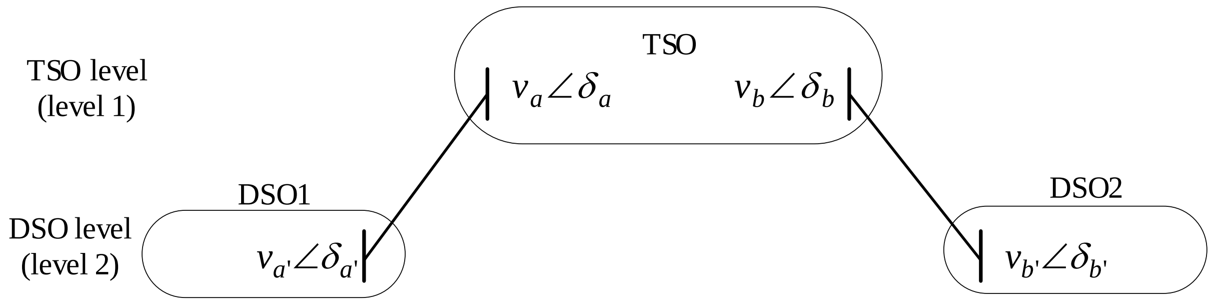


Figure 2.1 Interdependency of TSO and DSOs with coupling variables.

2.3.2 Characterization of Analytical Target Cascading

The general concept of analytical target cascading (ATC) is similar to the auxiliary problem principle (APP) and alternating direction method of multipliers (ADMM) (Cohen 1980, Michelena, Park et al. 2003, Tosserams, Etman et al. 2006, Boyd, Parikh et al. 2011, DorMohammadi and Rais-Rohani 2012). The ATC procedure (which is suitable for multilevel management of complex engineering systems) first decomposes the system into a multilevel hierarchical structure (shown in Fig. 2.2) and recognizes parents and children. At the next step, penalty functions are introduced to model subproblems' interdependencies. Whereas in APP and ADMM, the duality concept is applied, and penalty functions are introduced, and then the system is decomposed into several subproblems. As shown in Fig. 2.1, TSO in the upper-level is hierarchically connected to DSOs in the lower-level. Thus, ATC is a suitable method to solve the collaborative TSO+DSO operation in a decentralized manner. In ATC, subproblems (also called elements or autonomous systems) in the upper-levels are parents of subproblems in the lower-levels. Subproblems in the lower-levels are children of subproblems in the upper-levels. Although a child has only one parent, a parent could have multiple children. This hierarchical interconnection means that there is no loop in the ATC structure. This further implies that subproblems at the same level do not share any connection/information. If we assume the ATC structure as a graph, subproblems and tie-lines are, respectively, nodes and edges of the graph.

By decomposing the system into parents/children, the dimensionality of each subproblem reduces. An iterative solution procedure can be applied to coordinate TSO and DSOs and determine the optimal solution of the SoS-based power system. In ATC, the coupling variables between two connected elements appear in the form of target variables and response copiers. TSO solves its OPF subproblem and propagates the target values down toward its children (i.e., DSOs).

Then, DSOs use the updated target values, solve their local OPF problems, and send the updated values of the response copiers back to TSO. The responses determined by the children define how close they are to the parent's targets (Tosserams, Etman et al. 2006).

To enforce the decentralized optimization problem to converge, a proper coordination strategy is required. Several methods have been proposed in the literature with different options to penalize the coupling variables into the objective functions. These options for selecting penalty terms and coordination strategies make ATC more flexible than ADMM and APP. Augmented Lagrangian block coordinate descent (AL-BCD) and exponential penalty function (EPF) are two popular ATC formulations that use coordination strategies with two loops, inner loop and outer loop. The penalty terms in these two methods are not separable, and thus the solution algorithm is a sequential procedure, as shown in Fig. 2.3. If no direct link exists among the subproblems in each level i, the subproblems (only those is level i) can be solved in parallel.

The ATC structure converges to first-order optimality conditions if the problem is convex (Michelena, Park et al. 2003). Thus, ATC provides the optimal solution for a convex problem. As shown in the literature, ATC shows good performance for non-convex problems, such as AC OPF presented In this chapter (DorMohammadi and Rais-Rohani 2013, Kargarian and Fu 2014, Marvasti, Fu et al. 2014, Kargarian, Fu et al. 2016). In addition, as explained in the following sections, a set of convex penalty functions, such as a quadratic function, are added to the objective function. These convex penalty functions act as local convexifiers for the subproblems and mitigate the non-convexity of the parents' and children's subproblems. The convergence and optimality of the decentralized algorithm, when applied to the studied problem, are demonstrated through several numerical simulations.

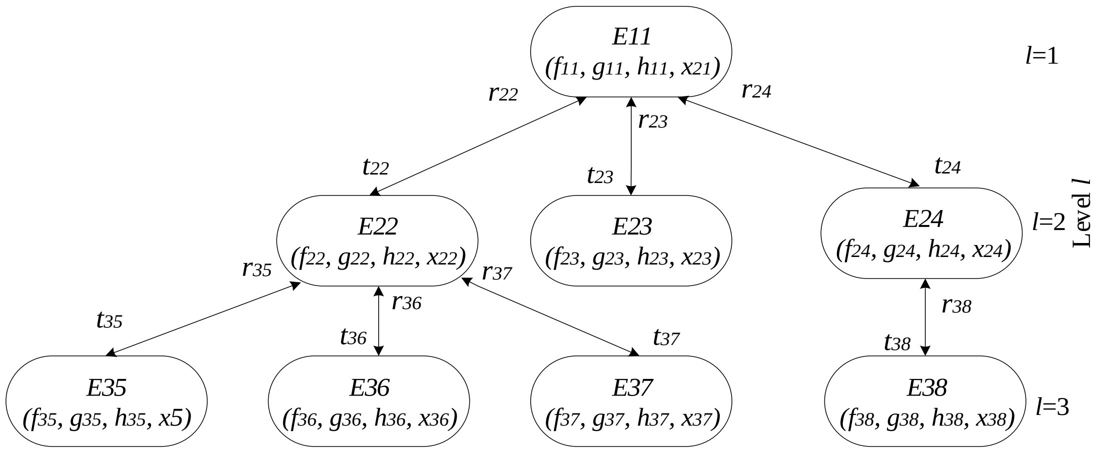


Figure 2.2 Decomposing a system into a multilevel hierarchical structure.

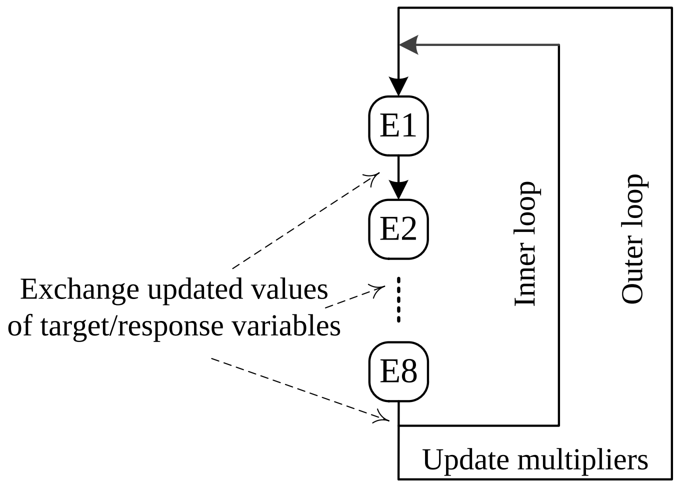


Figure 2.3 Solution procedure of AL-BCD and EPF algorithms.

2.3.3 Hierarchical Two-Level TSO+DSO Operation

In this section, we formulate the collaborative TSO+DSO OPF within the ATC framework. Consider that optimization (2.1) expresses a centralized OPF problem for the combined power transmission and distribution systems.

$$\min_{X} F(X) \tag{2.1}$$

$$s.t.\ \ g(X) \leq 0, \quad h(X) = 0$$

where X denotes all variables of the entire power system, F is the overall objective function, and g and h represent all inequality and equality constraints. The power system has a two-level hierarchical structure (a simplified version of Fig. 2.2). Thus, within the ATC framework, we can rewrite (2.1) as follows:

$$\min_{(x_{ij},t_{2j})} f_{11}\left(x_{11}, t_{2j}\right) + \sum_{j=2}^{N_d+1} f_{2j}\left(x_{2j}, t_{2j}\right) \tag{2.2}$$

$$s.t. \quad g_{ij}\left(x_{ij}, t_{2j}\right) \leq 0$$

$$h_{ij}\left(x_{ij}, t_{2j}\right) = 0$$

where subscript ij indicates subproblem j^{th} in level i, $X = \{x_{ij}, t_{2j}\}$, x_{ij} is local variables of subproblem j in level i, and t_{2j} represents the target variables. Note that in ATC, shared variables that couple TSO to DSOs (i.e., voltage of border buses, as discussed in Section II. A) appear in the form of target variables. Parameter N_d is the number of DSO in level 2, f_{11} is the objective function of TSO, f_{2j} is the objective function of DSO j in level 2, and g_{ij} and h_{ij} are compact representations of inequality and equality constraints of subproblem j in level i. If t is an empty set ($t = \{\}$), TSO and DSOs are isolated and can solve their local OPF subproblems completely separate from each other. However, if t is non-empty ($t \neq \{\}$), which is the case in the power systems, subproblems that share t (an element of t) need to achieve an agreement on its value.

To separate the TSO's and DSOs' OPF subproblems as well as the variables that are governed by each subproblem, response copiers are introduced. The response copiers are duplicates of the target variables. We consider that the target variables are the shared variables (voltage of border buses) that are handled by TSO and the response copiers are the shared variables that are governed by DSOs. We can include the response variables, denoted generically by vector r, in (2.2) by enforcing a set of consistency constraints as:

$$\text{C: } t_{2j} - r_{2j} = 0 \tag{2.3}$$

One consistency constraint is required for each target-response pair. We relax the consistency constraints in the objective function using a penalty term.

$$\min_{(x_{ij},t_{2j},r_{2j})} f_{11}(x_{11}, t_{2j}) + \sum_{j=2}^{N_d+1} f_{2j}(x_{2j}, t_{2j}) + \sum_{j=2}^{N_d+1} \pi(t_{2j} - r_{2j}) \tag{2.4}$$

Now, we can completely separate the local OPF subproblems of TSO and DSOs. Let us represent the target variables by their common notations in power system communities, i.e., $v\angle\delta$. Also, $\tilde{v}\angle\tilde{\delta}$ represents the response variables. The local OPF subproblem of TSO is:

$$\min_{(x_{11},\delta_{2j},v_{2j})} f_{11}(x_{11}, \delta_{2j}, v_{2j}) + \sum_{j=2}^{N_d+1} \pi(\delta_{2j} - \tilde{\delta}_{2j}) + \pi(v_{2j} - \tilde{v}_{2j}) \tag{2.5}$$

$$s.t. \quad g_{11}(x_{11}, \delta_{2j}, v_{2j}) \leq 0$$

$$h_{11}(x_{11}, \delta_{2j}, v_{2j}) = 0$$

And the local OPF subproblem of DSO j is:

$$\min_{(x_{2j},\tilde{\delta}_{2j},\tilde{v}_{2j})} f_{2j}(x_{2j}, \tilde{\delta}_{2j}, \tilde{v}_{2j}) + \pi(\delta_{2j} - \tilde{\delta}_{2j}) + \pi(v_{2j} - \tilde{v}_{2j}) \tag{2.6}$$

$$s.t. \quad g_{2j}(x_{2j}, \tilde{\delta}_{2j}, \tilde{v}_{2j}) \leq 0$$

$$h_{2j}\left(x_{2j}, \tilde{\delta}_{2j}, \tilde{v}_{2j}\right) = 0$$

The OPF subproblem (2.5) ((2.6)) is formulated using the local information of TSO (DSO j) as well as its shared variables with DSOs (TSO). The generation cost function of each subproblem (i.e., f) is quadratic as $f(p) = a + bp + cp^2$. The equality constraints h and the inequality constraints g of TSO and DSOs are as follows:

$$h: \begin{cases} \text{Nodal power balance equations} \\ \text{Voltage angle of the reference bus} = 0 \end{cases}$$

$$g: \begin{cases} \text{Generation capacity limits} \\ \text{Bus voltage limits} \\ \text{Line flow limits} \end{cases}$$

While TSO is allowed to decide about its local and target variables $v\angle\delta$, each DSO j determines its local and corresponding response variables $\tilde{v}\angle\tilde{\delta}$. That is, while $v\angle\delta$ is constant in the DSOs' OPF subproblems, $\tilde{v}\angle\tilde{\delta}$ is constant in the TSO's OPF subproblem. In the ATC framework, TSO sends the target values $v\angle\delta$ down to DSOs, and each DSO sends its response values $\tilde{v}\angle\tilde{\delta}$ back upward TSO.

An iterative procedure needs to be implemented to enforce the difference between $v - \tilde{v}$ and $\delta - \tilde{\delta}$ to zero and find the optimal solution of the entire two-level power system. Depending on the choice of the penalty function $\pi(\cdot)$, the iterative solution procedure could be implemented in a sequential or a parallel fashion. An algorithm in which the TSO and DSOs OPF subproblems are sequentially and iteratively solved is called block coordinate descent. The convergence of the algorithm is guaranteed in (Michelena, Park et al. 2003, Bertsekas 2005). This is independent of the choice of the penalty function since the constraint sets of TSO and DSOs are completely independent.

In (Kargarian and Fu 2014, Marvasti, Fu et al. 2014), we have applied AL-BCD and EPF methods to model the penalty function (π). In AL-BCD, the penalty term is

$$\lambda^T(t-r) + \|w \circ (t-r)\|_2^2 \qquad t = \{v, \delta\}, r = \{\tilde{v}, \tilde{\delta}\} \tag{2.7}$$

And in EPF, the penalty term is:

$$\alpha\left(\mathrm{e}^{(t-r)} - 1\right) + \beta\left(\mathrm{e}^{(r-t)} - 1\right) \quad t = \{v, \delta\}, r = \{\tilde{v}, \tilde{\delta}\} \tag{2.8}$$

where λ, w, α, and β are penalty multipliers, and "$\circ$" denotes the Hadamard product. Setting the penalty factor w to a small value enhances the accuracy of the distributed algorithm, but it increases the number of iterations. A large w potentially reduces the number of iterations, but it may degrade the accuracy of the results. Indeed w should be set to a large enough value (this value is problem-dependent) to balance the cost function f and the penalty function and make a trade-off between the accuracy and speed (Bertsekas 1999). The penalty multipliers λ, α, and β should be initialized close to their optimal values. A user may utilize historical data (e.g., a hot start strategy) or its experience to initialize the penalty multipliers.

Penalty functions (2.7) and (2.8) include non-separable terms. Thus, a sequential solution procedure (hierarchical and level by level, similar to Fig. 2.3) is required to solve the collaborative ATC-based TSO+DSO OPF. That is, in each iteration k, TSO (DSOs) needs to know the response (target) values of DSOs (TSO) in that iteration, i.e., r^k (t^k). Hence, when the TSO's (DSOs') OPF subproblem is being solved, the DSOs' (TSO's) OPF subproblems should stay idle (see (Kargarian and Fu 2014) for more details). This degrades the computational efficiency of the decentralized solution procedure.

2.4 Diagonal Quadratic Approximation Method for Parallel Solution

It is highly desirable to solve the OPF subproblems in a parallel manner, as shown in Fig. 2.4, especially when multiple levels of hierarchy (e.g., TSO, DSO, and microgrid levels) and many subproblems exist. In this chapter, diagonal quadratic approximation (DQA) and truncated diagonal quadratic approximation (TDQA) are presented to parallelize the solution procedure of the collaborative ATC-based TSO+DSO OPF. In these two algorithms, a subproblem with the longest solution time determines the algorithms' solution time in each iteration. In contrast, in a sequential algorithm, such as AL-BCD, the summation of TSO's solution time and the longest solution time of DSOs determines the overall solution time of the algorithm.

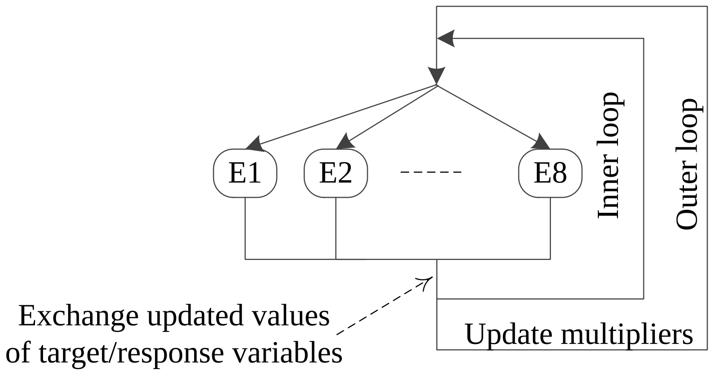


Figure 2.4 Solution of OPF subproblems with parallel ATC coordination strategy.

2.4.1 Diagonal Quadratic Approximation (DQA)

The objective functions f, i.e., the generation cost functions, in TSO and DSO subproblems are convex functions. In addition, the local equality and inequality constraints of each subproblem are fully separable. In ATC, we have the flexibility to select the penalty term $\pi(\cdot)$ to relax the consistency constrains in the local objective functions. We have followed the concept of augmented Lagrangian and selected a combination of linear and quadratic penalty functions as in

(2.7). The augmented term, i.e., the quadratic term, improves the convergence performance compared with the ordinary Lagrangian function. In addition, this penalty term acts as a local convexifier and enhances the behavior of subproblems. However, this quadratic term of the augmented Lagrangian penalty function is not separable. We apply the diagonal quadratic approximation (DQA) method to make the augmented Lagrangian terms separable(Ruszczyński 1995, Li, Lu et al. 2008). Consider the penalty function corresponding to the voltage angle. We expand its quadratic term as:

$$\left\|\delta_{ij}-\tilde{\delta}_{ij}\right\|_2^2=\left\|\delta_{ij}\circ\delta_{ij}+\tilde{\delta}_{ij}\circ\tilde{\delta}_{ij}-2(\delta_{ij}\circ\tilde{\delta}_{ij})\right\| \tag{2.9}$$

We use the first-order Taylor expansion for multiple variable scalar functions to linearize the cross term $\delta_{ij}\circ\tilde{\delta}_{ij}$ at the point $(\delta_{ij}^{k-1}, \tilde{\delta}_{ij}^{k-1})$ [34, 35].

$$\delta_{ij}\circ\tilde{\delta}_{ij}\cong\tilde{\delta}_{ij}^{k-1}\circ\delta_{ij}+\delta_{ij}^{k-1}\circ\tilde{\delta}_{ij}-\delta_{ij}^{k-1}\circ\tilde{\delta}_{ij}^{k-1} \tag{2.10}$$

where δ_{ij}^{k-1} and $\tilde{\delta}_{ij}^{k-1}$ are respectively targets and responses determined in the previous iteration $k-1$ and are constant in the current iteration k. Thus, we can approximate the quadratic penalty term (2.9) as:

$$\left\|\delta_{ij}-\tilde{\delta}_{ij}\right\|_2^2=\left\|\delta_{ij}^{k-1}-\tilde{\delta}_{ij}\right\|_2^2+\left\|\delta_{ij}-\tilde{\delta}_{ij}^{k-1}\right\|_2^2+\mathrm{C} \tag{2.11}$$

where C is a constant. The same Taylor expansion is implemented on the quadratic penalty term corresponding to voltage magnitudes. Now, the OPF problem of the entire two-level power system can reformulate as:

$$\min_{(x_{ij},\delta,v,\tilde{\delta},\tilde{v})}\sum_{i=1}^{N_l}\sum_{j}f_{ij}\left(x_{ij},\delta_{(i+1)j},v_{(i+1)j}\right)$$

$$+\sum_{i\in N_{l,i\neq 1}}\sum_{j}\left(\lambda_{ij,\delta}^{T}\left(\delta_{ij}-\tilde{\delta}_{ij}\right)+\left\|w_{ij,\delta}\circ\left(\delta_{ij}^{k-1}-\tilde{\delta}_{ij}\right)\right\|_2^2+\left\|w_{ij,\delta}\circ\left(\delta_{ij}-\tilde{\delta}_{ij}^{k-1}\right)\right\|_2^2\right)$$

$$+\sum_{i\in N_{l,i\neq 1}}\sum_{j}\left(\lambda_{ij,v}^{T}\left(v_{ij}-\tilde{v}_{ij}\right)+\left\|w_{ij,v}\circ\left(v_{ij}^{k-1}-\tilde{v}_{ij}\right)\right\|_2^2+\left\|w_{ij,v}\circ\left(v_{ij}-\tilde{v}_{ij}^{k-1}\right)\right\|_2^2\right) \tag{2.12}$$

where N_l denotes the number of levels, which is two. This optimization problem is subject to all-in-once constraints, i.e., all constraints of TSO and DSOs. We now decompose (2.12). The local OPF subproblem of TSO in iteration k of the ATC procedure is

$$\min_{(x_{11},\delta,v)} f_{11}\left(x_{11},\delta_{22},v_{22},\delta_{23},v_{23},\dots,\delta_{2(N_d+1)},v_{2(N_d+1)}\right) \tag{2.13}$$

$$+\sum_{j=2}^{N_d+1}\lambda_{2j,\delta}^{T}\delta_{2j}+\left\|w_{2j,\delta}\circ\left(\delta_{2j}-\tilde{\delta}_{2j}^{k-1}\right)\right\|_2^2+\sum_{j=2}^{N_d+1}\lambda_{2j,v}^{T}v_{2j}+\left\|w_{2j,v}\circ\left(v_{2j}-\tilde{v}_{2j}^{k-1}\right)\right\|_2^2$$

subject to the local constraints of TSO (e.g., nodal power balance, line flow limits, etc.). The penalty term depends on the target variables $v_{2j}\angle\delta_{2j}$ while using the response values $\tilde{v}_{2j}^{k-1}\angle\tilde{\delta}_{2j}^{k-1}$ determined by DSOs in the previous iteration $k-1$. TSO solves its local OPF problem and finds the target values. Likewise, the local OPF subproblem of each DSO j is reformulated as

$$\min_{(x_{2j},\tilde{\delta}_{2j},\tilde{v}_{2j})} f_{2j}\left(x_{2j},\tilde{\delta}_{2j},\tilde{v}_{2j}\right)+\lambda_{2j,\delta}^{T}\left(-\tilde{\delta}_{2j}\right)+\lambda_{2j,v}^{T}\left(-\tilde{v}_{2j}\right)$$

$$+\left\|w_{2j,\delta}\circ\left(\delta_{2j}^{k-1}-\tilde{\delta}_{2j}\right)\right\|_2^2+\left\|w_{2j,v}\circ\left(v_{2j}^{k-1}-\tilde{v}_{2j}\right)\right\|_2^2 \tag{2.14}$$

subject to the local constraints of DSO j. The penalty term depends on the response variables $\tilde{v}_{2j}\angle\tilde{\delta}_{2j}$ while using the target values $v_{2j}^{k-1}\angle\delta_{2j}^{k-1}$ determined by TSO in the previous iteration $k-1$. Formulations (2.13) and (2.14) allow a parallel solution of the TSO's and DSOs' OPF subproblems since each subproblem needs the target/response values determined by other subproblems in iteration $k-1$. The DQA coordination strategy is proven to converge and its convergence rate is discussed in (Ruszczyński 1995) and (Li, Lu et al. 2008).

2.4.1.1 Parallel Solution Procedure of DQA

Figure 5 illustrates the solution procedure of DQA to coordinate the OPF subproblems of TSO and DSOs. Although the problem's structure has a hierarchical two-level form, the presented coordination strategy is a parallel procedure that allows a simultaneous solution of TSO's and DSOs' subproblems. The DQA solution strategy includes two loops, inner loop and outer loop. The inner loop updates the target and response values while the penalty multipliers are fixed. This improves the linearization. The inner loop stops when the difference between each target (response) determined in two consecutive iterations is less than a threshold. Indeed, the inner loop seeks to find the best targets and responses for a given set of multipliers. If the targets and responses are determined more precisely, the penalty multipliers are updated more accurately in the outer loop. If the penalty multipliers are updated more accurately, the algorithm takes fewer iterations to update the multipliers. Thus, although the inner loop increases the computational cost (corresponding to the inner loop iterations), it might reduce the number of outer loop iterations in which the multipliers are updated (i.e., the method of multipliers). The steps are discussed in details as follows:

Step1: Set the initial value of local variables x of each subproblem, target values $\{\delta, v\}$, response copiers $\{\tilde{\delta}, \tilde{v}\}$, penalty multipliers λ and w, and parameters Γ and τ. Set the outer loop iteration index $k=1$ and the inner loop iteration index $l=0$.

Step2: Increase the inner loop iteration by one, i.e., $l=l+1$. Solve TSO's and DSOs' local OPF subproblems in parallel using targets and responses that are determined in the previous inner loop iteration ($l-1$), i.e., $\delta^{k-1,l-1}$ and $\tilde{\delta}^{k-1,l-1}$. Note that in the first iteration, the subproblems are solved using the initial values.

Step3: Check the following inner loop convergence criterion

$$\max\left(\|\delta^{k,l}-\delta^{k,l-1}\|,\left\|\tilde{\delta}^{k,l}-\tilde{\delta}^{k,l-1}\right\|,\|v^{k,l}-v^{k,l-1}\|,\|\tilde{v}^{k,l}-\tilde{v}^{k,l-1}\|\right)\leq\epsilon_{inner} \qquad (2.15)$$

where ϵ_{inner} is the stopping threshold of the inner loop. If differences between target (and response) values determined in the current and previous iterations are less than the acceptable threshold, then we should stop the inner loop, set $X^k=X^{k,l}$ (where $X=[x,\delta,v,\tilde{\delta},\tilde{v}]$), and go to Step 4; otherwise:

$$X^{k,l}=X^{k,l-1}+\Gamma(X^{k,l}-X^{k,l-1}) \qquad (2.16)$$

where Γ is the step size, which determines a value among the current solution and the previous one (i.e., if $\Gamma\cong 0$, the algorithm uses with the previous solution, and if $\Gamma\cong 1$ the algorithm uses with the current solution), and then go to Step 2. Note that we update the initial values of all local and shared variables.

Step4: If $\max\{\left\|\delta^k-\tilde{\delta}^k\right\|,\|v^k-\tilde{v}^k\|\}\leq\epsilon_{outer}$ (i.e., if the difference between each target-response pair is less than the criterion), where ϵ_{outer} is the outer loop stopping threshold, TSO+DSO OPF has converged and the optimal values are $\bar{X}^*=\bar{X}^k$, otherwise increase the outer loop iteration index by one (i.e., $k=k+1$) and set the inner loop index to zero (i.e., $l=0$) and update the penalty multipliers as follows:

$$\lambda_\delta^k=\lambda_\delta^{k-1}+w_\delta^{k-1}\circ(\delta^{k-1}-\tilde{\delta}^{k-1}) \qquad (2.17)$$

$$\lambda_v^k=\lambda_v^{k-1}+w_v^{k-1}\circ(v^{k-1}-\tilde{v}^{k-1}) \qquad (2.18)$$

$$w_{\delta}^{k} = \tau_{\delta} w_{\delta}^{k-1} \quad (2.19)$$

$$w_{v}^{k} = \tau_{v} w_{v}^{k-1} \quad (2.20)$$

and then go to Step 2 (note that multipliers will be updated for every outer loop iteration). Parameter τ should be equal or large than one, i.e., $\tau \geq 1$ (DorMohammadi and Rais-Rohani 2012). Depending on the optimization problem characteristics, a wide range of τ can be selected to reduce the computational cost and/or enhance the solution accuracy. Based on our experience, setting τ close to one provides an accurate solution while the computational burden is reasonable.

$\Gamma \in (0,1)$ is the step size that affects the linearization accuracy of the second-order penalty term. A small Γ leads to more accurate results, but it decreases the convergence speed. Parameters ϵ_{outer} and ϵ_{inner} should be significantly smaller than Γ; otherwise, the obtained solution might not be optimal.

The ATC method is proven to converge to an accumulation point (i.e., the shared variables converge to a unique point) that satisfies the first-order optimality conditions of the local optimization problems. This accumulation point also satisfies the first-order optimality conditions of the original problem (Michelena, Park et al. 2003). In addition, (Ruszczyński 1995) provides the convergence proof and convergence rate of the diagonal quadratic approximation method when applied to separate subproblems of the augmented Lagrangian approach. It is worth to mention that the quadratic penalty terms act as local convexifiers and improves the performance of ATC when applied to non-convex problems.

2.4.2 Truncated Diagonal Quadratic Approximation (TDQA)

The collaborative ATC-based TSO+DSO optimal power flow converges when the optimal values of Lagrange multipliers are found. As explained in the DQA solution procedure, the multipliers are not updated in the inner loop. The inner loop helps to improve linearization. Each iteration of the outer loop, in which the multipliers are updated, might take many inner loops. Thus, the inner loop increases the computational effort. Obtaining a high accurate solution of the OPF subproblems in the inner loop is not necessary as the inner loop solution might not be the overall optimal solution. If we only solve the outer loop and update the Lagrange multipliers after every iteration, the multipliers move quickly toward the optimal values. Thus, we omit the inner loop and only consider the outer loop and update the Lagrange multipliers after every iteration. This single-loop coordination strategy is called truncated diagonal quadratic approximation (TDQA) (Ruszczyński 1995, Li, Lu et al. 2008). Note that one can consider the inner loop, but limiting its iterations with any extra criterion (in addition to DQA criterion) rather than allowing it to be able to go to infinity. This procedure is also TDQA as the inner loop is truncated compared with DQA.

The solution procedure of the TSO+DSO operation with TDQA is summarized in the following pseudocode. It has a similar structure as DQA except that DQA has an inner loop to decrease the gap between targets and responses and then update multipliers while TDQA performs this only by the outer loop. In the case study section, we show that TDQA provides promising results for the collaborative TSO+DSO operation. Although the inner loop enhances the targets and responses accuracy over the course of iterations, it is not necessary for convergence. Indeed, updating penalty multipliers in the outer loop (which is based on the method of multipliers) guarantees the convergence of the ATC-based algorithm to the first optimality conditions. That is, TDQA might slightly increase error; however, its convergence is still ensured.

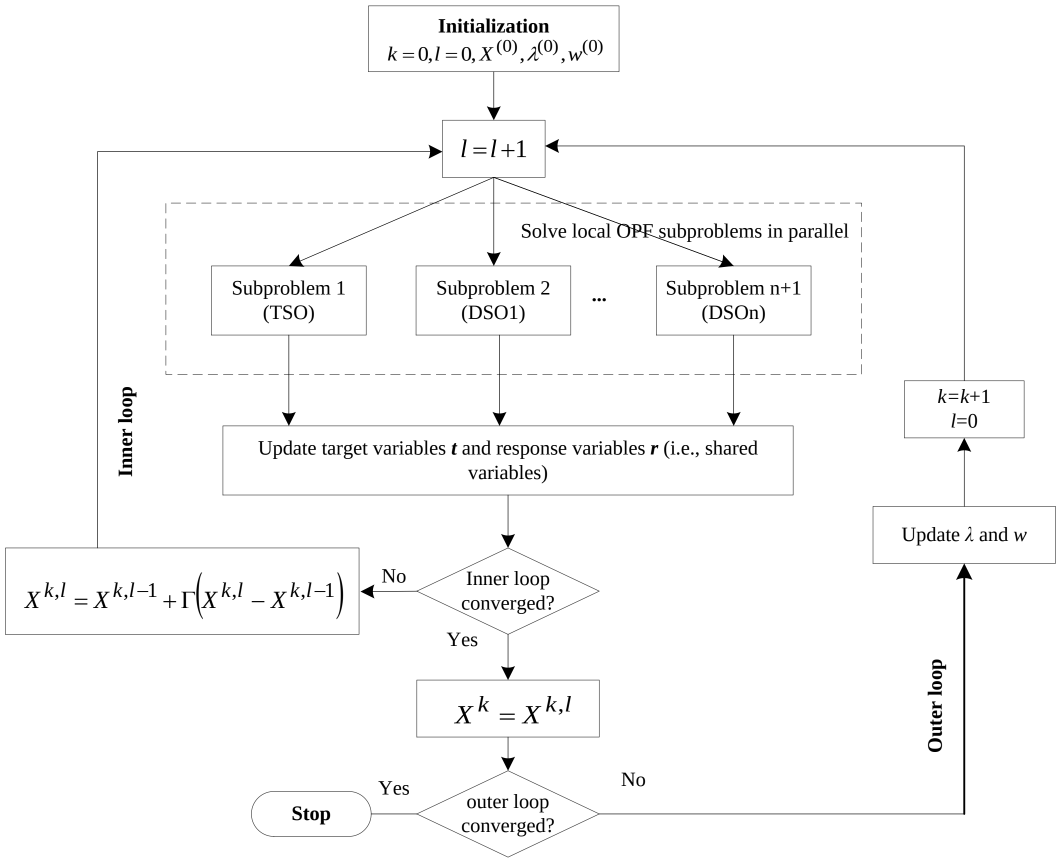

Initialization
$k=0, l=0, X^{(0)}, \lambda^{(0)}, w^{(0)}$
$l=l+1$
Solve local OPF subproblems in parallel
Subproblem 1 (TSO)
Subproblem 2 (DSO1)
...
Subproblem n+1 (DSOn)
Update target variables ***t*** and response variables ***r*** (i.e., shared variables)
Inner loop
$X^{k,l} = X^{k,l-1} + \Gamma\left(X^{k,l} - X^{k,l-1}\right)$
No
Inner loop converged?
Yes
$X^k = X^{k,l}$
Stop
Yes
outer loop converged?
No
Outer loop
Update λ and w
k=k+1
l=0

Figure 2.5 DQA solution algorithm.

2.5 Numerical Results

We implement the DQA and TDQA coordination strategies on a 6-bus and the IEEE 118-bus test systems. The numerical simulations show the efficiency and convergence of the ATC-based collaborative TSO+DSO algorithm, even for the non-convex OPF problems. All computations are carried out using the quadratic programming solver of Matlab on a 2.6GHz personal computer with 16GB of RAM.

```
Solution Algorithm of TDQA
1: initialize $X = [x, \delta, v, \tilde{\delta}, \tilde{v}]$, $\lambda$, $w$, and $\tau$
2: while max $(\|\boldsymbol{\delta}^k - \tilde{\boldsymbol{\delta}}^k\|, \|\boldsymbol{v}^k - \tilde{\boldsymbol{v}}^k\|) \leq \epsilon_{outer}$ , $k = k + 1$ do
3:   Solve (3.13) and (3.14) in a parallel manner and determine $X^k$
4:   Update $X$:  $X^k = X^{k-1} + \Gamma(X^k - X^{k-1})$
5:   Update multiplier $\lambda^k$ and $w^k$ using (2.17)-(2.20)
6: end while
```

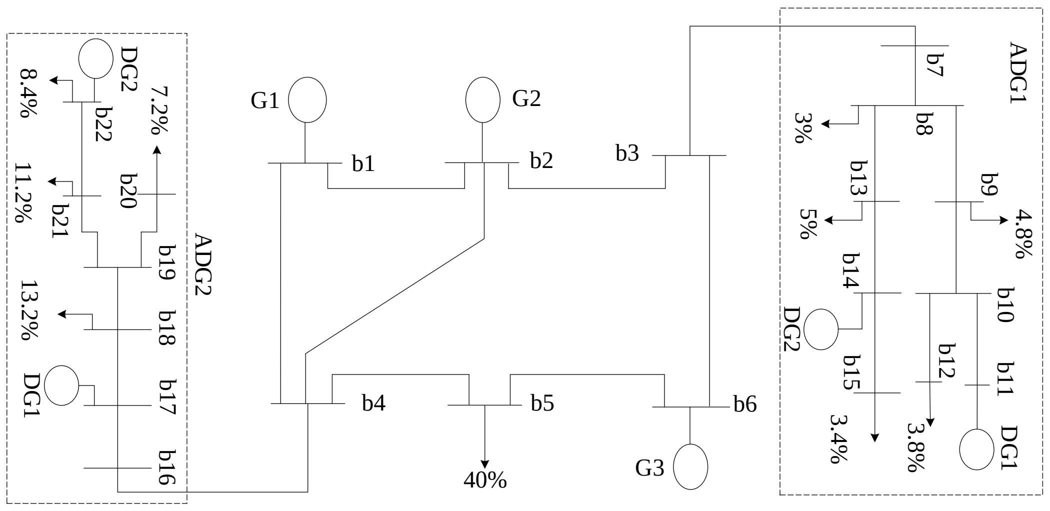


Figure 2.6 Six-bus test system.

2.5.1 6-Bus Test System

The system topology is shown in Fig. 2.6. The transmission system includes six buses, seven transmission lines, and three generators. Two active distribution grids are connected to the transmission system. Active distribution grid one consists of nine buses, five loads, and two DGs. Active distribution grid two includes seven buses, four loads, and two DGs. The total load is 256MW. The resistance of each distribution line is 40% of the line's reactance. The reactive power limit of each generating unit is 60% of its active power limit. The reactive power consumption of each load is assumed to be 30% of its active power demand. The rest of the information is given in (Kargarian and Fu 2014). We study and analyze the following four cases:

Case 1: Centralized OPF implementation considering a single operator for transmission and distribution networks

Case 2: The proposed decentralized decision-making with respect to autonomy of TSO and DSOs

Case 3: Sensitivity of the proposed collaborative TSO+DSO OPF to variation of input parameters

Cases 4: Comparison between TDQA, APP, ADMM, and ALAD

Case 1: We ignore autonomy and information privacy of TSO and DSOs and consider that the transmission and distribution networks are operated by the same operator using the centralized OPF method. Although this is not a realistic case since TSO and DSOs are autonomous, the centralized method provides the reference results that can be used to evaluate the performance of decentralized decision-making. Since the ratio of lines reactance to resistance is large in the transmission system, DC OPF is a reasonable approximation of AC OPF. Thus, DC OPF is used for TSO, whereas AC OPF is used for DSOs[1]. To model the reactive mismatch at border buses between TSO and DSOs, we consider voltage magnitude at transmission terminals equal to one, while the voltage magnitude at the distribution terminals can vary between 0.95 and 1.05 (note

[1] A user may deploy a linearized model of AC OPF. Since OPF is solved continually, in each time interval, the user can deploy results of the previous interval (i.e., a hot start) to linearize OPF around the operating point. Also, the user may convexify AC OPF using techniques such as semidefinite programing or second order cone programing Lavaei, J. and S. H. Low (2012). "Zero duality gap in optimal power flow problem." IEEE Transactions on Power Systems **27**(1): 92-107, Kocuk, B., S. S. Dey and X. A. Sun (2016). "Strong SOCP relaxations for the optimal power flow problem." Operations Research **64**(6): 1177-1196.

that since the voltage at the transmission terminal is close to one, the reactive mismatch in the tie-line is not significant compare with the case that AC OPF is considered for TSO). The total operating cost of the system is \$3,396. The operating costs of TSO, DSO1, and DSO2 are respectively \$2,375, \$351.7, and \$669.3. The voltage phasors of buses that connect TSO to DSO1 are $1\angle-0.0293$ and $0.998\angle-0.0385$, and the voltage phasors of buses that connect TSO to DSO2 are $1\angle-0.0459$and $1.0018\angle-0.0797$.

Case 2: In this case, autonomy and information privacy of the three systems (i.e., TSO, DSO1, and DSO2) are taken into account, and each system is operated by an independent operator. We have considered the same operation horizon for TSO and DSOs, e.g., 5-minute intervals. The OPF problems are run for one snapshot, and it is assumed that the entities start solving their subproblems simultaneously. Note that even if the operation intervals of TSO and DSOs are not the same, to allow a collaborative operation, we can consider the operation horizon equal to the longest interval. TSO is the parent, and its children are DSOs 1 and 2. A tie-line connects the border bus b3 of the transmission system to the border bus b7 of ADG1, and another tie-line links the border bus b4 of the transmission system to the border bus b16 of ADG2. Thus, voltage of buses b3 and b7 are shared variables between TSO and DSO1, and voltage of buses b4 and b16 are shared variables between TSO and DSO2. TSO includes four target variables, and each DSO has two response variables. We analyze cold start and hot start conditions.

Cold start: The initial values for targets/responses are set to zero, and the initial values of penalty multipliers/parameters are λ^0=1000, w_δ^0=1500, $w_{v7,ADG1}^0=30$, $w_{v16,ADG2}^0=10$, Γ=0. 9, and $\tau=1$. For DQA, the inner and outer loops' convergence thresholds are $\epsilon_{inner}=0.004$ and $\epsilon_{outer}=1.4\times10^{-4}$, respectively. Note that TDQA has only the outer loop with the convergence threshold of $\epsilon_{outer}=1.4\times10^{-4}$. The DQA coordination strategy converges after 33 outer loop iterations, and the total number of executed inner loops is 40, whereas TDQA converges after 34 iterations. Figure 2.7(a) shows the updating process of target-response pair corresponding to the voltage angles of bus 3 of TSO and bus 1 of DSO1 over the course of iterations (note that for DQA, we show the updating process over the course of overall iterations, i.e., inner and outer loops). The difference between each pair of target-response becomes smaller, and it is less than the convergence threshold in iteration 40 (34) of DQA (TDQA). Although in several iterations (e.g., iteration 18 in TDQA) a target and its corresponding response value might differ less than the stopping threshold, the algorithms stop when the differences between every pair of target-response become less than the stopping threshold. Although TDQA takes more (outer loop) iterations than DQA, it does not need the inner loop. Overall, DQA needs 40 iterations (sum of inner and outer loops iterations), six iterations more than TDQA. Although DQA needs more iterations than TDQA, it finds the solution more precisely, especially when lower ϵ is chosen. Figure 2.7(b) shows the average difference between the target-response values over the course of iterations. Note that for DQA, the updating process is shown over the course of overall iterations. The error decays faster in TDQA than DQA because while DQA tries to enhance the solution by repeating the inner loop with the fixed multipliers, TDQA seeks to improve the solution by updating the multipliers. This reduces the overall number of function evaluations and the computational time of TDQA (DQA and TDQA take respectively 3.97 and 3.62 seconds to converge); however, it might slightly increase the overall error. Table 2.1 shows the generation dispatch for the three systems. Since the stopping threshold is not zero, the dispatch results obtained from the centralized and decentralized algorithms are slightly different. However, as shown in Table 2.2, the operating costs determined by the decentralized and decentralized algorithms are similar. The operating costs of TSO, DSO1, and DSO2 determined by DQA are \$2,379.8, \$ 350.8, and \$669.3, and they are \$2,378.3, \$351.7, and \$ 669.3 when using TDQA. The total power system operating costs determined by DQA and TDQA are \$3,399.9 and \$3,399.3 that are almost the same as the cost obtained by the centralized OPF (i.e., \$3,396). Note that the sensitivity of the solver and solution to changes in power generated by TSO's and DSOs' generators might be different since the cost functions and geographical locations of the units are different. Thus, although power dispatch of DSOs and TSO units are slightly different from the centralized results, both algorithms yield almost the same operating cost.

To evaluate the performance of the proposed ATC-based TSO+DSO OPF in more detail, we formulate two convergence indices. The first index is the Euclidean norm of mismatch between the power flow in tie-lines connecting transmission system to active distribution grids (P_l^{ATC}) and the optimal value obtained by the centralized OPF (P_l^*):

$$P_{mis} = \left\| \frac{P_l^* - P_l^{ATC}}{P_l^*} \right\| \tag{2.21}$$

The second index is the relative distance of the total cost determined by ATC (f^{ATC}) from the optimal value determined by the centralized OPF (f^*):

$$rel = \frac{|f^* - f^{ATC}|}{f^*} \tag{2.22}$$

The values of the two convergence measures are zero at the optimal point. Hence, the closer these convergence measures get to zero, the better solution is obtained. Figure 2.8 shows P_{mis} and rel values over the course of iterations. The values of the convergence measures decrease when more iterations are carried out. They are small enough upon the algorithm convergence.

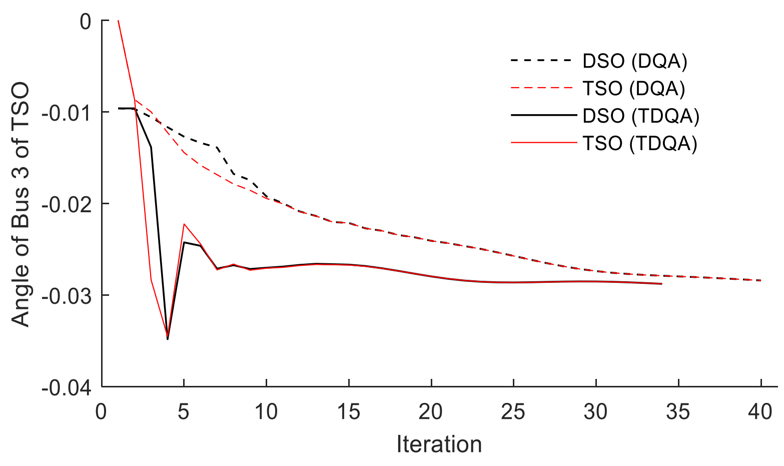
Table 2.1 Power Output of Generating Units

Algorithm	TSO			DSO1		DSO2	
	G1	G2	G3	DG1	DG2	DG1	DG2
Centralize	129.41	34.03	25	15	18	25	13.17
Dec. (DQA)	126.65	37.07	25	14.91	18	25	13.17
Dec. (TDQA)	127.93	35.72	25	15	18	25	13.17

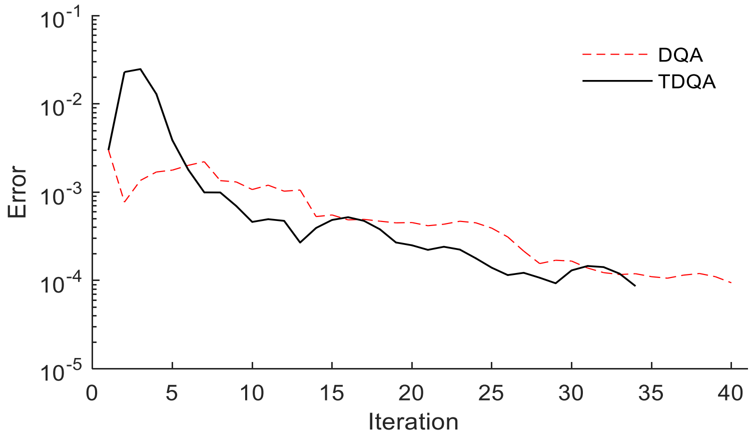
Table 2.2 Operating Cost Obtained by Different Algorithms

Algorithm	TSO	DSO1	DSO2	Total cost
Centralize	\$2,375	\$351.7	\$669.3	\$3,396
Dec. (DQA)	\$2,379.8	\$350.8	\$669.3	\$3,399.9
Dec. (TDQA)	\$2,378.3	\$351.7	\$ 669.3	\$3,399.3

Hot start: In practice, we usually have good initial values for the variables and penalty multipliers. For example, when we solve the OPF problem for interval ω, we have the optimal results of interval $\omega - 1$. We know that, in most cases, the OPF input parameters, e.g., power demand, vary slightly from interval $\omega - 1$ to interval ω. Thus, the solution of interval $\omega - 1$ can be utilized to initialize the problem in interval ω. This strategy is called a hot start. We can solve the problem faster and more precise by selecting appropriate initial values. We assume that the load changes 5% between intervals $\omega - 1$ and ω and use the solution obtained in interval $\omega - 1$ to initialized the variables and penalty multipliers in interval ω. Figure 2.8 shows the convergence measure for the hot start and cold start cases.



(a)



(b)

Figure 2.7 Target and response values corresponding to b3 of TSO and b7 of DSO1 and b) the average difference between targets and responses.

While DQA and TDQA take 40 and 34 iterations using the cold start, they take 16 and 17 iterations using the hot start, respectively. The rel index of cold start is 9×10^{-4}, whereas it is 1.06×10^{-4} using the hot start. Note that since we have good initial conditions (i.e., good guesses for target/response and penalty multipliers), DQA and TDQA behaviors are similar.

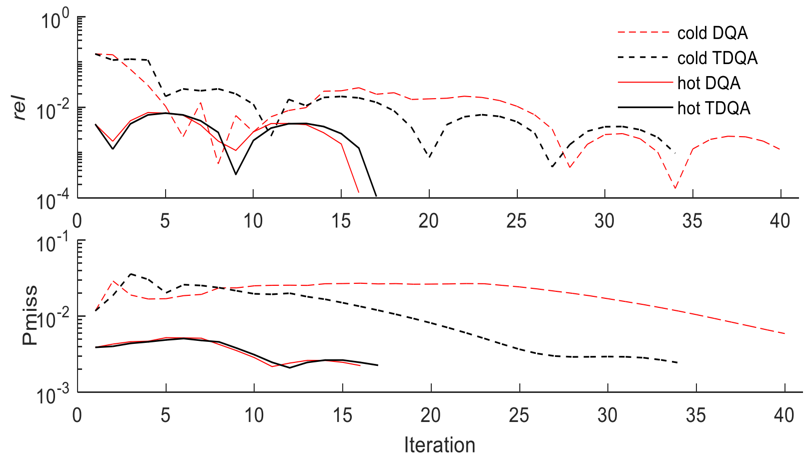


Figure 2.8 Convergence property of DQA- and TDQA-based collaborative OPF.

A Full AC OPF: We have tested the proposed algorithm on a full AC OPF (i.e., AC OPF for TSO and AC OPF for DSOs). The initial values for targets/responses and multipliers are the same as the cold start strategy. We stop the algorithm after 60 iterations. The rel indices obtained from both approaches are shown in Fig. 2.9. The decentralized algorithm provides acceptable rel indices. Note that the combination of DC OPF and AC OPF provides good results in the first few iterations; however, the rel index gradually goes down for the case of the full AC OPF (a user may run the algorithm more than 60 iterations to get a smaller rel index). The DC OPF approximation for TSO slightly increases the error (because of the linearization of AC OPF) but enhances the decentralized algorithm performance. The user may prefer to use such an approximation to get faster results from the decentralized algorithm. Note that using DC OPF for TSO and AC OPF for DSOs is aligned with the power industry.

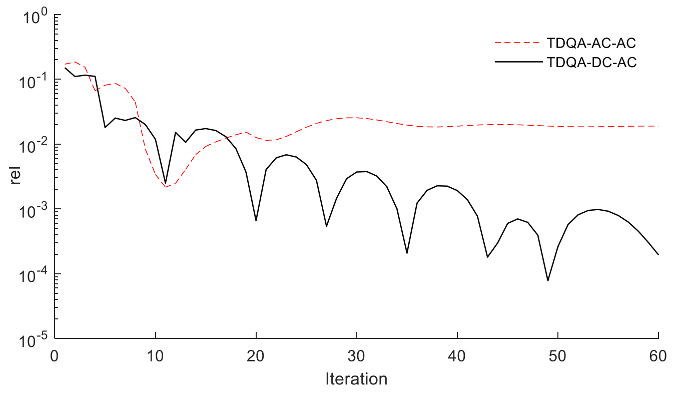
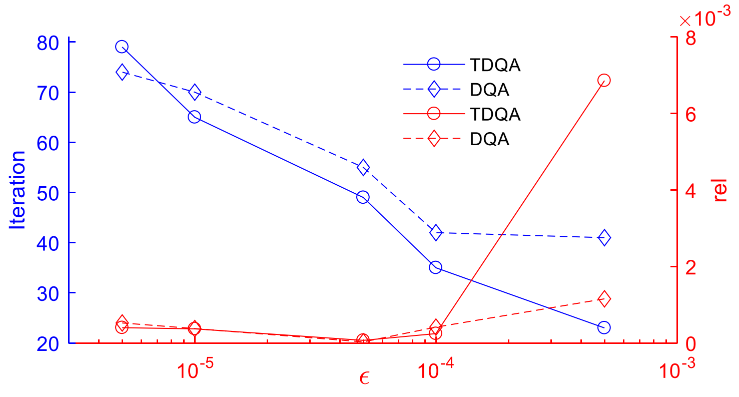


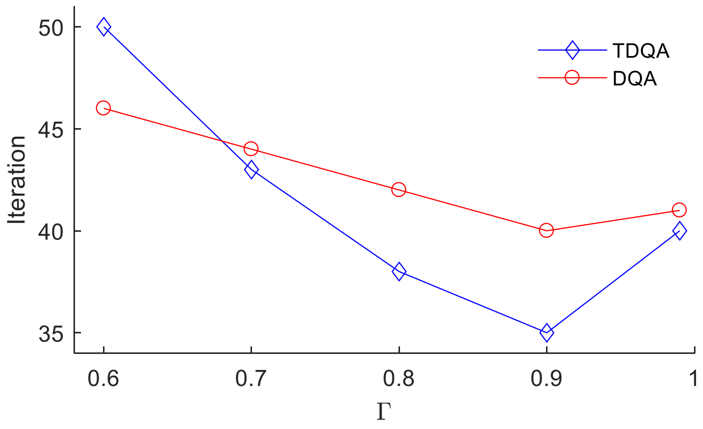
Figure 2.9 The rel index for DC OPF+AC OPF and a full AC OPF.

Case 3: To evaluate the convergence behavior of the proposed collaborative TSO+DSO OPF with respect to variations of DQA/TDQA parameters, we perform multiple sensitivity analyses. This provides a user with insights on how to initialize the algorithms' parameters. We initialize λ^0 =1000, w_δ^0 =1500, $w_{v7,ADG1}^0 = 30$, $w_{v16,ADG2}^0 = 10$ and Γ=0.9. We select various stopping thresholds as 5×10^{-4}, 1×10^{-4}, 5×10^{-5}, 1×10^{-5} and 5×10^{-6} and demonstrate the relative error and number of iterations in Fig. 2.10(a). By decreasing the stopping criteria, the relative error decreases generally, but the number of iterations increases. One can select a small enough stopping threshold to make a trade-off between the stepped and error. When the stopping threshold is large, DQA's error is slightly smaller than that for TDQA. However, for the small thresholds, the relative errors of TDQA and DQA are almost the same. Overall, comparing the error and number of iterations shows that TDQA has better performance than DQA.

We set $\epsilon_{outer} = 1.4 \times 10^{-4}$, λ^0 =1000, w_δ^0 =1500, $w_{v7,ADG1}^0 = 30$, $w_{v16,ADG2}^0 = 10$, and evaluate the convergence behaviors with respect to the step size Γ. Parameter Γ reflects the level of dependency of target-response variables in each iteration to their values obtained in the previous iteration. We vary Γ in the range of $\{0.6, 0.7\ \dots, 0.8, 0.99\}$. Figure 2.10(b) shows that, in general, increasing Γ decreases the number of iterations and computational time. For DQA and TDQA, the least number of iterations is obtained by setting Γ= 0.9. DQA has more stability to variation of Γ. This is because of the existence of the inner loop in which the algorithm seeks to reduce the error between the target and response values without updating the penalty multipliers.



(a)



(b)

Figure 2.10. a) Iteration and *rel* index vs. stopping criterion, and b) iterations vs. Γ.

Setting different initial values for the penalty multipliers changes the speed of the algorithms and accuracy of the obtained results. We select different initial values for multipliers λ and w and the step size Γ and calculate the rel index. Figure 2.11 shows a contour plot of the rel index versus variations of initial values of λ, w, and Γ. If λ^0=1000 and w_δ^0=1000, setting the step size Γ to 0.6 provides the least error ($rel = 2.48 \times 10^{-4}$) after and 51 iterations. If the user selects the parameter badly (e.g., λ^0=2500, w_δ^0=2500), Γ equal to 0.5 yields the relative error of 0.0054 within 75iterations. Note that although we get the least rel with $\Gamma = 0.6$, λ^0=1000, and w_δ^0=1000, it takes a relatively long time to converge.

Case 4: We consider a full AC OPF and implement the proposed algorithm and three other methods, namely ADMM(Boyd, Parikh et al. 2011), APP(Kim and Baldick 1997), and ALAD (that is based on ATC)(Tosserams, Etman et al. 2006). We compare the TDQA-based TSO+DSO optimal power flow to OPF solved by the other three methods (Kargarian, Mohammadi et al. 2017). Although all these four methods are based on the augmented Lagrangian relaxation, TDQA and APP solve the problem in a fully parallel manner while ADMM and ALAD are sequential solution algorithms. This means that in iteration k of ADMM and ALAD, DSOs cannot solve their subproblems without having the TSO's shared variable values in iteration k, whereas in TDQA and APP, TSO and DSOs solve their subproblems in parallel as they need their neighbors' shared variable values determined in iteration $k-1$. Each iteration of TDQA and APP takes around 0.1 seconds while it is 0.16 seconds for ADMM and ALAD. That is, each iteration of TDQA (and APP) is generally faster than that in ALAD and ADMM. Figure 2.12 shows the rel index after 100 iterations. All four methods converge to acceptable rel indices. APP takes many iterations for the rel index to go below an acceptable threshold, whereas the other three methods reach an acceptable threshold after the first few iterations. Since ALAD and ADMM are sequential algorithms and TDQA is a parallel one, the solution time of TDQA, in each iteration, is faster than that for ALAD and ADMM. In the simulations, we have tried to provide reasonable and fair conditions to compare the four algorithms. Although the TDQA algorithm shows a good performance for the considered TSO+DSO OPF In this chapter, we cannot make a solid conclusion that which algorithm has a better overall performance. The performance of the algorithms depends on the type of the problem and initial values of variables and penalty multipliers/factors (refer to [3] for more details).

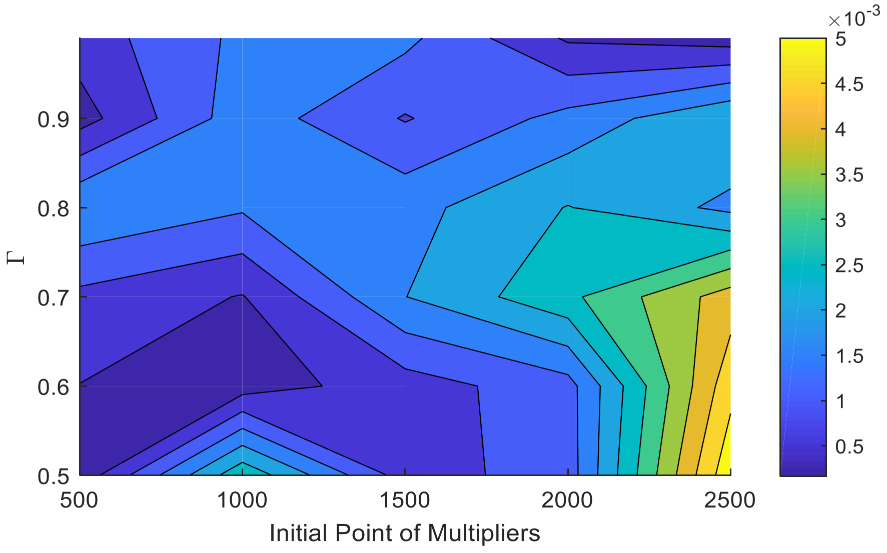


Figure 2.11. Convergence measure rel versus initial values of multipliers and Γ.

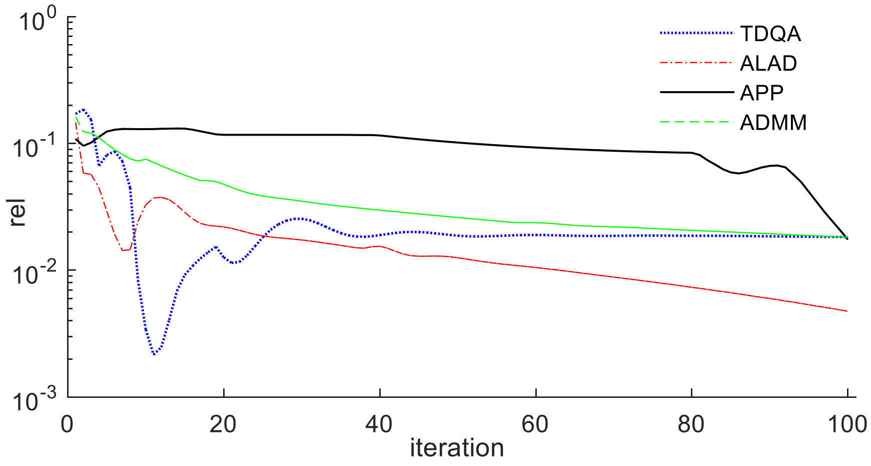


Figure 2.12. Comparison between TDQA, APP, ADMM, and ALAD.

It should be noted that the considered OPF problem in this chapter has two levels. For problems with several levels of hierarchy, for instance, if TSO (level 1), DSOs (level 2), and microgrids (level 3) want to solve a collaborative OPF, the parallel solution algorithm, such as TDQA, is expected to be faster than the sequential one.

2.5.2 IEEE 118-bus system

Cold start: This test system comprises 31 autonomous systems: one TSO and 30 DSOs. The transmission system includes 118 buses, 186 lines, and 54 generators. Each active distribution grid consists of two generating units. We use a cold start, i.e., the initial values of the local variables of TSO and DSOs as well as targets/responses (i.e., shared variables) are set to zero. The initial values for penalty multipliers are selected as λ^0 =100, w^0=100, Γ=0.6, and the stopping thresholds are $\epsilon_{inner} = 0.2$ and $\epsilon_{outer} = 0.002$. As shown in Table 2.3, the collaborative ATC-based TSO+DSO operation with the DQA coordination strategy converges after 66 outer loop and 122 inner loop iterations (around 2 seconds). If we use the TDQA coordination strategy, the algorithm takes 99 iterations (only outer loop) to converge after less than 2 seconds). The overall operating cost of the system determined by DQA and DTQA is \$5,115.6 and \$5,120.6, respectively. To evaluate the accuracy of results, we ignore autonomy and information privacy of TSO and DSOs and solve the centralized OPF. The overall operating cost is \$5,098.6. Although the error of DQA and TDQA is negligible, DQA is slightly more accurate.

Hot start: Since in practice, we can use a hot start scenario, we are potentially capable of speeding up the convergence process. We choose appropriate initial values and run a hot start scenario. The results show that TDQA and DQA respectively take 11 and nine iterations to converge, which is much faster (almost ten times) than the cold start scenario. Figure 2.13 shows the rel index over the course of iterations.

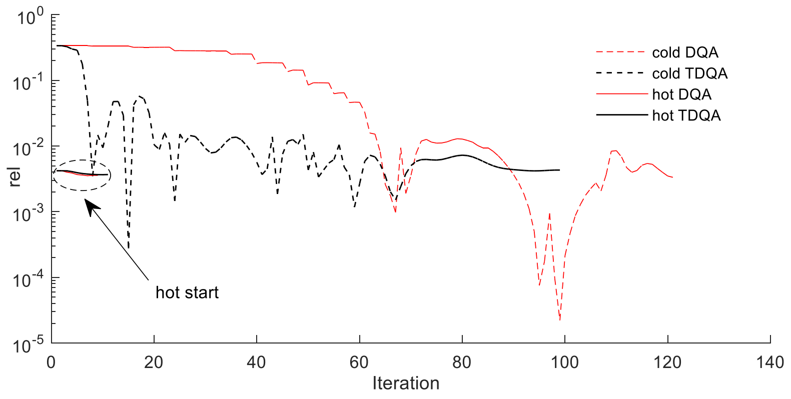


Figure 2.13. rel index for the IEEE 118-bus system (cold and hot start scenarios).

Table 2.3. TSO+DSO OPF for the IEEE 118-bus

Algorithm		Total cost	Iteration
Centralized TSO+DSO OPF		\$5,098.6	-
Decentralized TSO+DSO OPF cold start	DQA	\$5,116.5	92
	TDQA	\$5,120.6	98
Decentralized TSO+DSO OPF hot start	DQA	\$5,117.4	9
	TDQA	\$5,116.8	11

2.6 Conclusion

Power systems are being transformed into distributed energy infrastructures in which electricity is generated in transmission and distribution levels. This chapter presents a decentralized OPF algorithm for the collaborative management of transmission and distribution systems. The proposed algorithm is based on analytical target cascading (ATC) for multilevel hierarchical optimization. The OPF problem associated with TSO is formulated in the upper level of the hierarchy while OPFs of DSOs are assigned to the lower level. As TSO and DSOs are autonomous systems, information privacy plays a critical role in joint management decisions. In the proposed framework, TSO and DSOs exchange only limited information, namely target and response (shared) variables. These entities do not need to reveal their commercially sensitive information to other parties. Two coordination strategies, namely DQA and TDQA, are presented to coordinate TSO and DSOs in a decentralized manner. DQA and TDQA allow parallel execution of local OPF problems (associated with TSO and DSOs).

The numerical tests on a 6-bus and the IEEE 118-bus systems show the accuracy and convergence performance of our proposed ATC-based collaborative TSO+DSO OPF method. Although both DQA and TDQA coordination strategies provide promising results, the collaborative TSO+DSO operation with TDQA usually determines optimal results with almost the same accuracy as DQA while usually taking fewer iterations. Using a hot start scenario to initialize target/response pairs and multipliers enhances the solution speed of the proposed TSO+DSO operation algorithm, especially for large systems.

CHAPTER 3. ACCELERATED AND ROBUST ANALYTICAL TARGET CASCADING FOR DISTRIBUTED OPTIMAL POWER FLOW

3.1 Introduction

Distributed optimization algorithms are sensitive to the choice of initial values and the level of importance of each term in objective functions. If initial values, in particular, penalty parameters, are not set appropriately and the level of importance of objective terms are not balanced, the algorithm may converge slowly, oscillate around the optimal point, or diverge. This chapter presents an accelerated, robust analytical target cascading (AR-ATC) to solve optimal power flow (OPF) distributedly. A function is designed to determine a balancing coefficient with respect to initial values. Incorporating this coefficient in ATC makes a tradeoff between the convergence speed and solution accuracy by adjusting penalty terms with respect to generation cost functions. If multipliers are initialized to large values, the proposed function creates a balancing coefficient to avoid premature convergence or divergence. If multipliers are initialized to small values, the proposed function adjusts them to enhance the convergence speed. We name the proposed algorithm an accelerated, robust ATC since it enhances the solution speed and makes it more robust against initialization. Mathematical justifications and simulation studies are performed to analyze the effectiveness of AR-ATC. Potential applications of the proposed method to other distributed approaches such as ATC with exponential penalty functions and auxiliary problem principle (APP) is also studied numerically.

Nomenclature

Indices, Sets, Parameters, and Functions:

i, j	Index for subproblem j at level i.
k	Index for iterations.
u	Index for generating units.

Ω_{ij}	Set of generators in areas j at level i.
a_u, b_u, d_u	Cost function coefficients of generating unit u.
$D(\lambda)$	Dual function.
$\nabla D(\lambda)$	Gradient of dual function.
f_{ij}	Generation cost function of subproblem j at level i.
f_1, f_2	Objective function of subproblems one and two.
f_1^*, f_2^*	Conjugate of f_1 and f_2.
∇f	Gradient of f.
g_{ij}	Inequality constraints of subproblem j at level i.
h_{ij}	Equality constraints of subproblem j at level i.
π	Penalty function.
SP_{ij}	Subproblem j located at level i.
α	Balancing coefficient.
β	Constant coefficient.
λ	Lagrange multipliers of ALAD.
λ_{AR}	Lagrange multiplier of AR-ALAD.
L	Lipschitz constant.
ω	Penalty multipliers.
$z^\star$	Optimal value of z.

Variables:

p_u	Power generated by unit u.
r_{ij}	Response variables of subproblem j at level i.
t_{ij}	Target variables of subproblem j at level i.
x	Set of all variables of the centralized problem.
x_{ij}	Set of variables of subproblem j at level i.

3.2 Contribution

The contributions of this chapter are summarized as follows:

- An accelerated, robust analytical target cascading is proposed to make ATC more robust to the tuning parameter.
- A new function is designed to make a tradeoff between the convergence speed and solution accuracy by adjusting penalty terms with respect to generation cost functions.

3.3 OPF in the Context of ATC

Assume that a power system is divided into several areas. Depending on interconnections between the areas, each area is assigned to a level to create a multilevel hierarchical structure. These levels refer to computational levels that determine the sequence of solving OPF subproblems (SPs) corresponding to the areas (Tosserams, Etman et al. 2006, Kargarian, Mohammadi et al. 2018). Although the application of ATC was successfully demonstrated to the OPF problem and many engineering problems (Tosserams, Etman et al. 2006, DorMohammadi and Rais-Rohani 2013, Kargarian, Mohammadi et al. 2018), research on improving its convergence performance is ongoing.

Consider the following compact centralized OPF problem.

$$\min f(x) = \sum_{\forall u} a_u p_u^2 + b_u p_u + d_u \tag{3.1}$$

$$s.t \quad h(x) = 0$$

$$g(x) \leq 0$$

$h(x)$ is the nodal power balance equations for the whole system, and $g(x)$ is limits of generators, line flows, and bus voltages. Without loss of generality and for the sake of explanation, we decompose (3.1) into two OPF subproblems SP_{11} and SP_{22}.

SP_{11}:

$$\min \underbrace{\sum_{\forall u \in \Omega_{11}} a_u p_u^2 + b_u p_u + d_u}_{f_{11}} + \pi(t_{22} - r_{22}) \tag{3.2}$$

$$\begin{aligned} s.t. \quad & h_{11}(x_{11}, t_{22}) = 0 \\ & g_{11}(x_{11}, t_{22}) \leq 0 \end{aligned}$$

SP_{22}:

$$\min \underbrace{\sum_{\forall u \in \Omega_{22}} a_u p_u^2 + b_u p_u + d_u}_{f_{22}} + \pi(t_{22} - r_{22}) \tag{3.3}$$

$$\begin{aligned} s.t. \quad & h_{22}(x_{22}, r_{22}) = 0 \\ & g_{22}(x_{22}, r_{22}) \leq 0 \end{aligned}$$

t_{22} are a copy of shared variables between SP_{11} and SP_{22} that are variables in SP_{11} and known in SP_{22}. r_{22} is another copy of shared variables that are variables in SP_{22} and known in SP_{11}. For DC OPF, for instance, shared variables are voltage angles of tie-lines terminals connecting areas one and two. The mismatch between shared variables is forced by consistency constraints, $c = t_{22} - r_{22}$, that are relaxed in local objective functions by augmented Lagrangian penalty term $\pi(t_{22} - r_{22})$. In the ATC framework, different options exist for modeling π, and several coordination strategies can be applied to coordinate subproblems. We select quadratic penalty terms and ALAD coordination strategy (Tosserams, Etman et al. 2006, Boyd, Parikh et al. 2011).

$$\pi(t - r) = \lambda^{\dagger}(t - r) + \|\omega \circ (t - r)\|_2^2 \tag{3.4}$$

The symbol † denotes transpose and ∘ is the Hadamard product. ALAD is a message-passing type synchronous distributed approach (Tosserams, Etman et al. 2006). It is a one-loop sequential coordination strategy that solves upper-level subproblems, updates t, propagates t toward lower-level subproblems, solves lower level-subproblem given t values, and updates r. If $|t - r| < \epsilon$, ALAD stops, otherwise, multipliers are updated according to the concept of the method of multipliers (Bertsekas 1999, Tosserams, Etman et al. 2006).

$$\lambda^{k+1} = \lambda^k + 2(\omega^k)^2(t^k - r^k) \qquad , \qquad \omega^{k+1} = \beta\omega^k \tag{3.5}$$

Penalty multiplier ω is used as the step size to update λ, and β is a constant term in the range of $\beta \geq 1$ (Bertsekas 1999, Tosserams, Etman et al. 2006). Selecting a large β might endanger the convergence. We recommend selecting β equal or close to one. We refer to (Kargarian, Mohammadi et al. 2018) for more details regarding formulating OPF in the context of ATC.

3.4 Accelerated and Robust ALAD

3.4.1 Intuitions for a Balancing Coefficient

If t and r converge together very fast, the ALAD feasibility criteria are satisfied after a few iterations; however, it does not ensure the optimality of results. Incorporating a proper coefficient in the objective functions of subproblems to make a balance between penalty terms π (corresponding to feasibility) and generation cost functions f (corresponding to optimality) can enhance the performance of ALAD. Intuitively, this is similar to a scaled form of gradient descent in which a coefficient is used to modify the step size and enhance the condition number (Bertsekas 1999).

Remark 1: The closer the condition number is to one, the faster gradient-based algorithms will be (Bertsekas 1999).

Remark 2: For a two-dimensional problem, for instance, the closer the condition number is to one, the closer the contour plot is to a circle. If the condition number is large, the contour plot will be elliptical, with one dimension being much more dominant (i.e., elongated) than another one.

Remark 3: In augmented Lagrangian, setting penalty multipliers to large values increases the problem condition number and may cause ill-condition.

3.4.2 ALAD with Balancing Coefficient α

Consider two OPF subproblems that are linked via a shared variable[2]. f_1 is the strictly convex cost function of subproblem one in level one, and f_2 is the strictly convex cost function of subproblem two in level two (we drop the level index for brevity). Variable t denotes the target of subproblem one, and r is the response of subproblem two. Let us consider only consistency constraint $c = t - r = 0$. The centralized problem can be formulated by minimizing $f = f_1 + f_2$ subject to c. We rewrite this problem as follows by adding an arbitrary coefficient α:

$$\begin{aligned} &\min \left(\frac{1}{\alpha}\right) f_1\,(t) + \left(\frac{1}{\alpha}\right) f_2(r) \\ &s.t. \quad t - r = 0 \end{aligned} \tag{3.6}$$

In the rest of this section, we investigate the impact of α on ALAD's behavior, Lipschitz continuity, and convergence rate. We also define some properties for an α that can enhance the convergence performance of ALAD. We name the coordination algorithm with the coefficient α AR-ALAD to differentiate it with the classical ALAD. AR-ALAD and ALAD are the same if $\alpha = 1$. The augmented Lagrangian of (3.6) is as follows:

[2] Although the discussions in Section 3.4 are based on a convex form of OPF, we have shown in Sections 3.5 and 3.6 that the proposed method can enhance the convergence performance of nonconvex nonlinear mathematical problems and AC OPF.

$$L(t, r, \lambda_{AR}) = \min_{t,r} \left(\frac{1}{\alpha}\right) f_1\,(t) + \left(\frac{1}{\alpha}\right) f_2(r) + \lambda_{AR}(t - r) + \omega^2(t - r)^2 \tag{3.7}$$

where t and r are variables, and λ_{AR} and ω are constant. The coordination strategy is based on the method of multipliers (Bertsekas 1999), and we have:

$$t^+ = t^{k+1} = \underset{t}{\operatorname{argmin}} \left(\frac{1}{\alpha}\right) f_1(t) + \underbrace{\lambda_{AR}(t - r^k) + \omega^2(t - r^k)^2}_{\pi(t-r)} \tag{3.8}$$

$$r^+ = r^{k+1} = \underset{r}{\operatorname{argmin}} \left(\frac{1}{\alpha}\right) f_2(r) + \underbrace{\lambda_{AR}(t^{k+1} - r) + \omega^2(t^{\,k+1} - r)^2}_{\pi(t-r)} \tag{3.9}$$

$$\lambda_{AR}^{k+1} = \lambda_{AR}^k + 2\omega^2(t^{k+1} - r^{k+1}) \tag{3.10}$$

where t^k and r^k are target and response values achieved at iteration k. Based on Karush-Kahn-Tucker (KKT) conditions, we calculate t^{k+1} and r^{k+1} as follows:

$$\left(\frac{1}{\alpha}\right) \partial f_1(t^{k+1}) + \lambda_{AR}^k + 2\omega^2(t^{k+1} - r^k) = 0 \rightarrow \partial f_1(t^{k+1}) = -\alpha\lambda_{AR}^{1/2} \tag{3.11}$$

$$\left(\frac{1}{\alpha}\right) \partial f_2(r^{k+1}) - \lambda_{AR}^k - 2\omega^2(t^{k+1} - r^{k+1}) = 0 \rightarrow \partial f_2(r^{k+1}) = \alpha\lambda_{AR}^{k+1} \tag{3.12}$$

where we define $\lambda_{AR}^{1/2} = \lambda_{AR}^k + 2\omega^2(t^{k+1} - r^k)$. At the optimal point, we have:

$$\partial f_1(t^\star) = -\alpha\lambda_{AR}^\star \tag{3.13}$$

$$\partial f_2(r^\star) = \alpha\lambda_{AR}^\star \tag{3.14}$$

We conclude that $\lambda^\star = \alpha\lambda_{AR}^\star$ where $\lambda^\star$ is the optimal value of ALAD Lagrange multiplier. By implementing the conjugate duality on (3.13) and (3.14), we obtain:

$$t^+ = -\nabla {f_1}^*(\alpha\lambda_{AR}^{1/2}) \tag{3.15}$$

$$r^+ = \nabla {f_2}^*(\alpha\lambda_{\mathrm{AR}}^+) \tag{3.16}$$

To analyze how the algorithm moves toward optimality, we derive the conjugate dual of (3.7) as:

$$D(\lambda) = -\left(\frac{1}{\alpha}\right) f_2{}^*(\alpha\lambda_{AR}) - \left(\frac{1}{\alpha}\right) f_1^*(\alpha\lambda_{AR}) \tag{3.17}$$

To obtain $\lambda_{AR}^{\star}$, we use gradient ascent and update λ_{AR} iteratively as follows:

$$\lambda_{AR}^{+} = \lambda_{AR} + \omega^2 \nabla D(\lambda^+) \tag{3.18}$$

Taking gradient of (3.17) and plugging (3.15) and (3.16) into that yields:

$$\nabla D(\lambda^+) = -\nabla f_2{}^*(\alpha\lambda_{AR}^+) - \nabla f_1{}^*(\alpha\lambda_{AR}^+) \cong t^+ - r^+ \tag{3.19}$$

We can now rewrite (3.18) as:

$$\lambda_{AR}^{+} = \lambda_{AR} + \omega^2 \nabla D(\lambda^+) = \frac{\lambda}{\alpha} + \omega^2(t^+ - r^+) \tag{3.20}$$

We can maximize $D(\lambda)$ or minimize $-D(\lambda)$. For ease of notation, we use $\widehat{D}(\lambda) = -D(\lambda)$. $\widehat{D}(\lambda)$ is a strictly convex function. Based on gradient descent, we minimize $\widehat{D}(\lambda)$ as:

$$\nabla \widehat{D}(\lambda^+) = r^+ - t^+ \tag{3.21}$$

Assume that $\nabla \widehat{D}(\lambda)$ is Lipschitz continuous and $\nabla \widehat{D}(\lambda_{AR}) \le L_{AR}$. Incorporating α results in $L_{AR} = \phi(\alpha)\, L$ where $\phi(\alpha)$ is a monotonically increasing function. Based on convexity and Lipschitz properties, we have:

$$\widehat{D}(\lambda^+) \le \widehat{D}(\lambda) + \nabla \widehat{D}(\lambda)\left(\frac{\lambda^+}{\alpha} - \frac{\lambda}{\alpha}\right) + \frac{1}{2}\phi(\alpha)\, L\widehat{D}(\lambda)\left\|\frac{\lambda^+}{\alpha} - \frac{\lambda}{\alpha}\right\|^2$$

$$= \widehat{D}(\lambda) + \nabla \widehat{D}(\lambda)\big(\lambda - \omega^2 \nabla \widehat{D}(\lambda) - \lambda\big) + \frac{1}{2}\phi(\alpha) L\big(\lambda - \omega^2 \nabla \widehat{D}(\lambda) - \lambda\big)^2$$

$$= \widehat{D}(\lambda) - \left(1 - \tfrac{1}{2}\phi(\alpha) L\omega^2\right)\omega^2\, \nabla \widehat{D}(\lambda)^2 \tag{3.22}$$

The minimum point will be achieved if $\widehat{D}(\lambda)$ decreases over iterations, i.e., $\widehat{D}(\lambda^{k+1}) \leq \widehat{D}(\lambda^k)$. To this end, $1 - \frac{1}{2}\phi(\alpha)L\omega^2 > 0$ and ω^2 should be less than $1/(\phi(\alpha)L)$. Selecting $\omega^2 = 1/(\phi(\alpha)L)$ satisfies this positivity condition. By plugging this ω^2 in (3.22), we have:

$$\widehat{D}(\lambda^+) \leq \widehat{D}(\lambda) - \frac{1}{2}\omega^2 \, \nabla\widehat{D}(\lambda)^2 \tag{3.23}$$

Since $\frac{1}{2}\omega^2 \, \nabla\widehat{D}(\lambda)^2$ is always positive or zero, after some iterations, $\widehat{D}(\lambda) = \widehat{D}(\lambda^\star)$ with $\lambda^\star$ denoting the optimal value for λ. This proves convergence. To find the convergence rate, setting $\omega^2 = 1/(\phi(\alpha)L)$ in (3.22) and borrowing convex properties, we can rewrite (3.22) for $\lambda^\star$ as:

$$\widehat{D}(\lambda^+) \leq \widehat{D}(\lambda^\star) + \nabla\widehat{D}(\lambda)\left(\frac{\lambda}{\alpha} - \frac{\lambda^\star}{\alpha}\right) - \frac{1}{2}\omega^2\left(\nabla\widehat{D}(\lambda)\right)^2 \rightarrow$$

$$\widehat{D}(\lambda^+) - \widehat{D}(\lambda^\star) \leq \frac{1}{2\omega^2}\left(2\omega^2\nabla\widehat{D}(\lambda)\left(\frac{\lambda}{\alpha} - \frac{\lambda^*}{\alpha}\right) - \omega^4\nabla\widehat{D}(\lambda)^2\right) \rightarrow$$

$$\widehat{D}(\lambda^+) - \widehat{D}(\lambda^\star) \leq \frac{1}{2\omega^2}\left(\frac{\lambda}{\alpha} - \frac{\lambda^\star}{\alpha}\right)^2 - \left(\frac{\lambda}{\alpha} - \omega^2\nabla\widehat{D}(\lambda) - \frac{\lambda^*}{\alpha}\right)^2 \rightarrow$$

$$\widehat{D}(\lambda^+) - \widehat{D}(\lambda^\star) \leq \frac{1}{2\omega^2}\left(\left(\frac{\lambda}{\alpha} - \frac{\lambda^\star}{\alpha}\right)^2 - \left(\frac{\lambda^+}{\alpha} - \frac{\lambda^\star}{\alpha}\right)^2\right) \tag{3.24}$$

By adding summation over iterations, we have:

$$\sum_{i=2}^{K}\widehat{D}(\lambda^i) - \widehat{D}(\lambda^\star) \leq \frac{1}{2\omega^2}\left(\sum_{i=2}^{K}\left(\frac{\lambda^{i-1}}{\alpha} - \frac{\lambda^\star}{\alpha}\right)^2 - \left(\frac{\lambda^i}{\alpha} - \frac{\lambda^\star}{\alpha}\right)^2\right) \tag{3.25}$$

We use λ^1 of ALAD as the initial value of the proposed method. To have a fair comparison with ALAD, we should start from λ^0. By plugging λ^0 in (3.25), using the telescopic sum, and after simplifications, we obtained the convergence rate as follows:

$$\left(\widehat{D}(\lambda^k) - \widehat{D}(\lambda^\star)\right) \leq \frac{1}{2\omega^2 k}\left(\left\|\lambda^0 - \frac{\lambda^\star}{\alpha}\right\|_2^2\right) \tag{3.26}$$

We conclude that AR-ALAD and ALAD have the same behavior when $\omega^2 = \alpha\omega_{AR}^2$ and $\lambda^0 = 0$.

3.4.3 Required Properties of Balancing Coefficient α

Expression (3.26) indicates that if $\alpha > 1$, AR-ALAD converges faster than ALAD. However, $\alpha \gg 1$ might endanger the Lipschitz continuity as $L_{AR} = \phi(\alpha)L$ and $\phi(\alpha)$ grows monotonically by increasing α. Since a large α increases the Lipschitz constant, it limits the upper bound of ω as ω^2 must be less than $1/(\phi(\alpha)L)$ to ensure the convergence of the augmented Lagrangian method (see (3.22)). If one selects $\alpha < 1$, it decreases the impact of ω. Although it is effective to find accurate results, it slows the convergence rate.

To demonstrate the effect of scaling on the Lipschitz continuity, we consider $f_1(t)$ and $f_2(r)$ to be quadratic functions. The dual function (3.17) is restated as:

$$D(\lambda) = \frac{-\frac{1}{2}(\mathrm{b}_1 + \alpha\lambda_{AR})^2\mathrm{a}_1^{-1} - \frac{1}{2}(\mathrm{b}_2 - \alpha\lambda_{AR})^2\mathrm{a}_2^{-1}}{\alpha} \tag{3.27}$$

where a_1 and a_2 are coefficients of quadratic terms and b_1 and b_2 are coefficients of linear terms of f_1^* and $f_2{}^*$, respectively. The Lipschitz of $D(\lambda)$ is:

$$L = \frac{1}{2}\left(-2b_1a_1^{-1} + 2b_2a_2^{-1} + \alpha\left(-\lambda_{AR,1}a_1^{-1} - \lambda_{AR,2}a_1^{-1} - \lambda_{AR,1}a_2^{-1} - \lambda_{AR,2}a_2^{-1}\right)\right) \tag{3.28}$$

Based on (3.28), increasing α increases L monotonically. Also, based on (3.22), increasing α makes the ω selection bound tighter. In other words, increasing α arbitrary for a specific ω will not improve the convergence performance and may endanger the convergence property. Furthermore, α affects the condition number. A very small or very large α may cause ill-condition. This is similar to the augmented Lagrangian relaxation concept in which selecting a large penalty factor is desirable, but it may result in ill-condition and gradient descent based solvers may have difficulties to direct the solution toward the optimal point iteratively (Bertsekas 1999).

3.5 Proposed Function to Calculate α

Finding the optimal (or a good enough) value for balancing coefficient α is not straightforward. This coefficient is problem dependent and its optimal value is unknown before solving a problem. For OPF, α depends on the system configuration and condition, e.g., loading condition. In this section, we propose a function to determine a proper balancing coefficient α that satisfies the required features mentioned in Section 3.4. The impact of this coefficient on the ATC-based ALAD coordination strategy is investigated using a mathematical benchmark problem and DC OPF for a typical power system.

3.5.1 Intuition Behind Proposed Function

Consider solving a centralized two-variable optimization with gradient descent. If the contour of the objective function is circular, the algorithm converges to the optimal point after a few steps. Since at each iteration the gradient direction is orthogonal to the curve, if the contour becomes elongated, the pattern of search direction may become zigzag with small improvement after each iteration or go toward a wrong direction. Consider a simple illustrative example.

$$\min_{x,y} f(x,y) = f_1(x) + f_2(y) \tag{3.29}$$

$$\text{where} \quad f_1(x) = x^2 + 2x \;\; \& \;\; f_2(y) = 0.03y^2 + 0.05y$$

We analyze two models: the original unscaled problem and a scaled form of the problem. Since f_1 and f_2 are in different orders, the contour plot of the unscaled problem is elongated as shown in Fig. 3.1a. Let us evaluate f_1 and f_2 after the first iteration and then scale the problem as follows:

$$\min_{x,y} f(x,y) = f_1(x) + \alpha f_2(y) \tag{3.30}$$

$$\text{where} \quad \alpha = \frac{f_1(x_0)}{f_2(y_0)}$$

Assume an initial point with $x_0 = 8$ and $y_0 = 8$. To demonstrate the effect of scaling, we investigate two cases with large (0.8) and small (0.2) step sizes. Figures 3.1(a) and (b) demonstrate the gradient descent search after ten iterations with step size=0.8. For the unscaled problem, since the surface is elongated, gradient descent follows a zigzag path with a small improvement after each iteration. After ten iterations, the relative error of the objective function as compared to the optimal solution is 0.95. For the scaled problem with a more circular contour (see Fig. 3.1b), a good enough solution is obtained after several big jumps and the relative objective error becomes 0.0092. A more detailed analysis of Fig. 3.1 shows that after several iterations x becomes close to its optimal value while there is a small improvement in y. This is due to the unbalanced importance levels of $f_1(x)$ and $f_2(y)$ that makes the algorithm more sensitive to $f_1(x)$. Hence, optimization pays more attention to minimizing $f_1(x)$ than $f_2(y)$.

For the unscaled problem with small step size, gradient descent moves slowly toward the optimal point with more dominant attention to $f_1(x)$ than $f_2(y)$. After ten iterations, the obtained solution is far from the optimal point. The obtained x value is good, but y is still far from the optimal point. Finding the optimal solution of the unscaled problem takes many iterations. But, for the scaled problem, from the first iteration, the search direction is along to the optimal point and a high-quality solution is obtained after ten iterations.

This scaling concept and the illustrative example provide us with intuitions to select the balancing coefficient α for modifying ALAD. In OPF, generation cost functions and penalty functions are not in the same order of magnitude since they have different natures. To make a balance between generation costs and penalty terms, we borrow the scaling idea and propose a function to calculate balancing coefficient α.

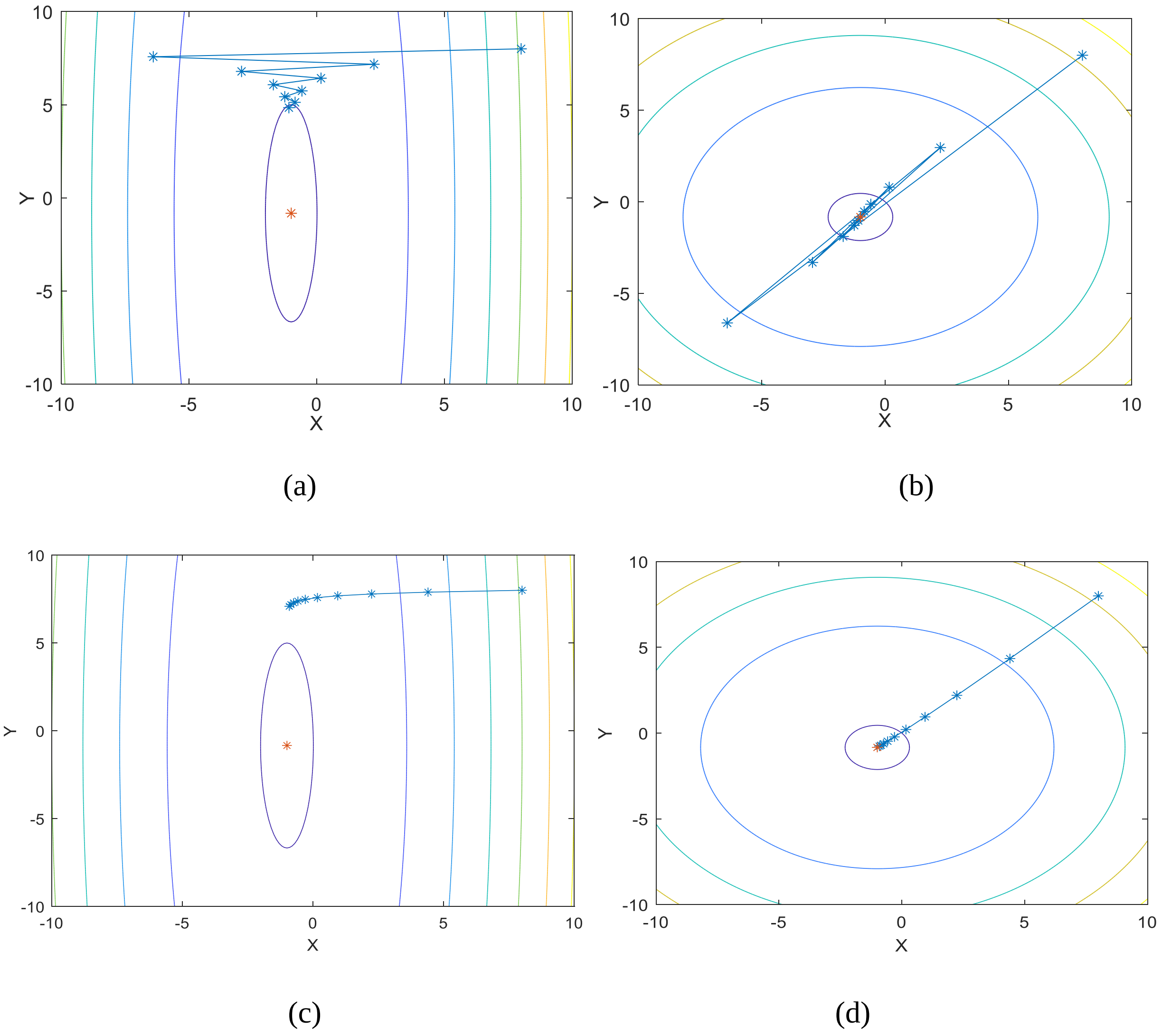


Figure 3.1 Contour plot for a) unscaled (step size=0.8), b) scaled (step size=0.8), c) unscaled (step size=0.2) and d) scaled (step size=0.2) problem.

3.5.2 Proposed Function

In ALAD, each subproblem consists of two terms, cost function f and penalty term π. Although the goal is to minimize f, the impact of π should not be neglected. Finding a good set of penalty multipliers to make a tradeoff between convergence speed, optimality, and feasibility is not straightforward. In the proposed approach, arbitrary values are selected for t_{ij}^0, r_{ij}^0, λ^0, and ω^0. The first iteration of the distributed algorithm is carried out to find x_{ij}^1, t_{ij}^1, and r_{ij}^1, and the balancing coefficient α is calculated by (3.31).

$$\alpha = \left| \frac{\sum_{\forall i} \sum_{\forall j} f_{ij}^1 \left(x_{ij}^1 \right)}{\sum_{\forall i} \sum_{\forall j} \pi_{ij}^1 \left(t_{ij}^1 - r_{ij}^1 \right)} \right| \tag{3.31}$$

where f_{ij}^1 and π_{ij}^1 are the values of cost function and penalty terms of subproblem i in level j at the end of iteration one. The basic idea is to make a balance between f and π at the beginning of iteration two. If $f_{ij}^1 \gg \pi_{ij}^1$ the priority is given to f_{ij}^1, and if $f_{ij}^1 \ll \pi_{ij}^1$ the priority is given to π_{ij}^1. The balancing coefficient α, which equal or larger than zero, assigns roughly the same priority to f and π. Consider that penalty multipliers ω are set to large values. This yields $f_{ij}^1 \ll \pi_{ij}^1$ and the proposed function produces an α less than one to reduce the level of importance of π and assign more weight to optimality (i.e., f). On the other hand, if penalty multipliers ω are initialized to small values, $f_{ij}^1 \gg \pi_{ij}^1$ and the proposed function produces an α larger than one to assign more weight to feasibility. We have investigated several alternatives for calculating an appropriate balancing coefficient and have observed that (3.31) is a suitable approach. Developing more sophisticated but efficient methods can be considered as future work.

We name the proposed algorithm with balancing coefficient α accelerated, robust ALAD (AR-ALAD) as it enhances the convergence speed of ALAD, prevents obtaining suboptimal results, and reduces the possibility of divergence when inappropriate initial conditions are selected. The AR-ALAD coordination strategy is shown in the following Algorithm.

Algorithm ATC-based AR-ALAD coordination strategy

```
1. **initialize** $x, t, r, \lambda, \omega, \alpha = 1$ and $k = 0$
2. Solve all subproblems $ij$
3. Update $\alpha = \sum_{\forall i}\sum_{\forall j} f_{ij}^{1}\,(x_{ij}^{1}) \,/\, \sum_{\forall i}\sum_{\forall j} \pi_{ij}^{1}\,(t_{ij}^{1} - r_{ij}^{1})$
4. **while** max $(\|t^{k} - r^{k}\|) \le \epsilon$ & $\left|(f_{ij}^{k} - f_{ij}^{k-1})/f_{ij}^{k-1}\right| \le \epsilon$, $k \leftarrow k + 1$ **do**
5.     Solve $f_{ij} + \alpha\pi_{ij}$ (or $\frac{1}{\alpha} f_{ij} + \pi_{ij}$) for each subproblem $ij$ sequentially
6:     Update multiplier $\lambda^{k+1} = \lambda^{k} + 2\omega^{2}(t^{k} - r^{k})$ & $\omega^{k+1} = \beta\omega^{k}$
7: **end while**
```

Remark 4: To consider the impact of penalty multipliers, initial values of variables, problem characteristics, and constraints, we use values obtained after running the algorithm for one iteration to calculate α. Our experience shows that this procedure provides a better α than other possible approaches. α varies with changing initial values of variables and penalty multipliers.

The features of AR-ALAD are investigated using two illustrative examples, a nonlinear mathematical benchmark problem (the first problem on page 6 of (DorMohammadi and Rais-Rohani 2013)), and DC OPF for a 22-bus power system . The mathematical problem includes four subproblems with quadratic cost functions and nonlinear constraints, and the power system is decomposed into two levels with one OPF subproblem in level one and two OPF subproblems in level two. We use several performance indices and statistical tests to analyze various features (i.e., robustness, acceleration, feasibility, and optimality) of AR-ALAD.

3.5.3 Robustness

A distributed algorithm should provide good results for different initial penalty multipliers. We call such a feature *robust* to penalty multipliers variation. To compare the robustness of AR-ALAD with ALAD, we run F-test, principal component analysis (PCA), and an analysis to determine the bound in which the solution varies with respect to ω.

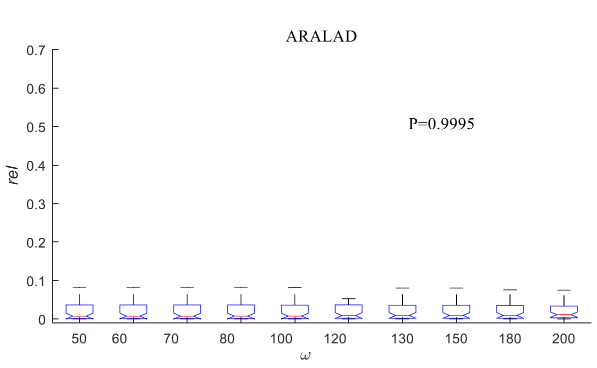
F-test is a statistical model that measures whether a group of variables are statistically (by observing mean values and variances) similar or different. In other words, it estimates the significance of a set of population-based on its average. F-test measurements can be carried out by analysis of variance or regression analysis (ANOVA) (Anscombe 1948). The most important factor in F-test is the P-value that demonstrates a big picture of a given dataset and determines the level of difference of data in that set. A typical threshold for P is 0.05, which means that if the P-value is smaller than 0.05, all values in the given dataset are significantly different. When P is equal to one, elements of the dataset are roughly the same statistically. We implement F-test with respect to ω. To generate the samples, we select ten different values for ω as {50, 60, 70, 80, 100, 120, 130, 150, 180, 200}, run ALAD and AR-ALAD based DC OPF for 100 iterations for each ω, and calculate the convergence index rel (see (2.22)) at each iteration.

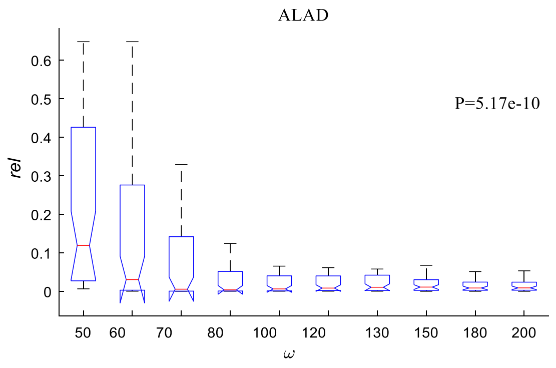
For the mathematical benchmark problem, we select different ω as {190, 230, 270, 310, 350, 390, 430, 470}, run ALAD and AR-ALAD for 50 iterations, and calculate the rel index at each iteration. Figures 3.2 and 3.3 demonstrate F-test results. Red lines between blue boxes show the mean value of rel for a specific ω, and black dashed lines depict the range of rel. For both OPF and the mathematical benchmark problem, while rel of ALAD varies considerably by changing ω, AR-ALAD is less sensitive to the variation of ω. A similar pattern is observed for the mean value of rel. For OPF, the average of rel for AR-ALAD is close to zero, and its upper bound is less than 10%. It means that the algorithm is less dependent on ω and starts from a good rel. F-test is also used to measure similarities between the obtained results. The P-value for AR-ALAD is 0.9995 for OPF and one for the mathematical problem, while the P-value for ALAD is less than

1×10^{-8}. It means that the rel index for different values of ω are almost the same for AR-ALAD, while the results are significantly different for ALAD.

In the second test, PCA is used to measure correlations between the results of distributed DC OPF obtained for the 22-bus system. Eigenvalues of the correlation matrix demonstrate how much valuable information exists on each dimension. The larger an Eigenvalue is, the more sensitive the distributed optimization will be to the corresponding parameter. To generate samples, we change ω from 100 to 200 with a step size of 10 and solve AR-ALAD and ALAD for 100 iterations for each ω. rel and the mean value of the primal feasibility criterion (i.e., $c = |t - r|$) are measured at each iteration. To consider the impact of rel and the mean value of primal feasibility simultaneously, we define $er = rel \times c$ and calculate the mean and variance of er for each ω. We create a three-axis dataset where axis one is the variance of er, axis two is the mean of er, and axis three is ω.

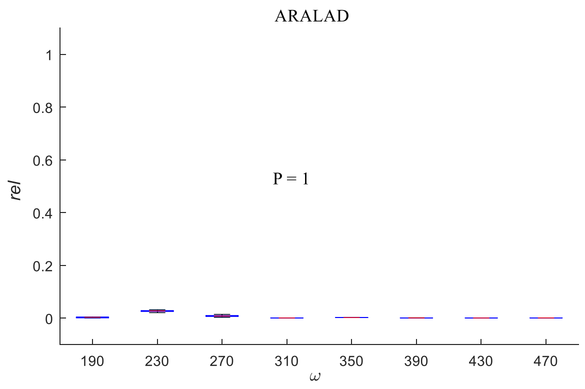
By using PCA, we measure the most irrelevant axis that has no important information. For instance, if ω-axis gets the lowest Eigenvalue, that means by changing ω in the dataset, values on the two other axes do not change significantly. As shown in Table 3.1, PCA results for er obtained by AR-ALAD demonstrates that the first Eigenvalue is near zero, which means that the impact of ω on the results is negligible. On the other hand, for ALAD, the first Eigenvalue is not small, which means that the results are sensitive to the variation of ω.

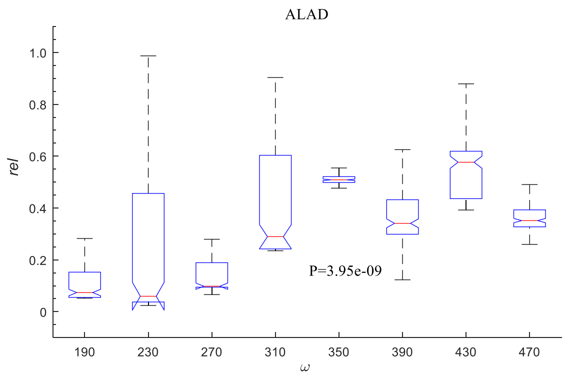




(a) (b)

Figure 3.2 F-test for OPF solved by a) AR-ALAD and b) ALAD.





(a) (b)

Figure 3.3 F-test for the mathematical problem solved by a) AR-ALAD and b) ALAD.

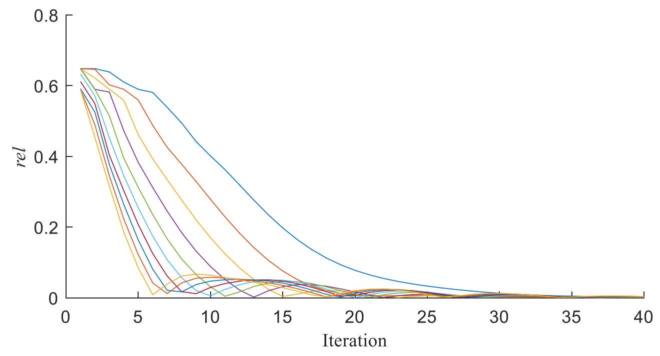
Table 3.1 Eigenvalues for AR-ALAD and ALAD obtained from PCA test

Eigenvalue	AR ALAD	ALAD
Eigenvalue1	0.0001	0.0731
Eigenvalue2	0.0174	0.4926
Eigenvalue3	2.9825	2.4343

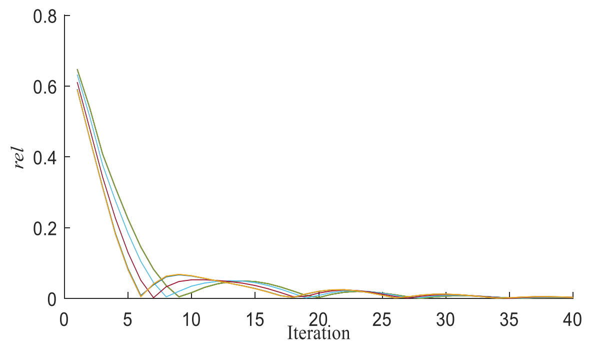
We implement another analysis to find a bound in which the rel index changes with variation of ω. Figures 3.4 and 3.5 illustrate the relative error versus iterations for various values of ω. Comparing the pointwise deviations shows that the rel values obtained from AR-ALAD vary slightly for a large range of ω, and AR-ALAD has a tighter bound than ALAD. That is, AR-ALAD is more robust to the variation of ω.

3.5.4 Acceleration

To find suitable penalty multipliers in terms of convergence speed, a naïve method is to run the algorithm multiple times and pick the best multipliers. However, this is time-consuming and inefficient. If the system condition changes, penalty multipliers need to be tuned again, especially when shared variables and cost function variables have different natures, which is the case in OPF as the cost function depends on power generations and shared variables are bus voltages. Based on our experience and analysis, the proposed algorithm not only enhances the robustness of ALAD but also outperforms in terms of convergence speed in most cases. Operators usually prefer selecting small penalty multipliers to ensure the convergence.

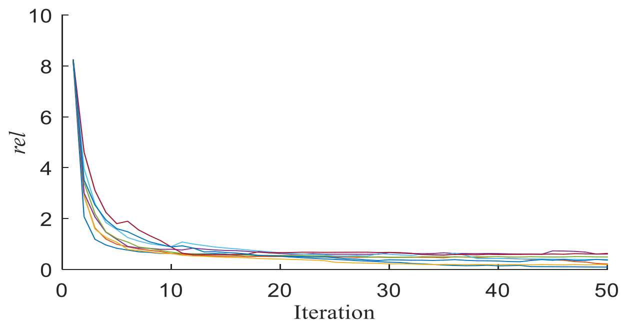


(a)

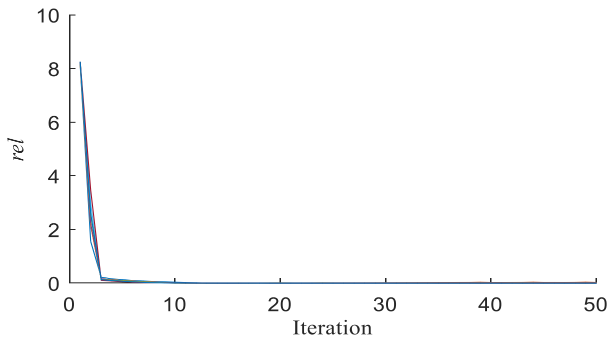


(b)

Figure 3.4 Relative error versus iteration obtained by a) ALAD, and b) AR-ALAD using various values of ω (for the OPF problem).



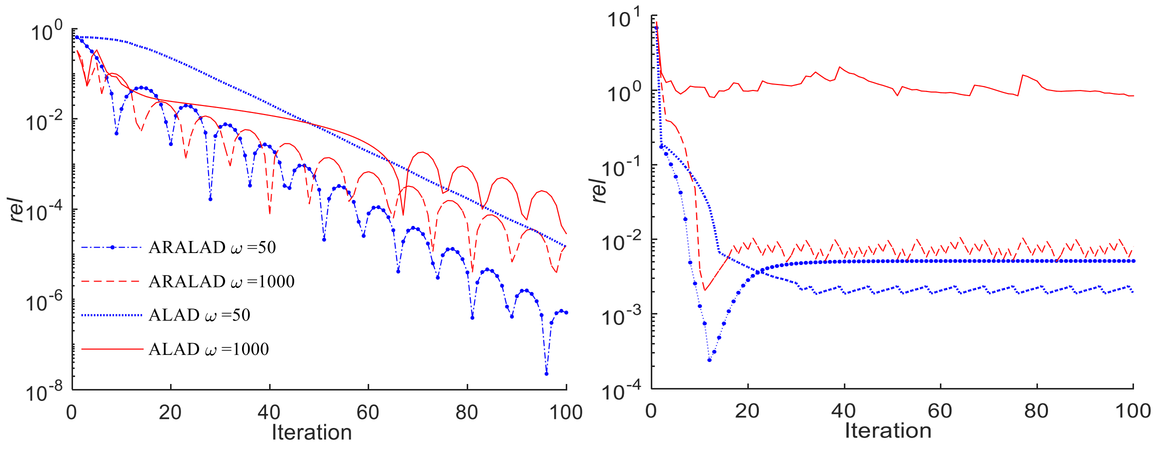
(a)



(b)

Figure 3.5 Relative error versus iteration obtained by a) ALAD, and b) AR-ALAD using various values of ω (for the mathematical benchmark problem).

In this case, the balancing coefficient α may become larger than one and helps to improve the convergence speed according to (3.26). Figure 3.6 depicts convergence processes for OPF and the mathematical benchmark problem with $\omega \in \{50{,}1000\}$. Increasing the value of ω reduces the balancing coefficient value. For OPF with $\omega = 50$ and $\omega = 1000$, for instance, the balancing coefficient α becomes 5.27 and 0.08. For the sake of comparison, 100 iterations are carried out. The rel indices obtained by AR-ALAD are less than those for ALAD. That is, the proposed AR-ALAD provides a specific rel value after fewer iterations than that of ALAD. For instance, for OPF with $\omega = 50$, while ALAD takes 100 iterations to reach $rel = 1e-5$, AR-ALAD takes 75 iterations to obtain the same rel value.

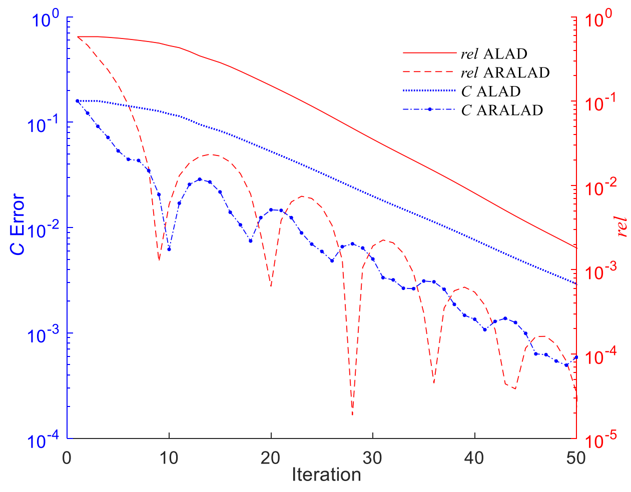


(a) (b)

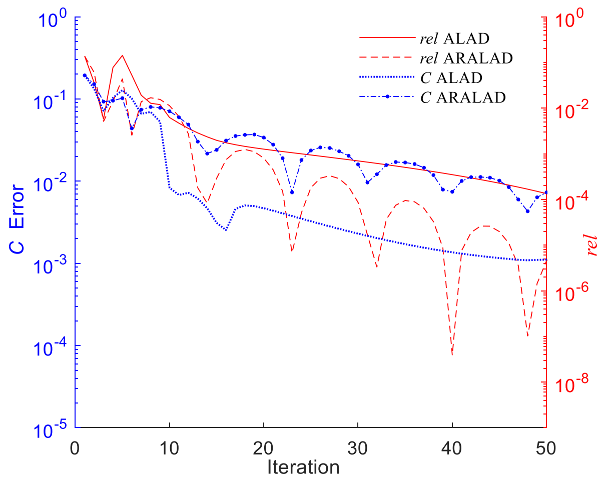
Figure 3.6 Relative error versus iteration for a) the OPF problem and b) the mathematical benchmark problem.

3.5.5 Feasibility and Optimality

The optimality (related to the objective function) and feasibility (related to consistency constraints satisfaction) are respectively measured by rel and the mean of violation of $c = t - r$. ω is set to 50 and 1000. The indices obtained by AR-ALAD and ALAD are illustrated in Figs. 3.7 and 3.8. Although for $\omega = 50$ both rel and c indices decrease gradually, their differences are less for AR-ALAD. In other words, both feasibility and optimality hold the same priority using AR-ALAD. However, for $\omega = 1000$, c vanishes fast in ALAD while rel is large upon the converges. For instance, as shown in Fig. 3.8(b), the difference between c and rel is large. The obtained solution is feasible, but not optimal as ALAD pays more attention to c rather than rel. In contrast, AR-ALAD makes a balance between the cost function and penalty terms and prevents a premature converges. Both rel and c indices decrease with almost the same pattern, and upon convergence, both feasibility and optimality are met. Comparing the results shows that using AR-ALAD does not degrade the feasibility and optimality by changing ω.

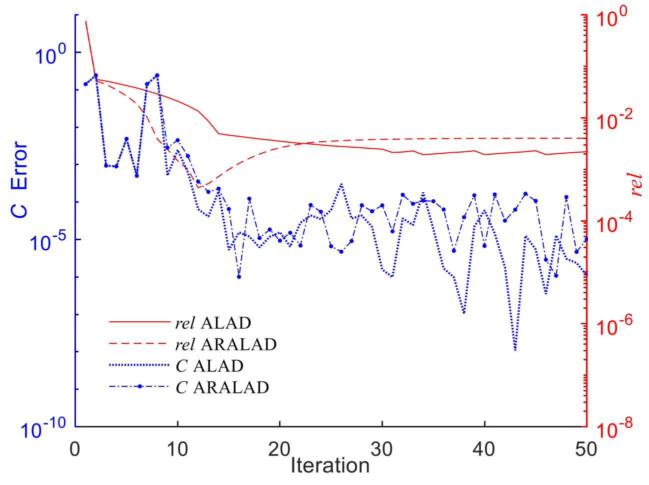


(a)

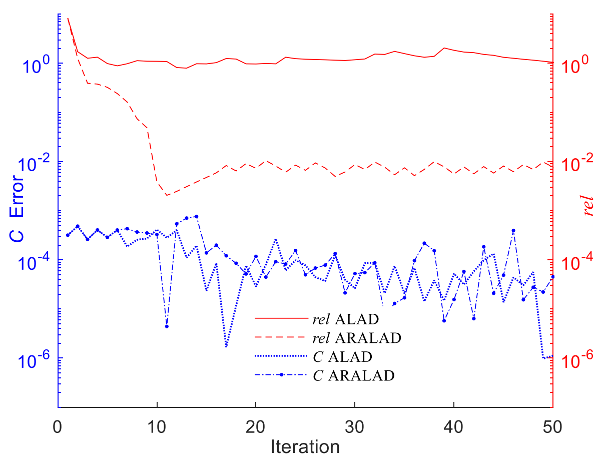


(b)

Figure 3.7 Mean of violation of c and rel versus iteration for the OPF problem: a) $\omega = 50$ and b) $\omega = 1000$.



(a)



(b)

Figure 3.8 Mean of violation of c and rel versus iteration for the mathematical benchmark problem: a) $\omega = 50$ and b) $\omega = 1000$.

3.6 Application of AR-ALAD to OPF

The proposed AR-ALAD is applied to solve DC OPF, second-order cone OPF, AC OPF, and security-constrained DC OPF. Five test systems, including a 6-bus system, a 22-bus system, the

IEEE two-area RTS-96 system with 48 buses, the IEEE 118-bus system, and a 3450-bus system, are used. All systems and equipment characteristics are given in . Simulations are carried out on MATLAB Optimization Toolbox using a 2.6GHz personal computer with 16GB of RAM. The centralized OPF is solved to provide benchmark results for comparison purposes, and the rel index is calculated to measure the difference between the distributed OPF and the classical centralized OPF.

3.6.1 DC OPF

The classical ALAD and the proposed AR-ALAD are applied to solve DC OPF. The initial values are set as $t^0 = r^0 = 0$, $\lambda^0 = 50$, and $\omega^0 = 50$. These penalty multipliers are neither small to slow the convergence speed nor large to result in converging to a suboptimal point or divergence. Hence, we can perform a fair comparison between ATC and AR-ATC. The stopping threshold is $\epsilon =$1e-4. Figure 3.9 shows the relative errors over the course of iterations. For all test systems, AR-ALAD outperforms in terms of the convergence speed and relative error. To reach a specific relative error, ALAD takes more iterations than AR-ALAD. ALAD takes eight (less than one second), 67 (more than three seconds), 205 (nine seconds), and 647 (128 seconds) iterations to converge to an error of 3e-11, 8e-4, 2e-4, and 3e-4 respectively for the 6-bus, 22-bus, 48-bus, and 118-bus systems. In contrast, AR-ALAD takes six (less than one second), 51 (less than three seconds), 21 (two seconds), and 35 (13 seconds) iterations to converge to a relative error of 4e-16, 3e-6, 1e-4, and 8e-3 respectively for the 6-bus, 22-bus, 48-bus, and 118-bus systems.

Large System: We have also studied a very large case with 3450 buses. The shared variables are initialized to zero. The simulations are carried out for 40 iterations with a small, a medium, and a large ω^0. The rel indices are shown in Fig. 3.10. The proposed AR-ALAD yields better results than those of ALAD for all studied cases.

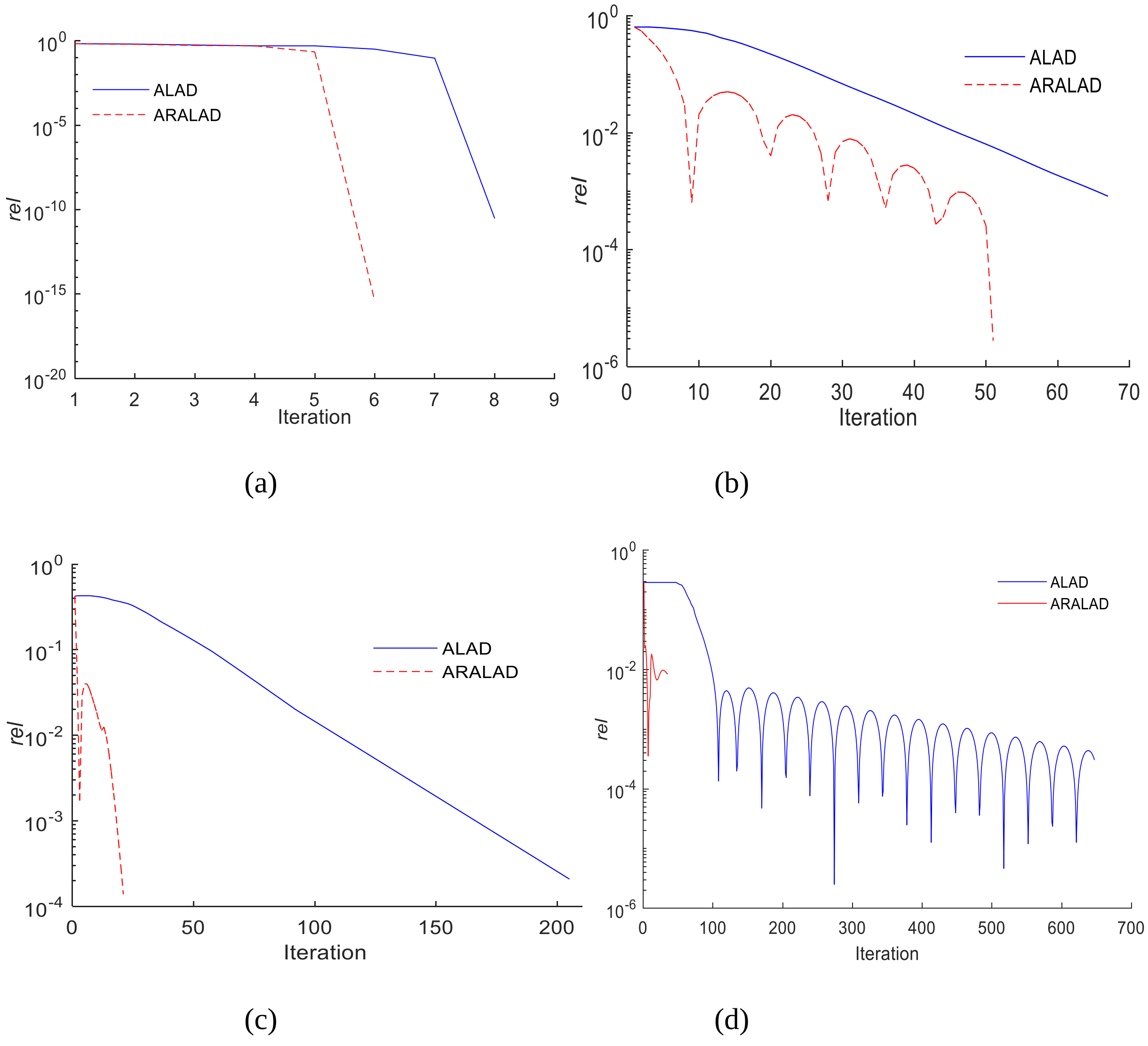


Figure 3.9 The rel index obtained by ALAD and AR-ALAD for a) the 6-bus, b) 22-bus, c) 48-bus, and d) IEEE 118-bus test systems.

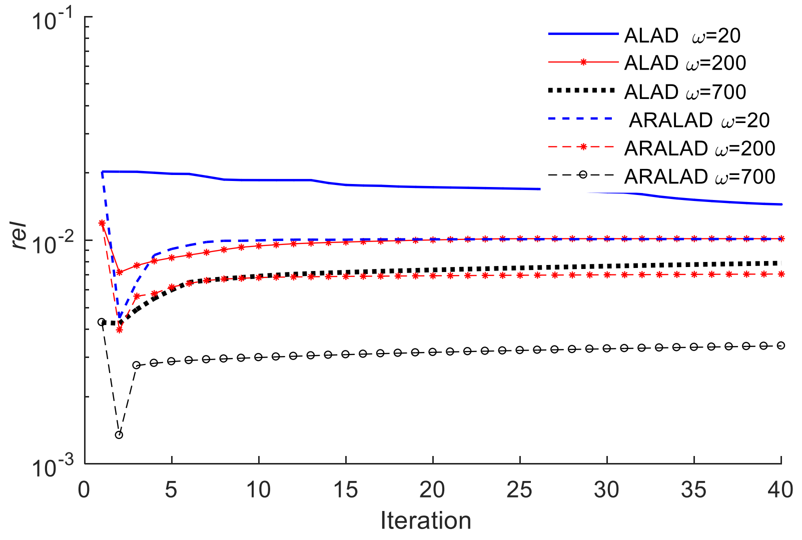


Figure 3.10 rel versus iterations for the 3450-bus system.

Sensitivity Analysis: The 22-bus system is selected to perform a sensitivity analysis with respect to the variation of ω. As shown in Fig. 3.11, ALAD takes fewer iterations with increasing ω; however, the relative error increases. Although by setting $\omega > 1000$ primal feasibility is within an acceptable threshold, ALAD converges to unacceptable points with relative errors larger than 100%. In contrast, AR-ALAD shows a stable convergence. The number of iterations and rel are almost constant. We conclude that selecting a small or moderate value for ω (e.g., $10 < \omega < 100$) and applying AR-ALAD results in obtaining a good convergence performance from the perspective of the number of iterations and relative error.

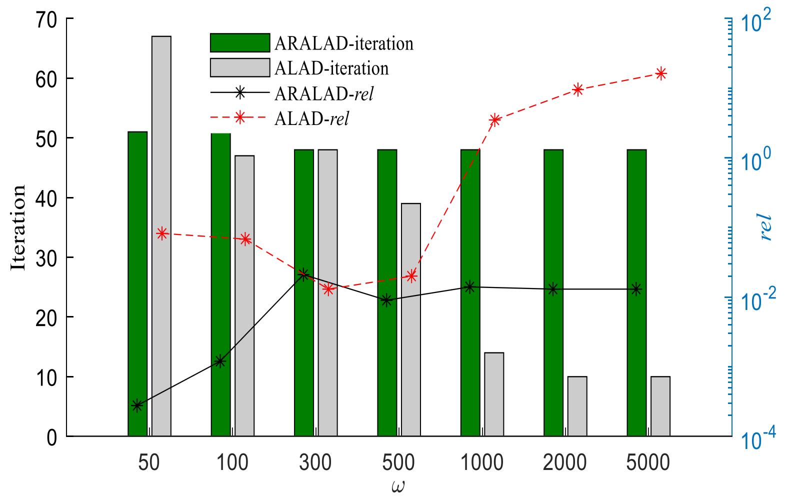


Figure 3.11 Number of iterations and relative error versus ω for the 22-bus system.

3.6.2 AC OPF

ALAD and AR-ALAD are carried out to solve a nonconvex AC OPF for the 22-bus test system. No relaxation, convexification, or linearization is implemented in this case. The initial values of shared variables (i.e., voltage magnitudes and angles of border buses) are set to zero, and four different initial values for Lagrange and penalty multipliers are selected as $\omega^0 = \lambda^0 = \{20, 200, 2000, 20000\}$. This wide range is selected to analyze the performance of AR-ALAD for small, medium, and large values of ω^0. Both ALAD and AR-ALAD based distributed AC OPF algorithms are run for 100 iterations. Although the considered problem is AC OPF that is nonlinear and nonconvex, the proposed algorithm still works for this problem. Figure 3.12 depicts the acceleration and robustness features of AR-ALAD. For the four values of penalty multipliers, AR-ALAD provides more accurate results faster than ALAD. While ALAD does not provide good results for $\omega^0 = \{20,20000\}$, AR-ALAD enhances the convergence performance significantly even for these two values of ω^0.

3.6.3 Security-Constrained DC OPF

The 48-bus system is selected, and line outages are considered. The initial values of shared variables are set to zero and $\omega^0 = \lambda^0 = 50$. The rel indices over 50 iterations are plotted in Fig. 3.13. The rel obtained by AR-ALAD is always less than that of ALAD.

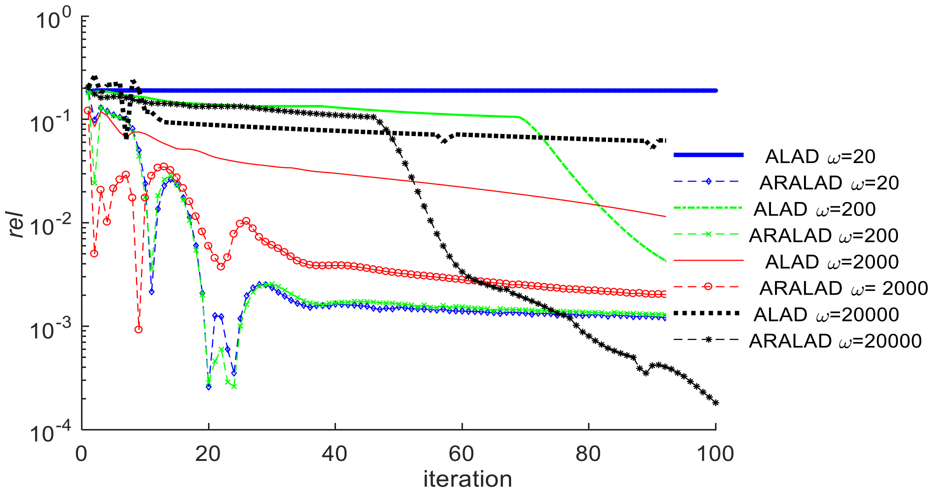


Figure 3.12 The rel index obtained by ALAD and AR-ALAD for AC OPF.

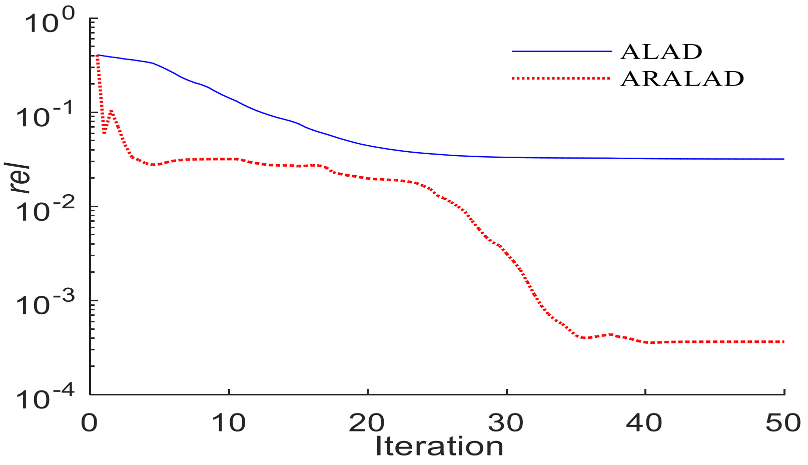


Figure 3.13 rel for security-constrained DC OPF for the 48-bus system.

While ALAD has a slow convergence, AR-ALAD reduces the error considerably at the first couple of iterations. The rel and mismatches of the consistency constraints after 50 iterations of ALAD are 3e-2 and 0.16, and they are, respectively, 3e-4 and 7e-3 for AR-ALAD.

3.7 Application of Proposed Method to Other Coordination Algorithms

To demonstrate the applicability of the proposed method on other coordination algorithms, we follow a similar procedure as of AR-ALAD to modify ATC with exponential penalty functions (EPF), APP, and Nesterov-based ADMM.

3.7.1 AR-EPF to Solve OPF

We refer to (DorMohammadi and Rais-Rohani 2013) for more details on EPF. The modified EPF with the balancing coefficient α is called AR-EPF. The initial values are set as $t^0 = r^0 = 0$, $\lambda^0 = 50$, and $\omega^0 = 50$. The stopping threshold is set to 1e-5. Figure 3.14 shows the convergence measure rel and the primal feasibility measure over iterations for 6-bus, 22-bus, 48-bus, and 118-bus systems. For all cases, AR-EPF shows much better performance as compared to EPF. While EPF takes 14 (more than one second), 106 (ten seconds), 52 (six seconds), 789 (421 seconds) iterations to converge respectively for the 6-bus, 22-bus, 48-bus, and 118-bus systems, AR-EPF converges after 6 (one second), 68 (8 seconds), 25 (5 seconds), and 550 (295 seconds) iterations for the same systems. The rel and feasibility measures obtained by AR-EPF are always below those indices obtained by EFP.

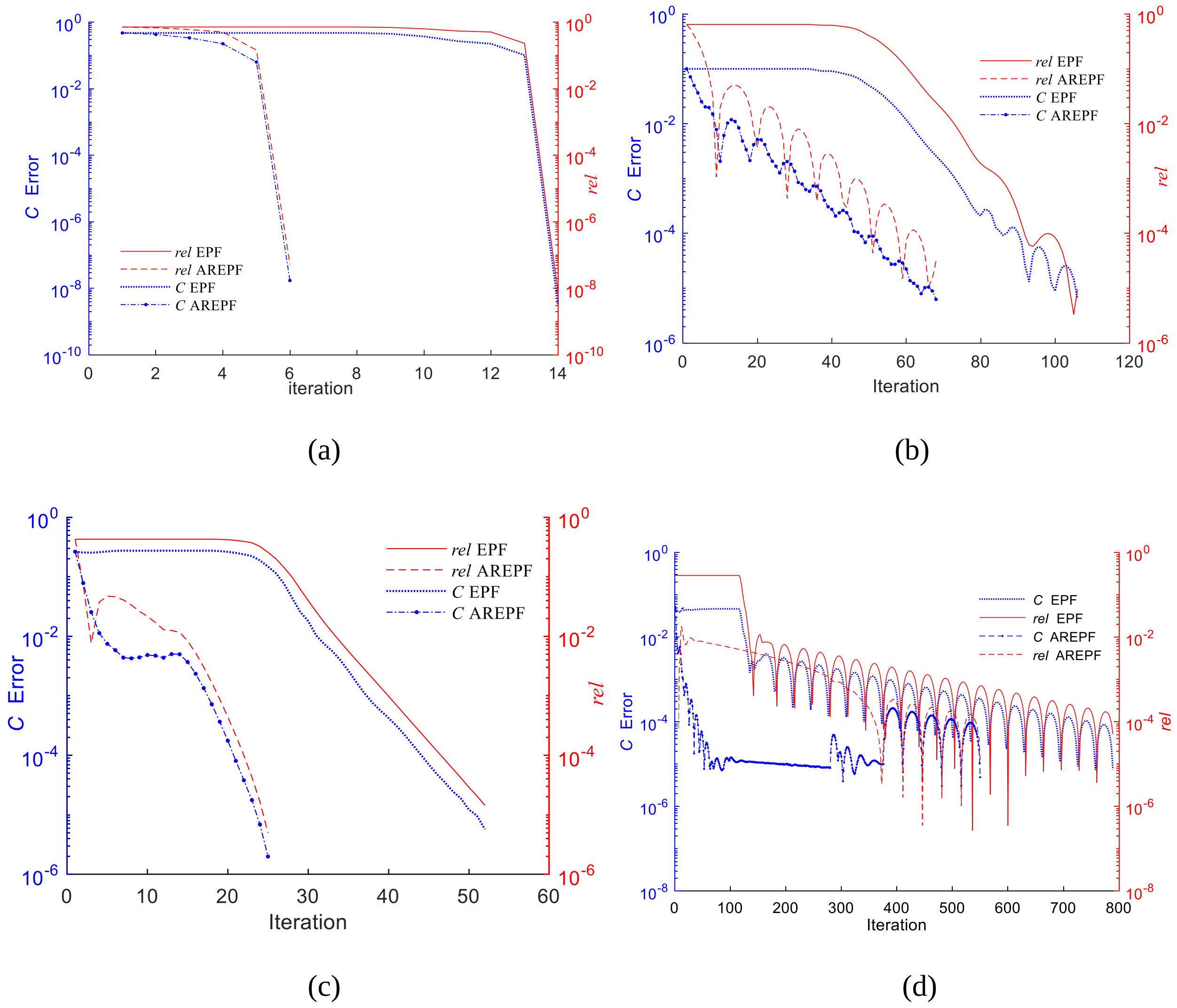
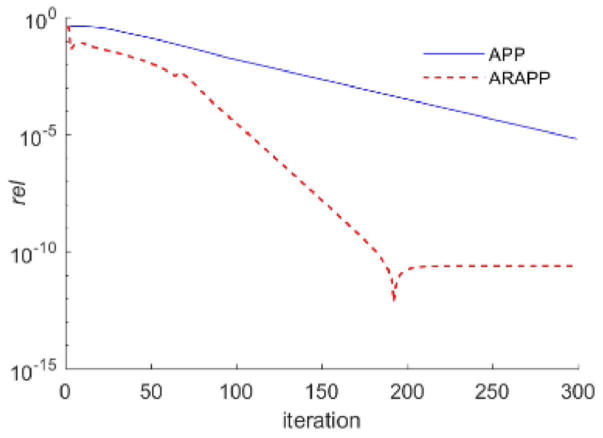
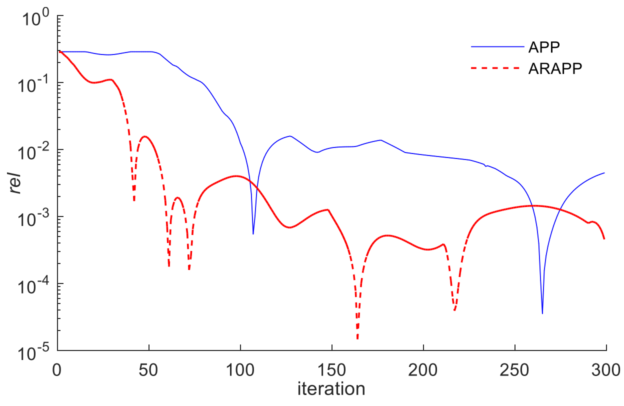


Figure 3.14 The rel and primal feasibility indices obtained by ALAD and AR-ALAD for a) 6-bus, b) 22-bus, c) 48-bus, and d) IEEE 118-bus test systems.

3.7.2 AR-APP

Distributed OPF with APP is presented in (Baldick, Kim et al. 1999). The modified algorithm with balancing coefficient α is called AR-APP. 300 iterations of the classical APP and AR-APP are carried out to solve DC OPF for the 48-bus and IEEE 118-bus systems. Figure 3.15 demonstrates that the rel index obtained by AR-APP is less than that of the classical APP.





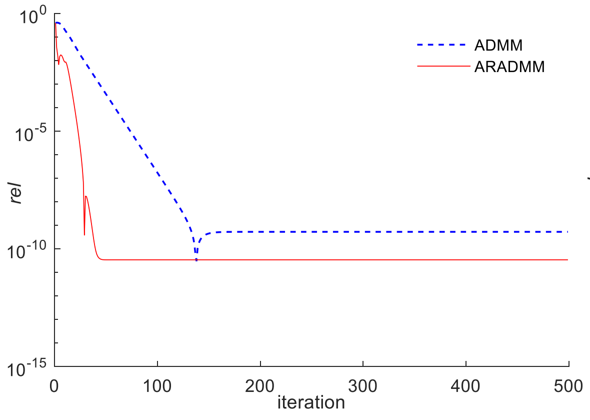
(a) (b)

Figure 3.15 The rel index obtained by APP and ARAPP for a) 48-bus system and b) IEEE 118-bus system.

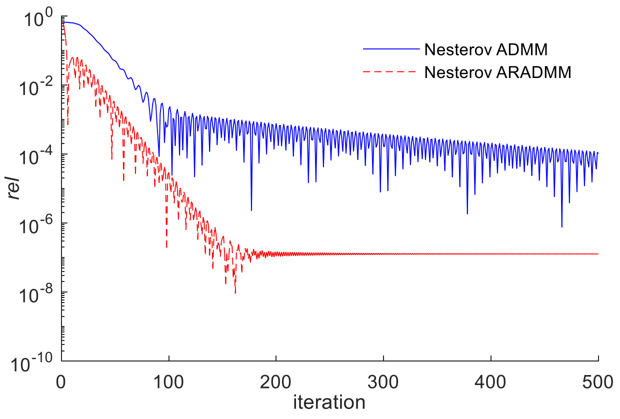
3.7.3 ADMM and Nesterov-based ADMM

The balancing coefficient α is incorporated in ADMM and Nesterov-based ADMM (Goldstein, O'Donoghue et al. 2014). The modified algorithms are named AR-ADMM and AR-Nesterov-based ADMM. ADMM and AR-ADMM are used to solve DC OPF for the 48-bus system. Multipliers ω^0 and λ^0 are set to 100. Fig. 3.16 (a) shows the rel index. While ADMM reaches a rel of 4e-4 after 50 iterations, this index is 3e-11 for AR-ADMM. We run both algorithms for 500 iterations. The rel indices obtained by ADMM and AR-ADMM are 5e-10 and 33-11, respectively.

Nesterov-based ADMM and AR-Nesterov-based ADMM are used to solve DC OPF for the 22-bus system. We set multipliers ω^0 and λ^0 to 30 and terminate the algorithms after 500 iterations. The value of coefficient α is calculated to be 14.97. Figure 3.16 (b) depicts that the rel index for AR-Nesterov-based ADMM is less than that for the classical Nesterov-based ADMM. At iteration 500, AR-Nesterov-based ADMM provides a rel equal to 1e-7 while it is 1e-4 for the classical Nesterov-based ADMM. It shows that the proposed approach can enhance the performance of algorithms with linear (e.g., ATC and APP) and superlinear convergence rates (Nesterov-based ADMM).



(a)



(b)

Figure 3.16 rel versus iteration for a) the 48-bus system and b) the 22-bus system.

3.8 Conclusion

This chapter presents an accelerated, robust ATC and its application for solving OPF in a distributed manner. The proposed algorithm enhances the convergence performance of ATC and decreases its sensitivity to initial values of penalty multipliers and imbalance of cost terms in optimization objective functions by incorporating a balancing coefficient in objective functions of subproblems. The simulations show that the proposed AR-ATC with ALAD coordination strategies provides promising results for DC OPF, AC OPF, and security-constrained DC OPF. For cases with small ω, AR-ATC reduces the number of iterations and the relative error, and for cases with large ω, AR-ATC prevents converging to a suboptimal point or divergence. The best performance of AR-ATC is obtained when multipliers were set to moderate values. Although this is case dependent, the proposed algorithm works more effectively when multipliers are initialized in the range of $10 \sim 50$. The proposed algorithm is applied to ATC with EPF coordination strategy, APP, and Nesterov-based ADMM. The results showed that AR-EPF, AR-APP, and AR-Nesterov-based ADMM outperform the classical forms.

Some future research directions are developing more sophisticated approaches to find an optimal balancing coefficient, using different balancing coefficient for different subproblems, updating the balancing coefficient over iterations, and using historical information and learning approaches to find a good balancing coefficient.

CHAPTER 4. MOMENTUM EXTRAPOLATION PREDICTION-BASED ASYNCHRONOUS DISTRIBUTED OPTIMIZATION FOR POWER SYSTEMS

4.1 Introduction

Iterative distributed optimization algorithms usually need synchronization of subproblems at each iteration. An iteration index is defined, and subproblems are solved once at each iteration. This degrades the scalability of distributed optimization and under-utilization of computation resources, particularly if subproblems are heterogeneous. To address these limitations, this chapter proposes a prediction-correction based asynchronous alternating direction method of multipliers (A-ADMM). At the end of each iteration, a subproblem no longer needs to wait for the most updated information of its neighbors. A momentum-based extrapolation method is developed to predict missing information corresponding to shared variables values. A correction step is designed using momentum values to prevent the predicted values to become far from the possible solution and hence avoid divergence. These predictions are integrated into distributed optimization to allow subproblems to be solved continuously with no need for synchronization at each iteration. The proposed A-ADMM significantly reduces the unproductive time and the under-utilization of computation resources if subproblems are computationally heterogeneous and enhance the solution speed even if subproblems are homogeneous. The proposed A-ADMM is applied to solve the optimal power flow problem. Numerical results on various cases show the promising performance of A-ADMM.

4.2 Contribution

The main contributions of this chapter are summarized as follows:

- An asynchronous ADMM is developed to solve OPF. The proposed A-ADMM, which uses the concept of prediction and correction, is computationally much more efficient than the classical synchronous parallel ADMM.

- A momentum-based extrapolation, which is inspired by the Nesterov technique for gradient descent (Nesterov 1983), is presented to predict the values of missing information while ensuring that the predictions are bounded and hence preventing A-ADMM divergence even if missing information is observed in several consecutive iterations.

- The prediction step of the proposed A-ADMM does not need a large dataset and works well, with negligible computation cost, from the first few iterations.

4.3 Distributed OPF with Synchronous ADMM

ADMM can be applied to solve various power system optimization problems. We explain the classical and the proposed ADMM algorithms in the context of OPF.

4.3.1 Classical Sequential ADMM

A typical OPF problem can be expressed by (4.1a)-(4.1i) (Zimmerman, Murillo-Sánchez et al. 2011). The objective function is to minimize generation costs. Nodal power balance constraints are given by (4.1b) and (4.1c). Constraints (4.1d) and (4.1e) enforce line flow limits at two ending terminals of a line. The upper and lower bounds of generating units are imposed by (4.1f) and (4.1g). Inequalities (4.1h) and (4.1i) are bus voltage magnitude and angle limits.

$$\min f(p) = \sum_{u} a_u p_u^2 + b_u P_u + c_u \qquad (4.1a)$$

$s.t.$

$$h_p(V, \theta, p_g) = P_{bus}(V, \theta\ ) + P_d - p_g = 0 \tag{4.1b}$$

$$h_q(V, \theta, q_g) = Q_{bus}(V, \theta) + Q_d - q_g = 0 \tag{4.1c}$$

$$g_{ls}(V, \theta) = |F_{ls}(V, \theta)| - F_{max} \leq 0 \tag{4.1d}$$

$$g_{lr}(V, \theta) = |F_{lr}(V, \theta)| - F_{max} \leq 0 \tag{4.1e}$$

$$p_u^{min} \leq p_u \leq p_u^{max} \qquad \forall u \tag{4.1f}$$

$$q_u^{min} \leq q_u \leq q_u^{max} \qquad \forall u \tag{4.1g}$$

$$V_i^{min} \leq V_i \leq V_i^{max} \qquad \forall i \tag{4.1h}$$

$$\theta_i^{ref} \ \leq \theta_i \leq \theta_i^{ref} \qquad \forall i \tag{4.1i}$$

where u is the index for generating units, V_i and θ_i voltage magnitude and angle of bus i, p_g and q_g are active and reactive power generation, and P_d and Q_d are active and reactive power demand. Although ADMM works for systems with multiple areas, for the sake of explanation, we consider that the system includes two connected areas i and j, and (4.1) is decomposed into two coupled OPF subproblems, each pertaining to an area. The OPF subproblem of each area is to minimize local generation costs subject to its local equality and inequality constraints (Kargarian, Mohammadi et al. 2018). Consider that $x_i = \{p_i, q_i, V_i, \theta_i\}$ and $x_j = \{p_j, q_j, V_j, \theta_j\}c$ denote local variables of areas i and j, and y are shared variables appearing in both OPF subproblems. The shared variables vary depending on the considered OPF model, AC or DC, and the way that power flow in tie-line connecting neighboring areas are modeled. We refer to (Kim and Baldick 2000, D. K. Molzahn, F. Dorfler et al. 2017, Kargarian, Mohammadi et al. 2018) for more details.

$$y \in \{p_{ij}, q_{ij}, V_{ij}, \theta_{ij}\} \tag{4.2}$$

where p_{ij} and q_{ij} refer to tie-line flows, and V_{ij} and θ_{ij} denote the voltage magnitude and angle of ending terminal of tie-lines. In the context of ADMM, shared variables are duplicated to create two sets of variables y_i and y_j, consistency constraint $y_i - y_j = 0$ is added in each subproblem (SP), and this constraint is then relaxed in the objective functions of SP_i and SP_j using a quadratic augmented Lagrangian function (Boyd, Parikh et al. 2011, D. K. Molzahn, F. Dorfler et al. 2017, Kargarian, Mohammadi et al. 2018).

$$L_i(x_i, y_i, \lambda) = \min_{x_i, y_i} \sum_{u \in \Omega_i} a_u p_u^2 + b_u P_u + c_u + \lambda(y_i - y_j) + \frac{\rho}{2} \|y_i - y_j\|^2 \tag{4.3}$$

$$L_j(x_\mathrm{j}, y_j, \lambda) = \min_{x_j, y_j} \sum_{u \in \Omega_j} a_u p_u^2 + b_u P_u + c_u + \lambda(y_j - y_j) + \frac{\rho}{2} \|y_j - y_j\|^2 \tag{4.4}$$

where Ω_i and Ω_j denote the set of generating units in areas i and j, respectively. The classical ADMM, which is a sequential algorithm, solves the distributed OPF problem iteratively as follows (Boyd, Parikh et al. 2011):

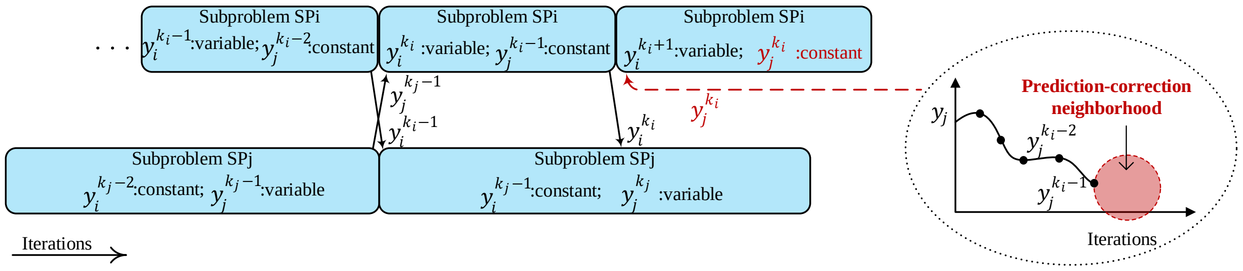
$$x_i^{k+1}, y_i^{k+1} = \operatorname{argmin} L_i(x_i, y_i, y_j^k, \lambda^k) \tag{4.5}$$

$$x_j^{k+1}, y_j^{k+1} = \operatorname{argmin} L_j(x_j, y_i^{k+1}, y_j, \lambda^k) \tag{4.6}$$

$$\lambda^{k+1} = \lambda^k + \rho\left(y_i^{k+1} - y_j^{k+1}\right) \tag{4.7}$$

where $k + 1$ is the current iteration. The problem is solved to find y_i^{k+1} and then this value is used to solve for y_j^{k+1}. Two criteria are checked for convergence.

$$\text{primal residual: } \mathscr{s}^{k+1} = \left|y_i^{k+1} - y_j^{k+1}\right| < \epsilon \tag{4.8}$$



Subproblem SPi
$y_i^{k_i-1}$:variable; $y_j^{k_i-2}$:constant
Subproblem SPi
$y_i^{k_i}$:variable; $y_j^{k_i-1}$:constant
Subproblem SPi
$y_i^{k_i+1}$:variable; $y_j^{k_i}$:constant
$y_j^{k_j-1}$
$y_i^{k_i-1}$
$y_i^{k_i}$
$y_j^{k_i}$
Subproblem SPj
$y_i^{k_j-2}$:constant; $y_j^{k_j-1}$:variable
Subproblem SPj
$y_i^{k_j-1}$:constant; $y_j^{k_j}$:variable
Iterations
Prediction-correction neighborhood
y_j
$y_j^{k_i-2}$
$y_j^{k_i-1}$
Iterations

Figure 4.1 Proposed prediction-correction based asynchronous ADMM over the course of iterations.

$$\text{dual residual: } d^{k+1} = \left|y_j^{k+1} - y_j^k\right| < \epsilon \tag{4.9}$$

Although ADMM is proven to work for convex problems, (Erseghe 2015) has reported its applicability for both DC OPF and non-convex AC OPF problems.

4.3.2 Synchronous Parallel ADMM

ADMM can be transformed into a parallel algorithm using the technique presented in (Wang, Wu et al. 2016, Mehrtash, Kargarian et al. 2019).

$$x_i^{k+1}, y_i^{k+1} = \operatorname{argmin} \sum_{u\in\Omega_i} a_u p_u^2 + b_u P_u + c_u + \lambda\left(y_i^{k+1} - \frac{y_i^k + y_j^k}{2}\right) + \frac{\rho}{2}\left\|y_i^{k+1} - \frac{y_i^k + y_j^k}{2}\right\|^2 \tag{4.10}$$

$$x_j^{k+1}, y_j^{k+1} = \operatorname{argmin} \sum_{u\in\Omega_j} a_u p_u^2 + b_u P_u + c_u + \lambda\left(\frac{y_i^k + y_j^k}{2} - y_j^{k+1}\right) + \frac{\rho}{2}\left\|y_j^{k+1} - \frac{y_i^k + y_j^k}{2}\right\|^2 \tag{4.11}$$

$$\lambda^{k+1} = \lambda^k + \rho\left(y_i^{k+1} - y_j^{k+1}\right) \tag{4.12}$$

Subproblems (4.10) and (4.11) are solved in parallel. This reduces the solution time significantly as compared to that of the classical sequential ADMM. As shown in Fig. 4.2.b, the idle time is reduced by replacing the classical sequential ADMM with the parallel synchronous

ADMM. In a certain time interval, more iterations of the parallel ADMM can be carried out, and the obtained solution might be better.

4.4 Proposed Prediction-Correction Based Asynchronous ADMM

Although synchronous parallel ADMM reduces the computational time, this algorithm still needs a perfect level of synchrony among subproblems and all subproblems must be solved once at each iteration. This results in possible long idle time, particularly if subproblems are heterogeneous with different computational burdens. Consider two subproblems and that T_i^k and T_j^k are, respectively, the solution times of subproblems i and j at iteration k. If $T_i^k \gg T_j^k$ or $T_i^k \ll T_j^k$, a long idle time will be observed at each iteration of the synchronous ADMM. This results in under-utilization of computation resources and increasing the overall solution time, particularly for large problems.

To allow subproblems to be solved as many times as possible in a specific time interval and reduce the computational time of distributed optimization, we propose an asynchronous ADMM (A-ADMM). This algorithm eliminates the need for synchrony between subproblems and the unproductive time of the synchronous ADMM, as illustrated in Fig. 4.2.c. Each subproblem keeps updating its local and shared variables regardless of whether it has received any information from its neighboring subproblems or not. If a subproblem has not received information from its neighbors, we propose to predict the missing information and then solve the corresponding OPF subproblem.

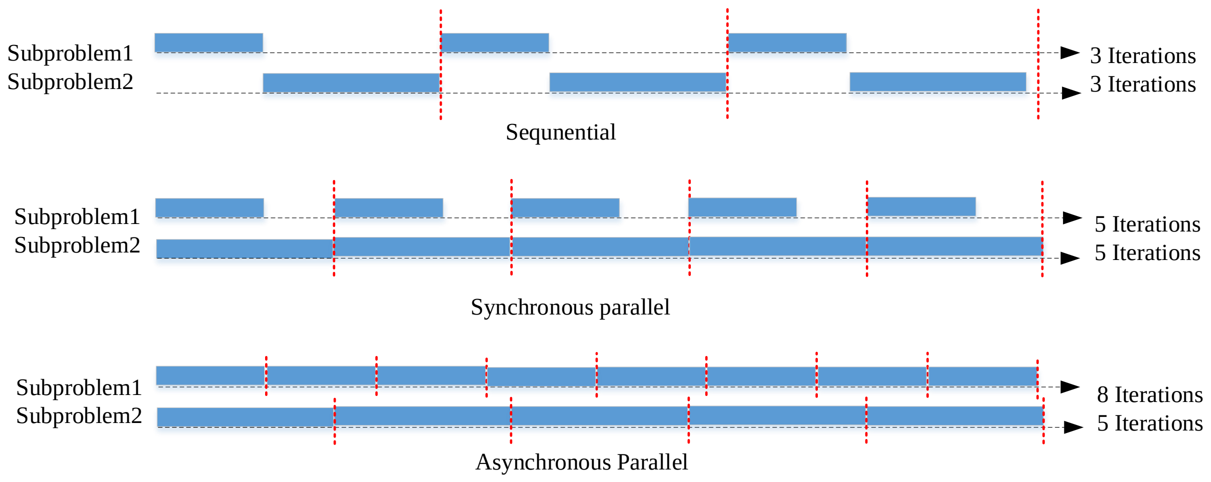


Figure 4.2 The idle time for a) the classical sequential ADMM, b) parallel synchronous ADMM, and c) the proposed parallel asynchronous ADMM.

4.4.1 Prediction-Correction Neighborhood

The procedure of the proposed A-ADMM is illustrated in Fig. 4.2. A series of values is obtained for shared variables over the course of iterations. At the end of each iteration k, if subproblem SP_i, for instance, has not received y_j^k from its neighbor, the data series can be used to predict this missing information. SP_i is then solved at iteration $k+1$ using the predicted y_j^k.

Classical linear extrapolation can be used to project the values of shared variables at the current iteration using the values of shared variables obtained in two previous iterations. Consider Fig. 4.2 where SP_i has not received y_j^k from SP_j. Without this value, SP_i cannot be solved at iteration $k+1$. The linear extrapolation predicts the missing information, i.e., y_j^k, required by SP_i as:

$$\hat{y}_j^k = \left(y_j^{k-1} - y_j^{k-2}\right)k + 2y_j^{k-2} - y_j^{k-1} \tag{4.13}$$

where y_j^{k-1} and y_j^{k-2} are the shared variable values received from SP_j at iterations $k-1$ and $k-2$, respectively. $\hat{y}_j^k$ is the predicted value for y_j at iteration k.

Guideline for Using Classical Linear Extrapolation: Our observations show that this method works well for predicting missing information for up to one or two iterations if the distributed algorithm's solution is far from the optimal results. If the solution is close to the optimal results, variations of shared variables are not significant, as shown in Fig. 4.3, and the classical linear extrapolation can be used to predict missing information for up to four or five consecutive iterations. Hence, we conclude that if the computational times of subproblems are considerably different, which leads to having missing information for several successive iterations, or if variations of shared variables are significant (the oscillatory region in Fig. 4.3 with several peaks and valleys), the classical linear extrapolation might not provide a good enough prediction for $\hat{y}_i$.

Momentum-Extrapolation based Prediction: Inspired by the Nesterov technique for accelerating gradient descent (Nesterov 1983), we propose a momentum-based extrapolation predictor for distributed optimization if computational times of subproblems are considerably different or variations of shared variables are significant. The proposed momentum-based extrapolation is a *prediction-correction* approach that uses the accumulation of predictions and corrections. The momentum term, which is the slope trend of shared variables' values from iterations one to k, adds a correction step to extrapolation and ensures that the predictions will be bounded even if missing information is observed in many consecutive iterations of ADMM. This is proven in the Appendix. By adding the momentum, predictions become more conservative as more iterations are carried out with missing information. As shown in Fig. 4.3, if a subproblem observes several iterations with missing information, the predicted values will become saturated to prevent large prediction errors while allowing a subproblem to be solved continuously. In the subsequent section, we will explain the detailed formulations of the momentum-based extrapolation in the context of asynchronous ADMM.

4.4.2 Momentum-Based Extrapolation to Formulate SP_i

We formulate a subproblem, SP_i, using the momentum-based extrapolation and explain the ADMM solution procedure over iterations. Due to the difference of subproblems' computational time, the number of iterations that SP_i is solved in a certain time span might be different than that of other subproblems. Hence, we assign an iteration index to each subproblem. k_i is the iteration index for SP_i. At iteration $k_i + 1$, the OPF subproblem SP_i is as follows:

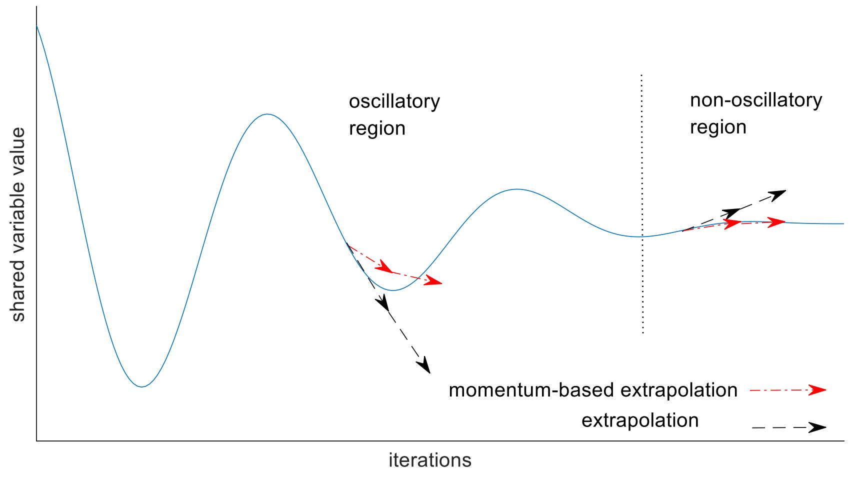


Figure 4.3 Extrapolation vs. momentum-based extrapolation predictions.

$$x_i^{k_i+1}, y_i^{k_i+1} = \operatorname{argmin} \sum_{u \in \Omega_i} a_u p_u^2 + b_u P_u + c_u + \lambda \left(y_i^{k_i+1} - \frac{y_i^{k_i} + \hat{y}_j^{k_i}}{2} \right) + \frac{\rho}{2} \left\| y_i^{k_i+1} - \frac{y_i^{k_i} + \hat{y}_j^{k_i}}{2} \right\|^2 \quad (4.14)$$

where $\hat{y}_j^{k_i}$ is the predicted value for y_j if missing information is observed at the beginning of iteration $k_i + 1$. A flag, fl_i, is introduced for each subproblem SP_i to determine whether SP_i should use the actual latest values of shared variable y_j or the predicted values. When SP_i receives information from SP_j, the flag becomes 1; otherwise, 0. At the beginning of each iteration $k_i + 1$, if the flag is one, the actual values of shared variables are used in (4.14) and the flag is set to 0; otherwise, the momentum-based extrapolation is implemented to predict $\hat{y}_j^{k_i}$. Assume that, for instance, flags corresponding to SP_i and SP_j are respectively 0 and 1, SP_i is solved with the predicted $\hat{y}_j$ and SP_j is solved with the actual y_i. At the beginning of $k_i + 1$ of SP_i, the momentum-based extrapolation is implemented to predict $\hat{y}_j$ as follows:

$$v_j^{k_i-1} = \mu v_j^{k_i-2} + \eta\left(\hat{y}_j^{k_i-1} - \hat{y}_j^{k_i-2}\right) \tag{4.15}$$

$$\hat{y}_j^{k_i} = \hat{y}_j^{k_i-1} + v_j^{k_i-1} \tag{4.16}$$

where $v_j^{k_i-1}$ is the ***momentum*** term corresponding to shared variable values received from SP_j from iteration one to iteration $k_i - 1$ (v_j^0 is zero). Parameters η and μ are weight coefficients for extrapolation and momentum terms, respectively. A large η may lead to increasing the oscillation of predicted values of $\hat{y}_j$ over iterations. The weight coefficient μ, which acts as a long-term damper, can make the prediction smoother and prevents overshoots and undershoots. Although μ can mitigate fluctuations, it may not function effectively if η is set to a large value. Parameters μ and η should be selected in a reasonable range. This is problem-dependent. If the algorithm is sensitive to prediction errors, we suggest selecting larger μ and smaller η. On the other hand, if the prediction errors within a bound do not affect distributed optimization significantly, we suggest setting a reasonably large extrapolation coefficient. Indeed, with changing the weight coefficients μ and η, the level of conservativeness in predictions can be controlled. For instance, for a given problem, if a user wants to be too conservative, the coefficients can be set in a manner for A-ADMM to use the latest received y_j always as the predicted values. It should be noted that although a conservative prediction might prevent A-ADMM to have a good jump towards the optimal solution, it still allows the algorithm to update the Lagrange multipliers that results in a small or medium jump towards the optimal solution after each iteration. We have discussed details of momentum and extrapolation parameters in the Appendix. Moreover, we have proved that if μ and η are selected properly, the predictions are always within a bound of the most updated true information. Figure 4.4 illustrates how the momentum-based extrapolation predicts $\hat{y}_j^{k_i}$ using the data series obtained at previous iterations. The solid green and red arrows show, respectively, the momentum and extrapolation terms at iteration k_i, while dashed green and red arrows reflect these terms after incorporating their corresponding weight factors. The solid blue line represents the prediction of missing information.

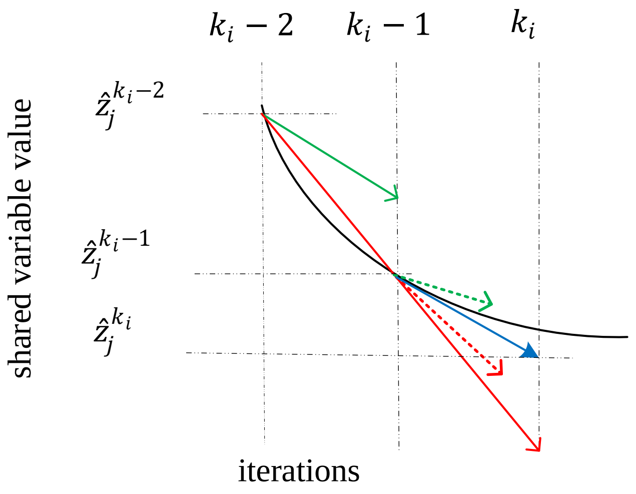


Figure 4.4 Momentum-based extrapolation prediction at iteration k_i.

If $fl_i^{k_i-1}$ is zero, predicted values are used to solve SP_i at iteration k_i. It means that SP_i moves one step forward while SP_j is still working to find its optimal solution. Assume that SP_j is solved, and $fl_i^{k_i}$ becomes one. Since SP_j was one step before SP_i, using the most updated values to solve SP_i at iteration $k_i + 1$ may not be efficient. We suggest estimating the values of shared variables received from SP_j to mimic the situation in which both OPF subproblems were at the same step. This would help SP_i to move towards the optimal solution faster. Therefore, to solve SP_i at each iteration $k_i + 1$, even if $fl_i^{k_i} = 1$, we use the momentum-based extrapolation to mimic the behavior of SP_j.

$$v_j^{k_i-1} = \mu v_j^{k_i-2} + \eta \left(y_j^{(k_i-1)^+} - \hat{y}_j^{k_i-2} \right) \tag{4.17}$$

$$\hat{y}_j^{k_i} = y_j^{(k_i-1)^+} + v_j^{k_i-1} \tag{4.18}$$

where $y_j^{(k_i-1)^+}$ denotes the values of shared variables received from SP_j between iterations $k_i - 1$ and k_i.

4.4.3 Discussions on Motivation Behind Momentum-Extrapolation Prediction

The momentum-based extrapolation predictor, which is inspired by the Nesterov technique for gradient descent (Nesterov 1983), is developed for ADMM because of its following features. 1) This approach does not need a large amount of historical information for predicting missing information. A large data set is not available when not many ADMM iterations are carried out. 2) The tuning parameters can make the prediction flexible and well-suited for ADMM. 3) This approach is very computationally cheap, which makes it suitable for integration into distributed optimization without adding additional computation costs.

4.4.4 Proposed Asynchronous ADMM Algorithm

The prediction-based A-ADMM pseudocode is given below. Each subproblem will be solved continuously using the values predicted by the momentum-based extrapolation. At least two data points are needed for predictions. Hence, up to the second iteration, the prediction steps are not implemented, and the most updated values of shared variables are used.

Proposed A-ADMM Algorithm to Solve OPF

```
1: **Initialize** $x_i^0, y_i^0, y_j^0, fl_i^0 = 1, \lambda_i^0 = \lambda_j^0, \rho$, and $k_i = 0$
2: **While** $\max(|y_i^{k_i} - \hat{y}_j{}^{k_i}|, |\hat{y}_i{}^{k_j} - y_j^{k_j}|) < \epsilon$ **do**
3:     **Repeat** for each $SP_i$
4:         Predict $\hat{y}_j^{k_i}$ of neighbor $j$ as follows
5:         **if** $fl_i^{k_i} = 0$
               $v_j^{k_j-1} = \mu v_j^{k_j-2} + \eta(\hat{y}_j^{k_i-1} - \hat{y}_j^{k_i-2})$
               $\hat{y}_j^{k_i} = \hat{y}_j^{k_i-1} + v_j^{k_i}$
6:         **elseif** $fl_i^{k_i} = 1$
               $v_j^{k_i-1} = \mu v_j^{k_i-2} + \eta(y_j^{(k_i-1)^+} - \hat{y}_j^{k_i-2})$
               $\hat{y}_j^{k_i} = y_j^{(k_i-1)^+} + v_j^{k_i-1}$
7:         **end**
8:         Restart $fl_i^{k_i}$
9:         Update $x_i$ and $y_i$ by
               $x_i^{k_i+1}, y_i^{k_i+1} = \text{argmin} \sum_{u \in \Omega_i} a_u p_u^2 + b_u P_u + c_u + \lambda_i \left( y_i^{k_i+1} - \frac{y_i^{k_i} + \hat{y}_j^{k_i}}{2} \right) + \frac{\rho}{2} \left\| y_i^{k_i+1} - \frac{y_i^{k_i} + \hat{y}_j^{k_i}}{2} \right\|^2$
10:        Send $y_i^{k_i+1}$ to neighboring OPF subproblems
11:        Update multipliers as $\lambda_i^{k_i+1} = \lambda_i^{k_i} + \rho \left( y_i^{k_i} - \hat{y}_j^{k_i} \right)$
12:        Set $k_i \leftarrow k_i + 1$
13: **end**
```

4.5 Numerical Results and Discussions

We implement the synchronous parallel ADMM and the proposed A-ADMM to solve DC OPF for a 22-bus system, the IEEE two-area RTS-96 system with 48 buses, and the IEEE 118-bus system, and AC OPF for the 48-bus and 118-bus systems.

- Computational costs of OPF subproblems of the 22-bus and 118-bus systems are heterogeneous.
- Computational costs of OPF subproblems of the 48-bus system are homogeneous.

Simulations are carried out with MATLAB on a personal computer with a 2.6GHz CPU and 16GB of RAM. Systems topology and characteristics are given in (2020). To analyze the performance of the proposed algorithm in detail, we have studied the impact of predictions, momentum-based extrapolation parameters, and penalty multipliers on the convergence performance of A-ADMM. Two convergence measures are used to analyze the results. The first index is the relative error between optimal generation costs obtained by centralized and A-ADMM algorithms (see (2.22)). The second index, which is a feasibility criterion, is the mean primal residual or the summation of all consistency constraints divided to the total number of shared variables (n_s).

$$s_{avg}^{k} = \frac{\sum |y_i^k - y_j^k|}{n_s} \tag{4.19}$$

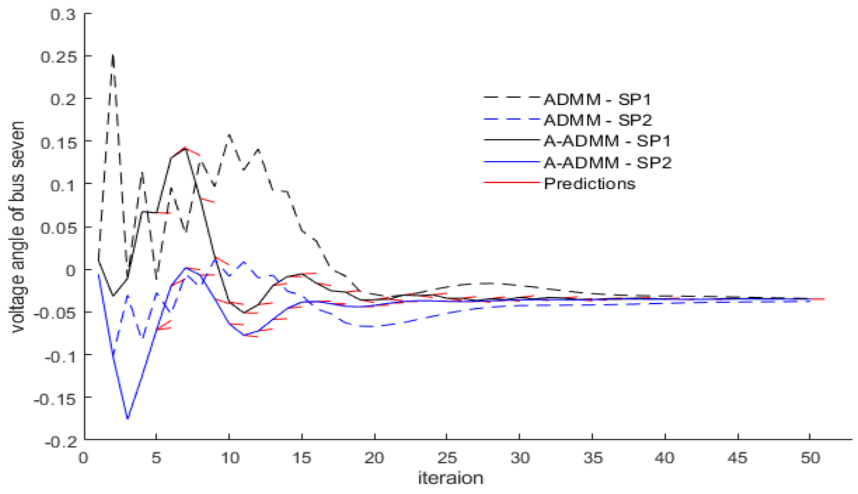
Note on Simulation Studies: To have a fair comparison between A-ADMM and ADMM, we have carried out both algorithms for the same number of iterations for each test case, except for simulations in Section IV.D, whose goal is to compare simulation time and the number of iterations. We have considered the iteration counter of the slowest subproblem in A-ADMM since subproblems have different iteration counters.

4.5.1 Impact of Prediction on Convergence Performance

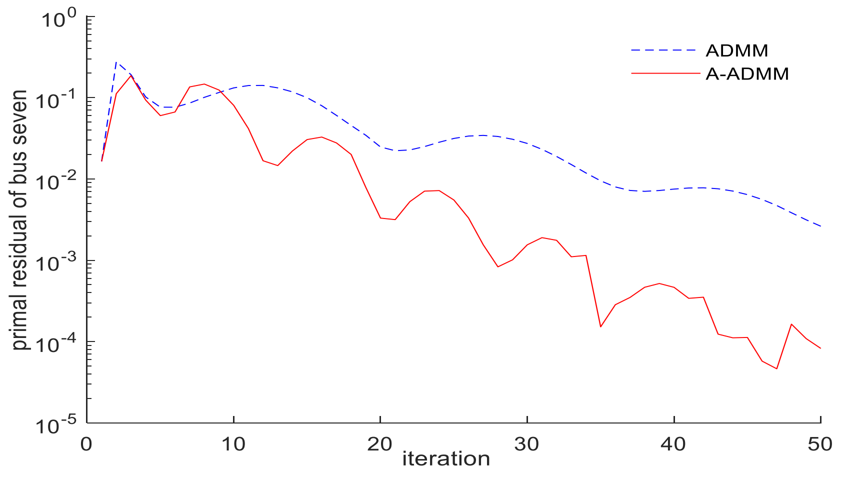
A-ADMM and ADMM are applied to solve DC OPF for the 48-bus system. It is discussed how predictions improve the performance of distributed optimization. The system has two zones. A DC OPF subproblem is formulated for each zone. Voltage angles of buses 7, 13, 23, 27, 39, and 41 are shared variables whose initial values are set to zero. Tuning parameters are set as $\lambda^0 = 100$, $\rho = 100$, $\eta = 0.2$, and $\mu = -0.8$. Figure 4.5a illustrates the values of shared variables corresponding to the voltage angle of bus seven (θ_7) over the course of iterations. The blue and black solid/ dashed lines show the shared variable values determined by A-ADMM/ ADMM. Both algorithms converge to the same values and make differences between shared variables zero. Red lines are predictions at each iteration. These predictions make the behavior of A-ADMM smoother than that of ADMM. Fewer jumps are observed in values obtained by A-ADMM. Figure 4.5b shows the primal residual, (4.8), which vanishes much faster if A-ADMM is used. The convergence speed for A-ADMM is remarkably better than that of ADMM. For example, A-ADMM's error reaches 0.0036 after 25 iterations while ADMM reaches this gap after 50 iterations.

4.5.2 Impact of Momentum-Extrapolation Parameters on A-ADMM

To demonstrate the impact of prediction parameters on the convergence performance, A-ADMM is implemented to solve DC OPF for the IEEE 118-bus system, which has three zones. Parameters λ^0 and ρ are set to 100, and prediction parameters are varied as $\eta = \{0.02, 0.25, 0.4\}$ and $\mu = \{-0.98, -0.75, -0.6\}$. By increasing η and decreasing μ, the momentum-extrapolation becomes less conservative and gives more weight to the trend of previous data. This increases the risk of large prediction errors, especially at peaks and valleys. In contrast, by decreasing η and increasing μ, the algorithm becomes more conservative, and predictions do not become far from the previous points.



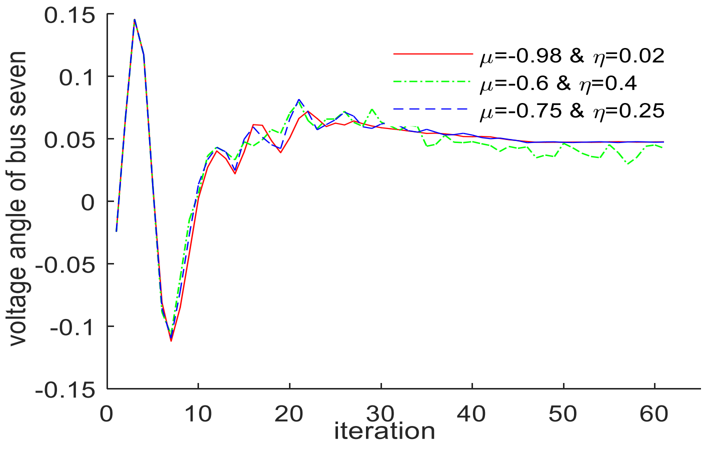
(a)



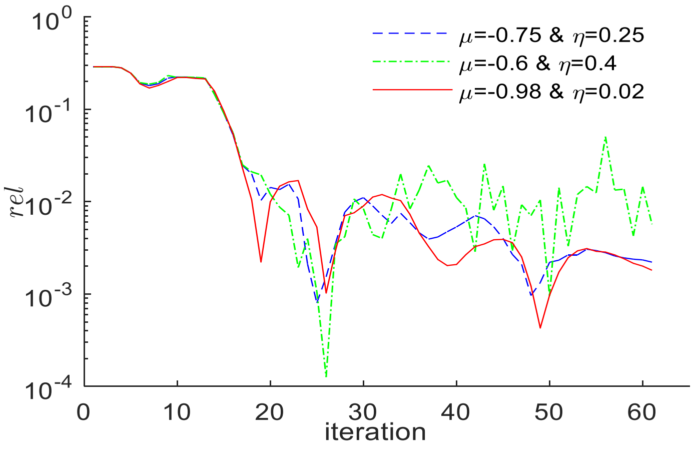
(b)

Figure 4.5 a) Shared variable values and b) the primal residual over iterations.

That is, predictions will be in a safe zone around the previous point. Figure 4.6 illustrates the predicted values for a shared variable (voltage angle of bus seven) and the rel index for different values of η and μ. While larger η makes predictions and the rel index less smooth, larger μ results in smoother predictions and rel. We suggest setting conservative training parameters for problems with many shared variables and tie-lines and selecting less conservative values for problems with a few shared variables. Observing Fig. 4.6, we conclude that $\mu = \{-0.98, -0.75\}$ and $\eta = \{0.02, 0.25\}$ are proper for this test system.



(a)



(b)

Figure 4.6 a) Predictions for a shared variable and b) rel obtained by A-ADMM.

4.5.3 Optimality Gap and Primal Residual

A-ADMM is applied to solve DC OPF for the 22-bus system, the 48-bus system, and the IEEE 118-bus system. The 22-bus system includes a transmission system connected to two active distribution systems. The initial values for shared variables are set to zero, and $\lambda^0 = \rho = 100$. For the 22-bus system, the 48-bus system, and the 118-bus system, $\{\mu, \eta\}$ are selected as {-0.7,0.3}, {-0.8,0.2}, and {-0.97,0.03}, respectively. The rel and $\mathscr{s}_{avg}$ indices are shown in Figs. 7-9. After the same number of iterations, A-ADMM provides better rel (optimality criterion) and $\mathscr{s}_{avg}$ (feasibility criterion).

Due to the heterogeneity of computational burdens of OPF subproblems for the 22-bus and 118-bus systems, there are subproblems that need to predict shared variables' values for several iterations. In contrast, due to the homogeneity of computation costs of the OPF subproblems for the 48-bus system, these subproblems would observe missing information for, usually, not more than one or two iterations. The results illustrated in Figs. 7-9 show that A-ADMM outperforms the classical synchronous parallel ADMM under these two possible situations. A-ADMM and ADMM are the same under perfect homogeneity of computation costs of subproblems; however, such an ideal condition is almost not attainable in real-world problems.

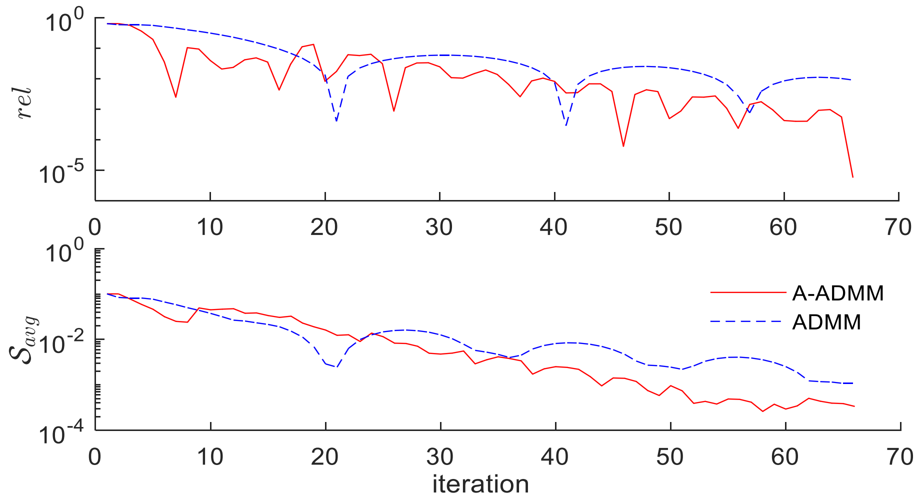


Figure 4.7 The rel and s_{avg} indices over iterations for the 22-bus system.

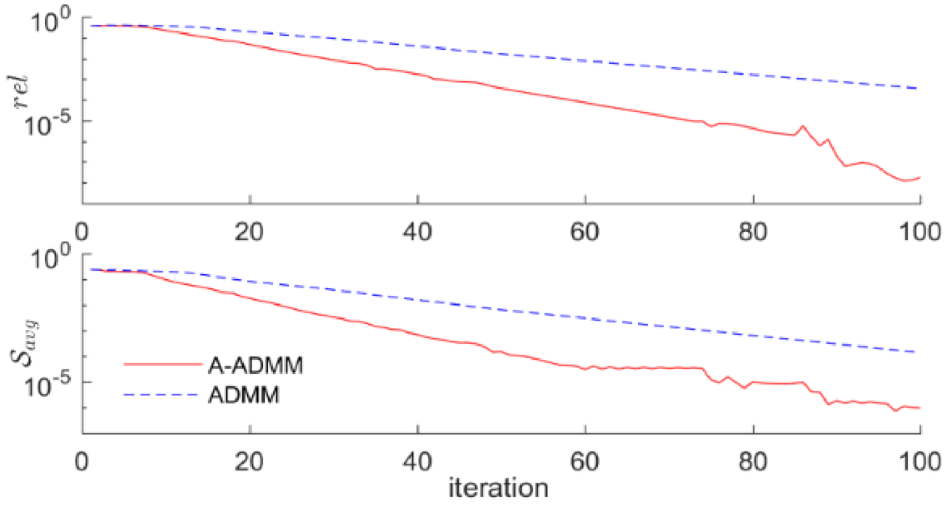


Figure 4.8 The rel and s_{avg} indices over iterations for the 48-bus system.

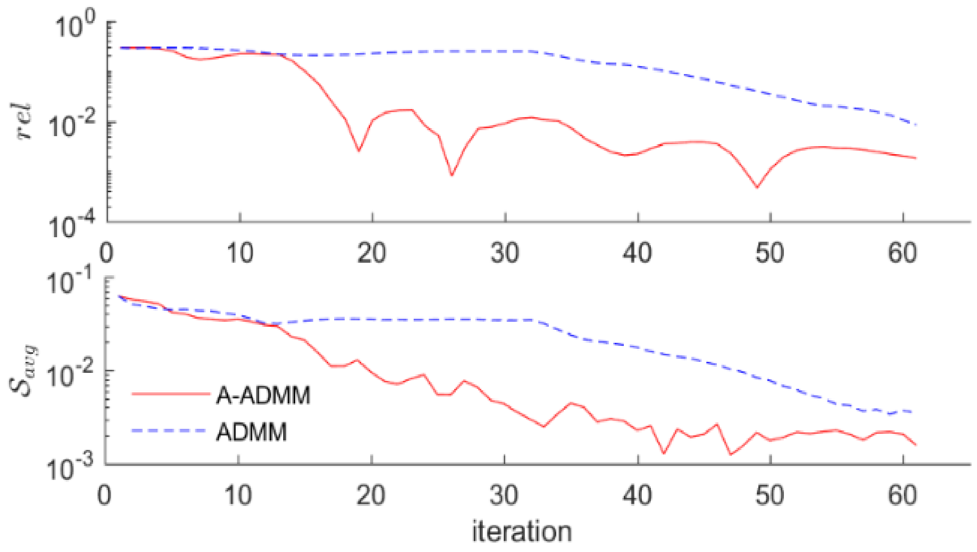
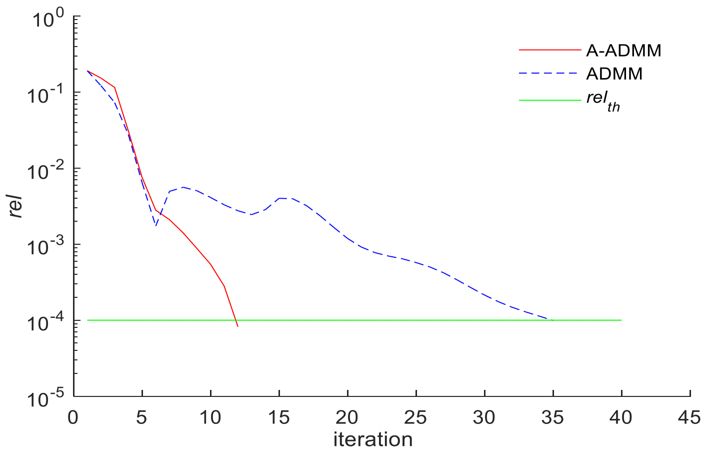


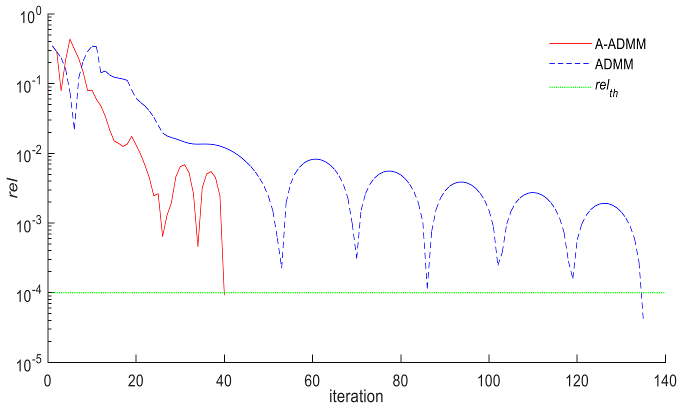
Figure 4.9 The rel and s_{avg} indices over iterations for the 118-bus system.

4.5.4 Simulation Time and Iteration Comparison

The desired rel index is set to $rel_{th} = 1e - 4$, and A-ADMM and ADMM are implemented to solve DC OPF for the 22-bus and 48-bus systems. Parameters μ and η are respectively set to -0.9 and 0.1, and $\lambda^0 = \rho = 500$. Figure 4.10 and Table 4.1 show the results. A-ADMM reaches to the desired rel value much faster than ADMM. For the 22-bus system, A-ADMM takes 12 iterations within 0.8 seconds, while ADMM needs 35 iterations and 1.1 seconds to achieve the desired rel value. For the 48-bus system, A-ADMM takes 95 fewer iterations and 70% less time than ADMM.



(a)



(b)

Figure 4.10 The rel index over iterations for a) 22-bus b) and 48-bus systems.

Table 4.1 Operating Cost Obtained by Different Algorithms

	22-Bus		48-Bus	
	ADMM	A-ADMM	ADMM	A-ADMM
Iterations	35	12	135	40
Time (sec)	1.1	0.8	6.8	2.1

4.5.5 Multipliers Analysis

The penalty multiplier ρ has a significant impact on the convergence behavior of ADMM. Large ρ values increase the solution speed while increasing the risk of finding a suboptimal solution or divergence. In contrast, a small ρ enhances the chance of finding the optimal solution with the cost of slowing the convergence speed. We evaluate the impact of ρ on A-ADMM using DC OPF for the 48-bus system and compare it with that of ADMM. Simulations are carried out for $\rho = \{100{,}500{,}2000\}$, $\mu = -0.8$, and $\eta = 0.2$. To have a fair comparison, ADMM and A-ADMM are implemented for 50 iterations. As illustrated in Figs. 11-13, for small, medium, and large values of ρ, A-ADMM works better than ADMM in terms of rel and $\mathcal{s}_{avg}$. The method presented in (Mohammadi and Kargarian 2020) can also be applied to adjust the value of ρ and enhance the convergence performance of A-ADMM.

4.5.6 AC OPF Analysis

We have applied A-ADMM and ADMM to solve AC OPF for the 48-bus and 118-bus systems. A fictitious bus is added in the middle of each tie-line, and a pseudo generation is added to this bus. The bus and the pseudo generation are duplicated, and a copy is assigned to each neighbor. MATPOWER is used to solve AC OPF subproblems. The distributed optimization parameters are set as $\mu = -0.8$, $\eta = 0.2$, $\lambda^0 = 0$ and $\rho = 40$ for the 48-bus system, and $\mu = -0.9$, $\eta = 0.1$, $\lambda^0 = 1$ and $\rho = 0.2$ for the 118-bus system. Figure 4.14 demonstrates the rel indices obtained by ADMM and A-ADMM. The algorithms are run for 60 iterations for the 48-bus system and 90 iterations for the 118-bus system. The rel indices obtained by A-ADMM are always below those obtained by ADMM. It shows that A-ADMM outperforms the synchronous parallel ADMM.

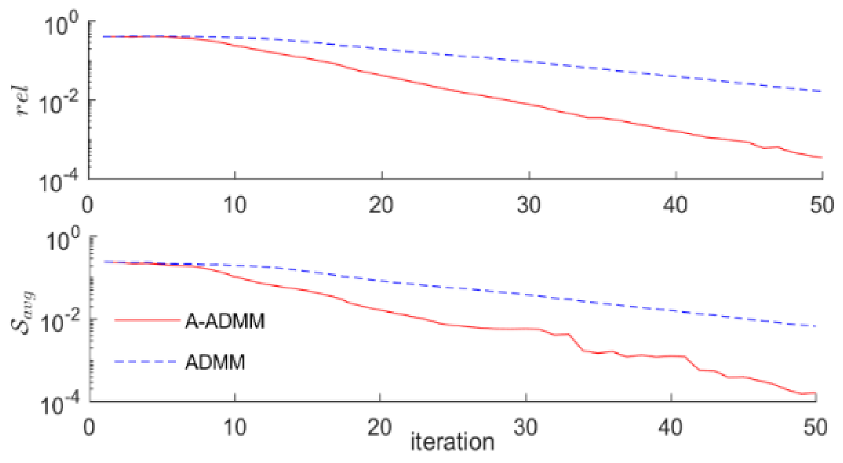


Figure 4.11 The rel and s_{avg} indices over iterations for $\rho = 100$.

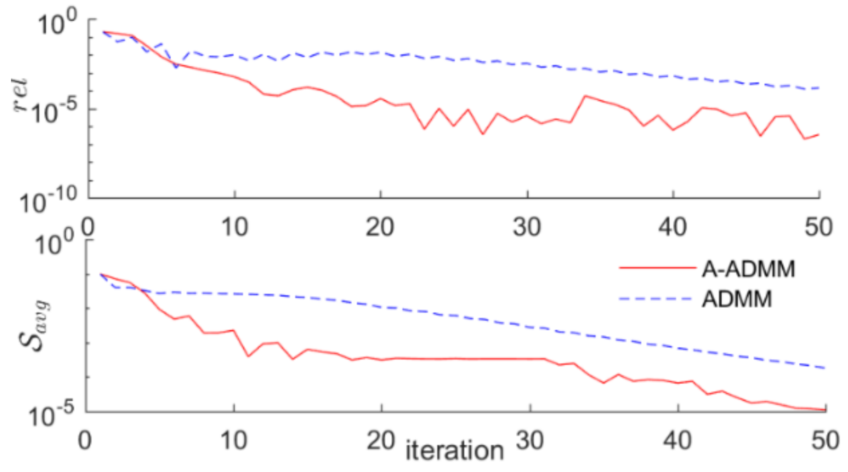


Figure 4.12 The rel and s_{avg} indices over iterations for $\rho = 500$.

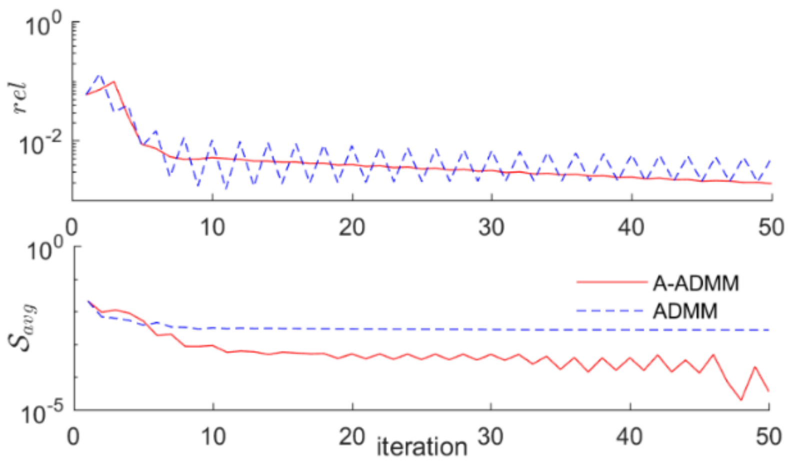
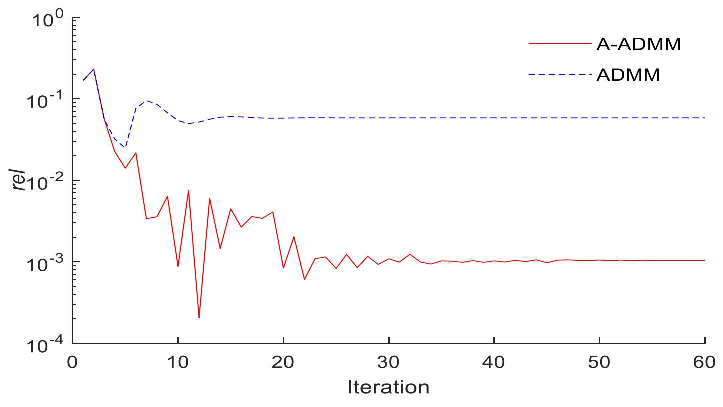
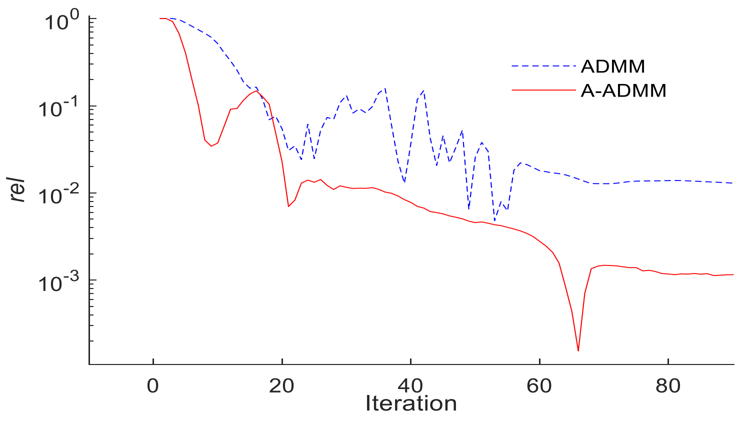


Figure 4.13 The rel and s_{avg} indices over iterations for $\rho = 2000$.



(a)



(b)

Figure 4.14 The rel index for AC OPF on a) 48-bus and b) 118-bus systems.

4.6 Conclusion

A momentum-extrapolation based asynchronous ADMM is proposed to solve OPF. A prediction-correction approach is designed to predict missing information within a neighborhood of the most updated values of shared variables. As compared to the classical synchronous parallel ADMM, the proposed A-ADMM reduces the unproductive time and under-utilization of computation resources if the computational burdens of OPF subproblems are heterogeneous.

Simulation results on several test cases show the promising performance of the proposed algorithm for solving DC OPF and AC OPF. It is illustrated that A-ADMM outperforms ADMM, in particular, if the sizes of subproblems are heterogeneous. After the same number of iterations, A-ADMM provides smaller optimality and feasibility gap measures. Also, it is discussed that the values of penalty multipliers do not degrade the convergence behavior of the proposed algorithm. The extrapolation and momentum coefficients should be set properly to obtain good predictions and enhance the algorithm’s performance. If subproblems are heterogeneous, the momentum term should dominate to have more conservative predictions and prevent large prediction errors.

CHAPTER 5. LEARNING-BASED ASYNCHRONOUS DOUBLE-LOOP ADMM FOR OPTIMAL POWER FLOW

5.1 Introduction

Majority of existing distributed optimization algorithms are synchronous in which subproblems are solved once at each iteration upon receiving all shared variables information from their neighboring subproblems. This synchronization requirement degrades the scalability of distributed optimization and under-utilization of computation resources, especially if subproblems are computationally heterogeneous. Few recent efforts have attempted to develop asynchronous algorithms for power systems operation. Although these algorithms solve subproblems in a parallel asynchronous manner, they have difficulties in proceeding toward the optimal solution with reasonable steps after implementing each iteration since they use the latest information of neighboring subproblems to carry out future iterations. This also forces faster subproblems to go to an idle mode after a couple of iterations, until updated information is received from other subproblems. To address these limitations, we propose a learning-based double-loop asynchronous alternating direction method of multipliers (LA-ADMM) that i) has information prediction capability and ii) allows a significant level of asynchrony between subproblems. We develop a momentum-extrapolation based prediction-correction technique to predict missing information. Furthermore, we present an online streaming-based anomaly classification method to observe the performance of predicted data and control Lagrange multipliers update while proceeding with predictions and solving subproblems asynchronously. The proposed LA-ADMM is applied to solve the optimal power flow problem for several test systems. Promising results are obtained as compared to the classical synchronous ADMM and asynchronous ADMM without the anomaly switch control.

5.2 Contribution

The main contributions of this chapter are summarized as follows:

- A double-loop asynchronous ADMM is proposed, with a momentum-extrapolation prediction-correction step, to solve the OPF problem distributedly.
- The designed prediction-correction step does not need large datasets and can be used at very early iterations with few available data points. In contrast, complex machine learning-based algorithms need a large dataset for predicting missing information, which makes them challenging to be used for predicting missing information in distributed optimization.
- An inner loop is embedded to prevent prediction error propagation in the case of a high level of asynchrony of subproblems. The concept of the inner loop is used for sequential distributed optimization in (DorMohammadi and Rais-Rohani 2013, Mohammadi, Mehrtash et al. 2018); however, it is new in the context of asynchronous distributed optimization.
- An anomaly detection switch is designed to control inner loop activation-deactivation automatically. Many efforts have attempted to develop anomaly detection techniques for online streaming (Zhang, Zhao et al. 2019); however, no effort has been reported in the literature on anomaly detection in the context of distributed optimization.

5.3 Asynchronous ADMM (A-ADMM)

Sequential and synchronous parallel ADMM optimal power flow algorithms are described in Chapter 4. Asynchronous parallel ADMM is presented recently in (Guo, Hug et al. 2017) to remove the need for synchrony between OPF subproblems and enhance the computation efficiency of distributed optimization. At each iteration, the latest values of shared variables obtained in the previous iteration are used to allow OPF subproblems to be solved continuously instead of waiting for all other subproblems to be solved. Using the latest updated information makes asynchronous ADMM (A-ADMM) conservative, slows moving toward the optimal solution after carrying out each iteration, and restricts the allowable level of asynchrony between subproblems.

5.4 Learning-based asynchronous ADMM

We improve asynchronous ADMM by adding a prediction-correction step to the algorithm that forecasts shared variable values based on information obtained over the course of iterations. At the beginning of each iteration, an OPF subproblem checks whether it has received share variable values from its neighboring subproblems. If not, a regression-based technique is used to predict the values of shared variables (considered as *missing information*) and allow the corresponding OPF subproblem to be solved. An anomaly switch controller is designed to bypass error propagation impact on Lagrange multipliers if the number of consecutive iterations with missing information goes beyond few iterations; otherwise, inevitable prediction errors may be accumulated and degrade convergence performance or even divergence.

5.4.1 LA-ADMM with Anomaly Detection and Inner Loop

We have observed that the momentum-based extrapolation prediction-correction approach, presented in Chapter 4, works well for predicting missing values of shared variables for up to a few iterations. If no updates of actual shared variables are received beyond several iterations, even the correction step may not prevent ADMM from moving toward a point with low-quality results or a suboptimal/infeasible point. We add a few steps to A-ADMM to avoid obtaining low-quality results if the level of asynchrony between OPF subproblems is large and missing information is observed in many consecutive iterations or if predicted values of missing information are not accurate enough.

By observing the sensitivity of optimization to variation of multipliers λ and inspiring by analytical target cascading (Tosserams, Etman et al. 2006, DorMohammadi and Rais-Rohani 2013), we modify the A-ADMM structure by adding a switch-controlled inner loop. Figure 5.1 shows the structure of the proposed learning-based A-ADMM (LA-ADMM). The switch controller is an autonomous monitoring unit that decides whether to activate the inner loop or not by observing the prediction step performance. If the inner loop is activated, multipliers λ will not be updated while shared variables are updated/predicted iteratively. By doing so, the algorithm reduces the primal residual (3.6) for a given λ, instead of updating λ in a wrong direction that may lead to divergence.

Inner loop switch controller with unsupervised anomaly detection for online streaming: The switch controller works based on anomaly detection. We develop an algorithm that reads shared variables predicted by the momentum-based extrapolation prediction-correction approach and identifies whether there is an anomaly in the stream of predicted values. We use the total generation cost function of LA-ADMM ($f_i^{LA-ADMM}$) at each iteration as an index to evaluate whether predictions cause a considerable error or not. The process of carrying out distributed optimization is similar to online steaming in which new data is obtained iteration by iteration. Thus, an anomaly detection technique is required that works for online streaming data even if not enough data is available at first few iterations. This requirement makes most machine learning techniques not applicable for anomaly detection when running distributed optimization iteratively. $f^{LA-ADMM}$ is updated at each iteration, and the number of iterations of LA-ADMM to convergence is problem-dependent. This makes labeling the online stream of predicted shared variable values challenging.

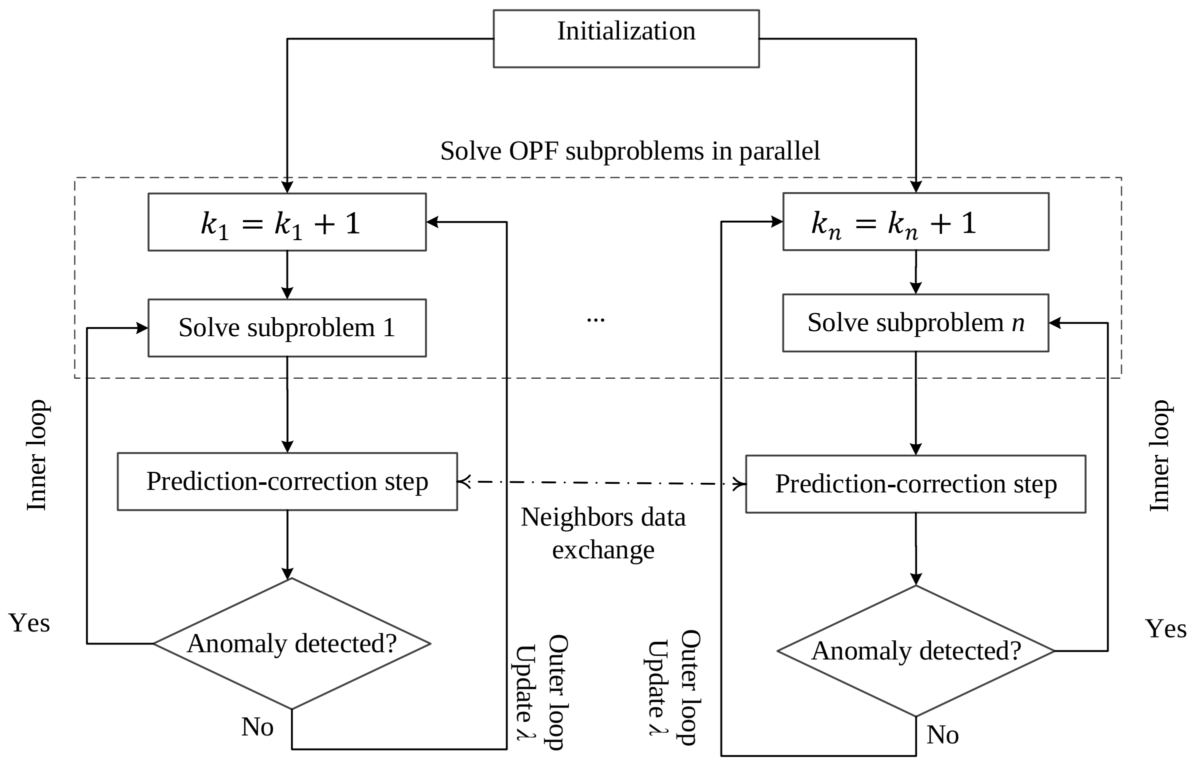


Figure 5.1 Flowchart of the proposed LA-ADMM algorithm.

We propose an unsupervised ground-up anomaly learner for online steaming that is suitable for distributed optimization applications. Such an effort has not been made in the literature. The proposed algorithm has two steps. In the first step, based on the $f_i^{LA-ADMM}$ curve obtained over the course of iterations 1 to $k-1$, we predict the generation cost function $\widetilde{f_i}^{LA-ADMM}$ at iteration k. We then measure how much the severity of anomaly is in the current iteration as compared to previous iterations. To do so, we implement the following steps.

- *Step 1.1. Wavelet denoising:* Denoising is widely used in image processing to reduce noise in pictures. Wavelet denoising splits data into low- and high-frequency categories. Most useful information exists in low frequency. Data in the high-frequency category is filtered out to reduce noise and smoothen data. We use wavelet to detect noise, or anomaly, in the generation cost curve over the course of distributed optimization iterations.

- *Step 1.2. Moving average:* The moving average function, usually named as smooth function, slides a window of size of five on the entire selected data and replaces each point with the mean value of the last five iterations. This is similar to a low pass filter. Large oscillations are mitigated using the moving average, while the trend of the curve will not deteriorate. Moving average is a smoothing technique in the time domain, whereas wavelet is in the frequency domain.

- *Step 1.3. Energy difference evaluation:* Wavelet denoising removes high frequency, and moving average keeps low frequency and smoothens larger jumps. For further noise reduction, we carry out denoising and moving average several times until the energy difference between two signals becomes less than a threshold (e.g., 0.1%). Desired curve $\widetilde{f_i}^{LA-ADMM}$, which is without anomaly and noise, is obtained upon completion of this step.

We calculate the relative energy difference between denoised $\widetilde{f_i}^{LA-ADMM}$ and original $f_i^{LA-ADMM}$ as $\epsilon = \frac{\left|\widetilde{f_i}^{LA-ADMM} - f_i^{LA-ADMM}\right|}{f_i^{LA-ADMM}}$. As an example, Fig. 5.2 demonstrates ϵ for simulation on the IEEE two-area RTS 96 system. The relative energy difference between wavelet and moving average is set to 10% and 0.1%, and the obtained curves are denoted by denoised I and II, respectively. Denoised I, with less smoothened noise, is closer to the original data, whereas the curve of denoised II is smoother. The threshold is problems dependent and can be set by the user.

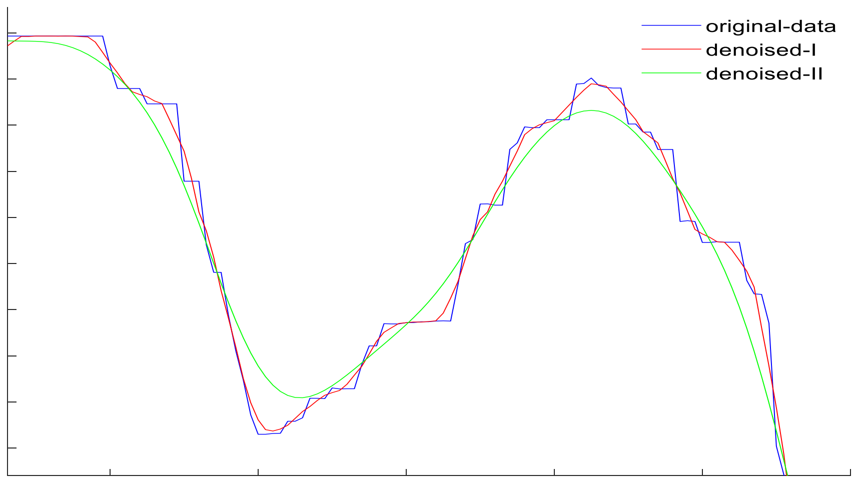


Figure 5.2 Comparison between $\widetilde{f_i}^{LA-ADMM}$ and $f_i^{LA-ADMM}$.

In the second step, the severity of anomaly is measured. At each iteration, we use $f^{LA-ADMM} - \tilde{f}^{LA-ADMM}$ as the anomaly measure and select the most recent 20 iterations to define the severity of anomaly. Anomaly measurement is a relative index, and selecting a single threshold to categorize the severity anomaly would degrade anomaly detection performance. We use k-means to categorize and sort anomaly to ϕ groups. Depending on which class the last iteration belongs to, we define how severe anomaly would be. For example, if $\phi = 10$ and classes larger than a certain threshold $\psi = 7$ counted as severe anomalies, any score larger than seven means that the inner loop should be activated. Since the Lagrange multipliers have the most critical impact on ADMM convergence behavior, they will not be updated in the inner loop. This will prevent the propagation of shared variables prediction errors on Lagrange multipliers. The anomaly detection algorithm is explained in Algorithm.

Algorithm: Anomaly detection for distributed, iterative LA-ADMM

```
1: Initialize $\phi$, $\psi$, and $f^{LA-ADMM}$
2: While $\epsilon < 0.001$ do
3:      $\chi =$ denoising($f^{LA-ADMM}$)
4:      $\chi' =$ smooth($\chi$)
5:      $\epsilon = \sum|\chi - \chi'| / \sum|\chi'|$
6:      $\tilde{f}_i^{LA-ADMM} = \chi'$
7:      class = k-means ($f^{LA-ADMM} - \tilde{f}^{LA-ADMM}, \phi$)
8:      if $\psi <$class
9:          activate inner loop
10:     else
11:         deactivate inner loop
12:     end
13: end
```

5.5 Numerical results

The classical ADMM and the proposed A-ADMM and LA-ADMM are implemented to solve the OPF problem for a 22-bus system, the IEEE two-area RTS 96 (48-bus) system, and the IEEE 118-bus system, and the results are compared. The systems' characteristics are given in (Kargarian, Fu et al.). To have comprehensive analyses, impacts of parameters and communication delay on the algorithms are investigated. The rel and s_{avg} indices (see (2.22) and (4.20)) are used for performance evaluation. The smaller rel and s_{avg} are, the better the obtained solution will be. Two possible conditions are studied in terms of the level of asynchrony of subproblems, low and high asynchrony levels. All simulations are carried out on a personal computer with a 2.6 GHz CPU and 16 GB of RAM.

5.5.1 Low Asynchrony Level

The level of asynchrony of OPF subproblems is low, which means computational times of OPF subproblems are not significantly different. A subproblem would not observe missing information from its neighbors for more than a few iterations, e.g., $2\sim5$ iterations.

Impact of Momentum-based Extrapolation Parameters: The IEEE two-area RTS 96 system is used, and a DC OPF subproblem is formulated for each area. Shared variables are voltage angles of border buses 7, 13, and 23 in area one and those of buses 27, 39, and 41 in area two. Shared variables are initiated to zero. Tuning parameters is set as $\lambda^0 = 100$ and $\rho = 100$, and two sets of momentum-based extrapolation parameters $\{\mu, \eta\}$ are selected as $\{-0.97, 0.03\}$ and $\{-0.7, 0.3\}$. The number of anomaly groups is set to $\phi = 8$. We have studied two cases with $\psi = 6$ and $\psi = 5$. The classical synchronous ADMM, A-ADMM and LA-ADMM are implemented and the rel index is plotted in Fig. 5.3. The inner loop reduces small oscillations in LA-ADMM as compared to A-ADMM and enhances the asynchronous algorithm performance. During iterations 83-93 in Fig. 5.3.a, LA-ADMM has less fluctuations than that of A-ADMM. Moreover, as depicted in Fig. 5.3.b, the inner loop prevents large prediction error propagation on λ due to having missing information in several consecutive iterations. Hence, LA-ADMM reaches more accurate results than that of A-ADMM.

Impact of Penalty Parameter: The value of penalty multiplier ρ affects convergence behaviors of ADMM, A-ADMM, and LA-ADMM. Large ρ may increase the convergence speed while increasing the risk of finding a suboptimal solution or divergence. In contrast, a small ρ improves the chance of finding the optimal solution while the pace of convergence might be slow.

We have set multipliers λ and ρ to 200 and run ADMM, A-ADMM, and LA-ADMM for 325 iterations. We have fixed μ and η to -0.97 and 0.03 to have a fair comparison. The rel and s_{avg} indices are depicted in Fig. 5.4. LA-ADMM and A-ADMM outperform ADMM in terms of rel and s_{avg}. We have set λ and ρ to 700, which is relatively large, and have observed that prediction errors have more impact on convergence performance than the case with small multipliers. Comparing rel and s_{avg} indices in Fig. 5.5 shows that the proposed LA-ADMM with anomaly detection provides smaller indices than those of A-ADMM and ADMM.

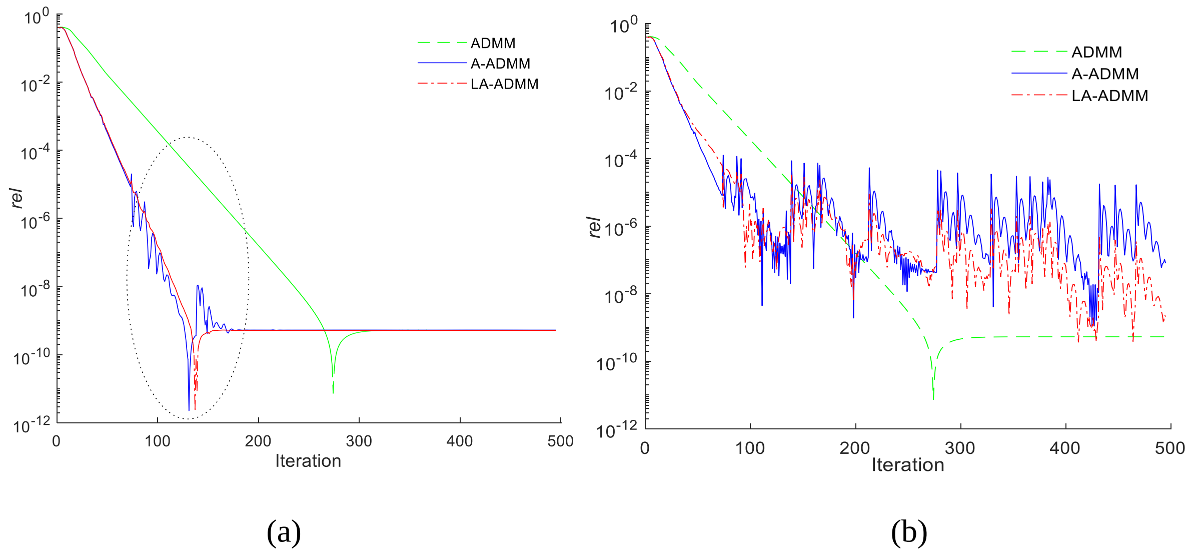
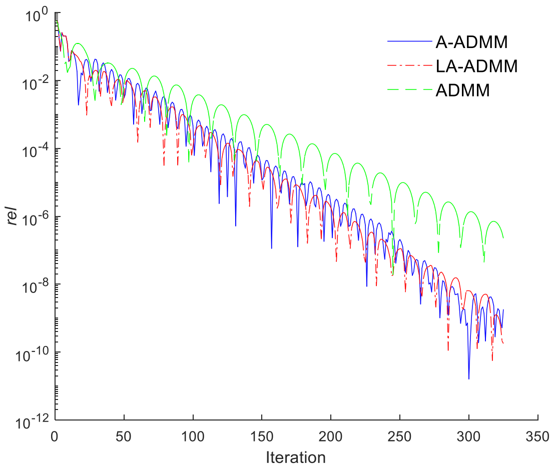
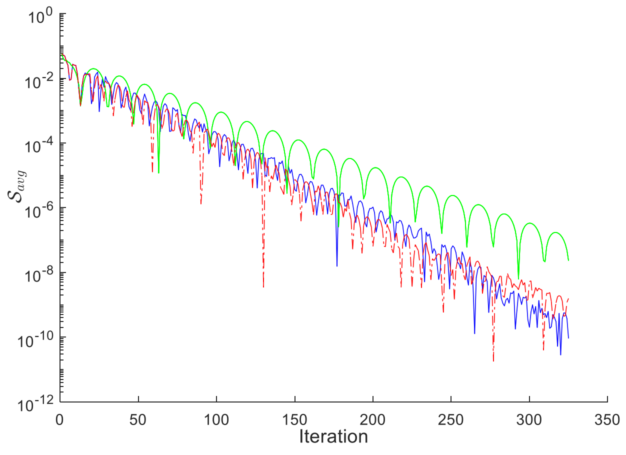


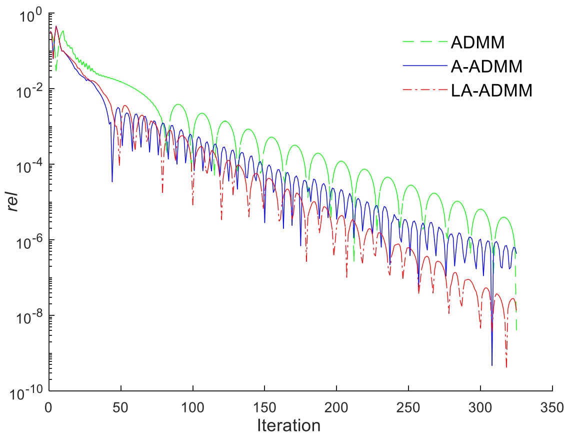
Figure 5.3 rel over the iteration for $\{\mu, \eta, \psi\}$ equal to a) $\{-0.7, 0.3, 6\}$ and b) $\{-0.97, 0.03, 5\}$.

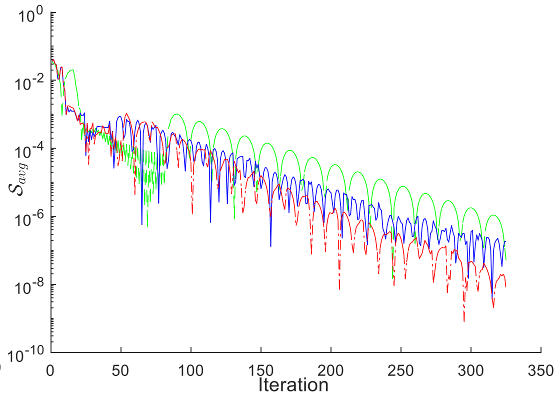




(a) (b)

Figure 5.4 a) rel and b) s_{avg} over iterations for relatively small multipliers.





(a) (b)

Figure 5.5 a) rel and b) s_{avg} over iterations for relatively large multipliers.

5.5.2 High Asynchrony Level

If computational burden of subproblems is considerably heterogeneous and/or communication failure/delay happens, subproblems may observe a high level of asynchrony. Under this condition, subproblems may not receive updated values of shared variables from their neighbors for several iterations, e.g., more than ten iterations. We simulate such a condition in this case and study the performance of A-ADMM and LA-ADMM.

We have intentionally manipulated missing information patterns and have created a wide range of possible situations for the level of asynchrony. Figure 5.6 illustrates the idle time of subproblems of the IEEE 118-bus system. For the first subproblem, for instance, the maximum number of iterations with missing information is five. However, we have manipulated it to create cases with 1~13 iterations with missing information. To simulate realistic yet more challenging conditions, we have used a significantly distorted communication delay pattern (compare blue and red patterns in Fig. 5.6).

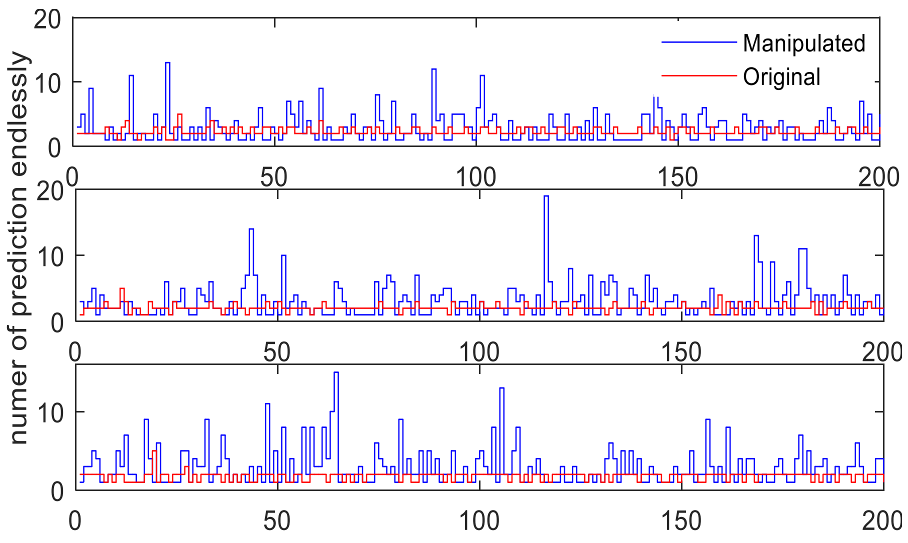


Figure 5.6 Designated flag for the 118-bus system for the first, second, and third subproblems.

Inner Loop Performance: We have studied the behavior of λ, rel, and s_{avg} over the course of iterations using the IEEE 118-bus system. Figure 5.7 demonstrates the impact of inner loop on the rel index. The solid red line and the dashed blue line depict rel of LA-ADMM and A-ADMM, respectively. The highlighted gray areas illustrate iterations in which the inner loop is activated. Wide/narrow gray areas indicate that more/less inner loop iterations are carried out at each outer loop iteration. Both algorithms are carried out for 600 iterations. The inner loop enhances the behavior of rel index obtained by LA-ADMM and prevents large errors if oscillations are observed in predictions and anomaly is detected.

To further analyze the impact of anomaly detection and inner loop on the convergence behavior, multiplier λ_1 corresponding to shared variable θ_{33} (voltage angle of bus 33 that is shared between subproblems one and two) is illustrated in Fig. 5.8. Multiplier λ_1 obtained by LA-ADMM oscillates at the beginning and shows a smoother behavior by carrying out more iterations. This pattern is similar to that of the classical sequential ADMM, which has a desirable multiplier update pattern. Although A-ADMM works well at the beginning, large fluctuations are observed when more iterations are carried out.

Performance Analysis Under Very High Level of Asynchrony: We study the performance of the proposed LA-ADMM under a very high level of asynchrony of subproblems that can be either because of extreme unbalance computational burden or information exchange failure. A prediction-only-based asynchronous algorithm would face challenges under such conditions due to probable significant prediction errors. We have applied A-ADMM and the proposed LA-ADMM on the 22-bus system, the 48-bus system, and the IEEE 118-bus system. We have manipulated the level of asynchrony to create very high levels of asynchrony. The tuning parameters are set as $\mu = 0.97$, $\eta = -0.03$, $\lambda^0 = 100$, $\rho = 100$, $\psi = 3$, and $\phi = 10$. The resultant rel indices for the three studied systems are shown in Figs. 5.9-11. The prediction-only-based A-ADMM does not perform well because of considerable prediction errors due to not receiving the actual values of shared variables for many iterations. The prediction-anomaly detection-based LA-ADMM works well for the three systems, which have different levels of asynchrony of subproblems. Comparing the performance of the classical synchronous ADMM and the proposed asynchronous LA-ADMM shows that both algorithms have roughly the same, good behaviors over the course of iterations. Note that the classical synchronous ADMM works well regardless of the level of asynchrony as this approach does not move forward until updated information is exchanged between all subproblems at each iteration (i.e., perfect level of synchrony). Thus, ADMM rel can be considered as a benchmark. Figures 5.9-11 show that although LA-ADMM is asynchronous, it provides rel indices that follow benchmark values over iterations.

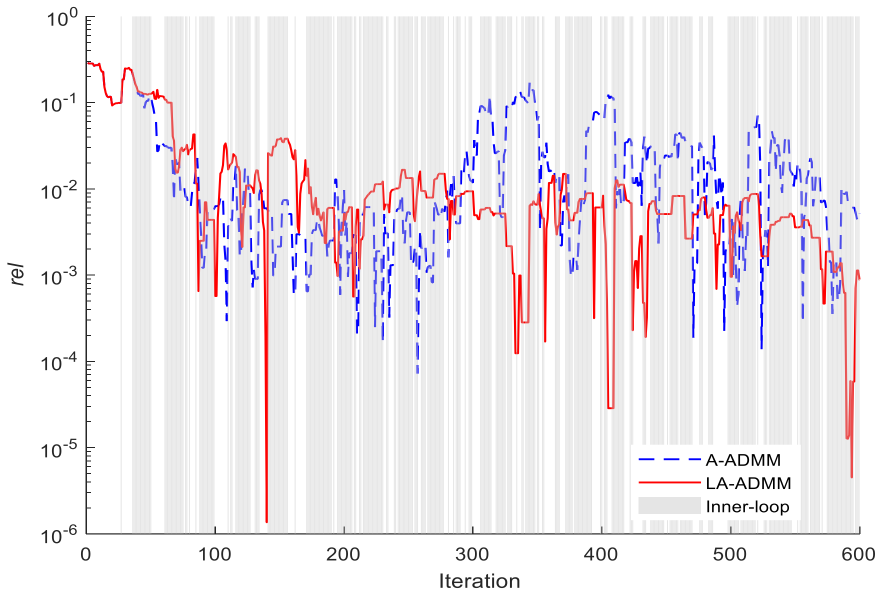


Figure 5.7 rel over the course of iterations under high asynchrony level.

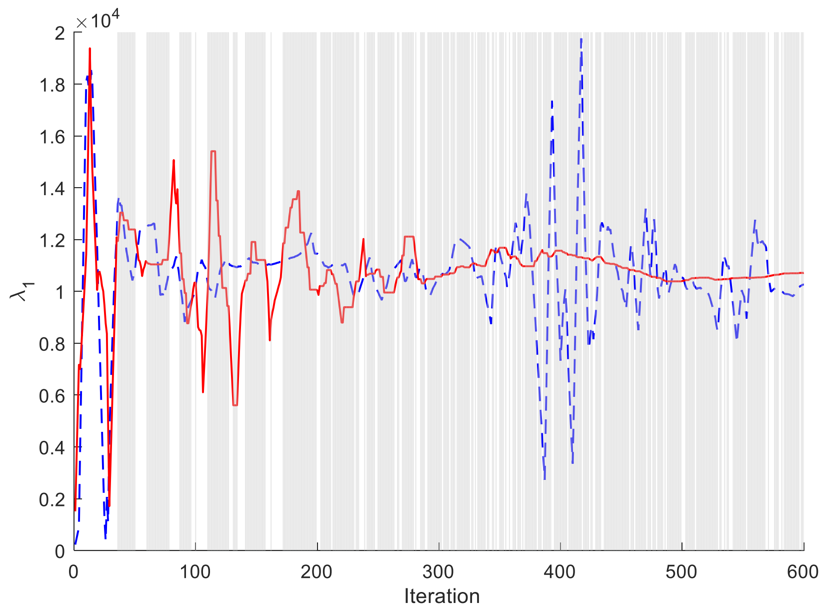


Figure 5.8 λ_1 over the course of iterations under high asynchrony level.

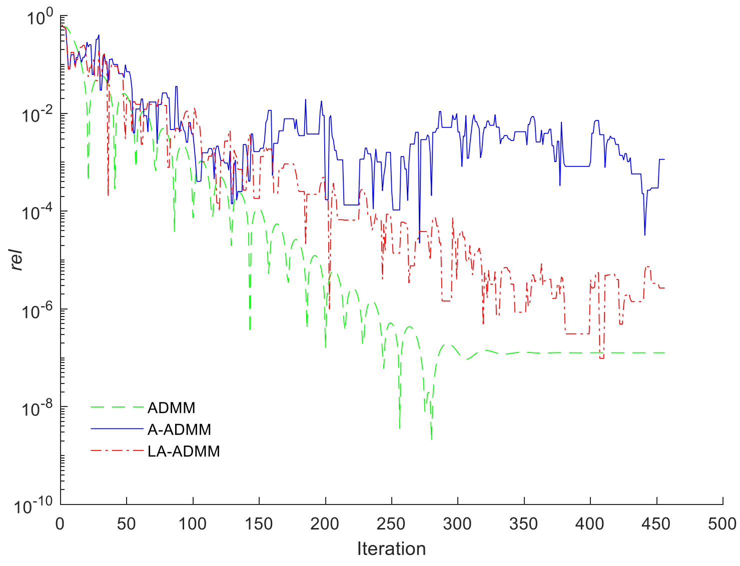


Figure 5.9 rel over iterations for the 22-bus system under a very high level of asynchrony.

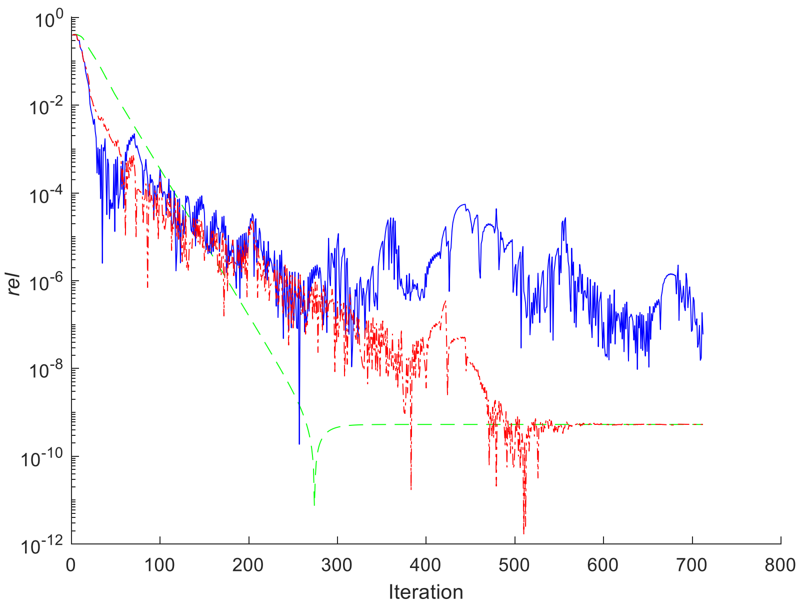


Figure 5.10 rel over iterations for the 48-bus system under a very high level of asynchrony.

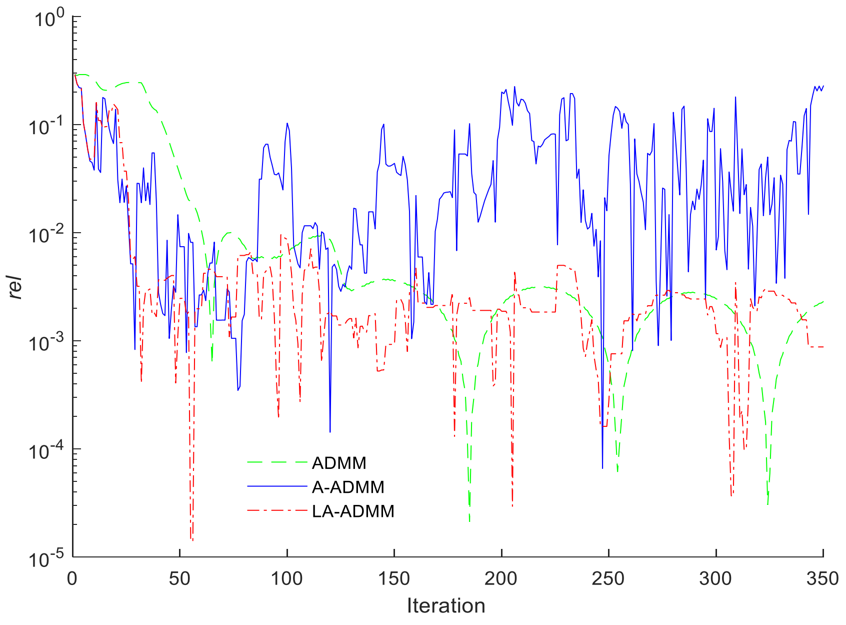


Figure 5.11 rel over iterations for the 118-bus system under a very high level of asynchrony.

5.6 Conclusion

A learning-based asynchronous ADMM is proposed to solve the OPF problem in a distributed manner. The aim is to address some limitations of synchronous and asynchronous ADMM. The proposed LA-ADMM has a momentum-extrapolation based prediction-correction step and an anomaly bypass loop. While the prediction-correction step estimates the values of missing share variables, the anomaly switch controller prevents prediction error propagation on Lagrange multipliers.

Various tests are implemented to solve OPF for three test systems and to analyze the performance of the proposed LA-ADMM. The results show that LA-ADMM outperforms synchronous and asynchronous parallel ADMM strategies, especially if the computational burden of subproblems are heterogeneous or missing information is observed in several consecutive iterations.

CHAPTER 6. CONCLUDING REMARKS AND FUTURE WORKS

6.1 Concluding remark

In this dissertation, several distributed optimization algorithms and techniques are proposed to solve the OPF problem in a distributed manner. In Chapter II, we present a decentralized OPF algorithm for the collaborative management of transmission and distribution systems. An ATC-based multilevel hierarchical distributed optimization is proposed in which TSO’s OPF is in the upper level of hierarchy and OPFs of DSOs are in the lower level. DC OPF is used for TSO, and AC OPF is used for DSOs. TSO and DSOs exchange only limited information of boundary buses (i.e., voltage angles of transmission-side buses and voltage angles and magnitudes of distribution-side buses) and are not required to reveal their commercially sensitive information to other parties. Two coordination strategies, namely DQA and TDQA, are presented to coordinate TSO and DSOs in a decentralized manner. DQA and TDQA solve local OPF problems associated with TSO and DSOs in parallel, which is more computationally efficient than sequential coordination strategies.

In Chapter III, we present an accelerated and robust ATC and its application for solving OPF. The proposed algorithm enhances the convergence performance of ATC and decreases its sensitivity to initialization and imbalance of cost terms in optimization objective functions by incorporating a balancing coefficient in objective functions of subproblems. The balancing coefficient is calculated at the end of the first iteration. The simulations show that the proposed AR-ATC with ALAD and EPF coordination strategies provides promising results for DC OPF and AC OPF. We have also tested the proposed technique on APP and ADMM and found that their performance is enhanced after incorporating the balancing coefficient. The best performance of AR-ATC is obtained when multipliers are set to moderate values.

In Chapter IV, we propose momentum-extrapolation-based asynchronous ADMM. A prediction-correction approach is designed to predict missing information of shared variables within a neighbourhood of the most updated values of these variables. Our proposed A-ADMM reduces the unproductive time of computational resources significantly in comparison with classical parallel ADMM. Simulation results on several test cases show the promising performance of the proposed algorithm for solving DC OPF and AC OPF. It is illustrated that A-ADMM outperforms ADMM, in particular, if the sizes of subproblems are heterogeneous.

In Chapter V, A learning-based asynchronous ADMM is proposed to address some limitations of A-ADMM developed in Chapter IV. The proposed LA-ADMM has an anomaly bypass loop. Although the prediction-correction step in A-ADMM predicts the missing values of share variables, it may have a large error after consecutive prediction. The anomaly switch controller prevents prediction error propagation on Lagrange multipliers. Various tests are implemented to solve OPF for three test systems and to analyze the performance of the proposed LA-ADMM. The results show that LA-ADMM outperforms synchronous and asynchronous parallel ADMM strategies, especially if the computational burden of subproblems are heterogeneous or missing information is observed in several consecutive iterations.

6.2 Future Work

The following suggestions are to extend the proposed distributed algorithms and techniques.

- Finding an optimal value for the balancing coefficient to further enhance the convergence performance of distributed OPF
- Implementing anomaly detection to enhance the performance of DQA

- Extending the proposed asynchronous learning-based ADMM by adding time series and machine learning techniques for predicting shared variable
- Using a convex version of AC OPF, e.g., second-order cone programming, in the proposed asynchronous technique
- Modeling uncertainties of renewable and load and outages of components in distributed OPF
- Using graph clustering or learning approaches to partition a power system to achieve a high convergence rate of distributed algorithms when applied to OPF
- Finding the best reference bus location to improve convergence performance of distributed OPF
- Extending the proposed algorithms and techniques to nonconvex and mixed-integer problems, such as unit commitment

REFERENCES

Abraham, M. P. and A. A. Kulkarni (2018). "ADMM-based algorithm for solving DC-OPF in a large electricity network considering transmission losses." *IET Generation, Transmission & Distribution* **12**(21): 5811-5823.

Ahmadi-Khatir, A., A. J. Conejo and R. Cherkaoui (2013). "Multi-area energy and reserve dispatch under wind uncertainty and equipment failures." *IEEE Transactions on Power Systems* **28**(4): 4373-4383.

Ahmadi-Khatir, A., A. J. Conejo and R. Cherkaoui (2013). "Multi-area unit scheduling and reserve allocation under wind power uncertainty." *IEEE Transactions on power systems* **29**(4): 1701-1710.

Amini, M. H., S. Bahrami, F. Kamyab, S. Mishra, R. Jaddivada, K. Boroojeni, P. Weng and Y. Xu (2018). Decomposition methods for distributed optimal power flow: Panorama and case studies of the DC model. *Classical and Recent Aspects of Power System Optimization*, Elsevier**:** 137-155.

Anscombe, F. J. (1948). "The validity of comparative experiments." *Journal of the royal statistical society. series A (General)* **111**(3): 181-211.

Aravena, I. and A. Papavasiliou (2015). *A distributed asynchronous algorithm for the two-stage stochastic unit commitment problem*. 2015 IEEE Power & Energy Society General Meeting, IEEE.

Ardakani, A. J. and F. Bouffard (2018). *Prediction of Umbrella Constraints.* 2018 Power Systems Computation Conference (PSCC), IEEE.

Bahrami, S. and M. H. Amini (2018). "A decentralized trading algorithm for an electricity market with generation uncertainty." *Applied Energy* **218**: 520-532.

Baker, K. and A. Bernstein (2019). "Joint Chance Constraints in AC Optimal Power Flow: Improving Bounds through Learning." *IEEE Transactions on Smart Grid.*

Baldick, R., B. H. Kim, C. Chase and Y. Luo (1999). "A fast distributed implementation of optimal power flow." *IEEE Transactions on Power Systems* **14**(3): 858-864.

Baroche, T., P. Pinson, R. L. G. Latimier and H. B. Ahmed (2019). "Exogenous cost allocation in peer-to-peer electricity markets." *IEEE Transactions on Power Systems* **34**(4): 2553-2564.

Bertsekas, D. P. (1999). *Nonlinear programming,* Athena scientific Belmont.

Bertsekas, D. P. (2005). "Dynamic programming and suboptimal control: A survey from ADP to MPC." *European Journal of Control* **11**(4): 310-334.

Bertsekas, D. P. (2014). *Constrained optimization and Lagrange multiplier methods*, Academic press.

Boyd, S., N. Parikh, E. Chu, B. Peleato and J. Eckstein (2011). "Distributed optimization and statistical learning via the alternating direction method of multipliers." *Foundations and Trends® in Machine Learning* **3**(1): 1-122.

Chang, T.-H., M. Hong, W.-C. Liao and X. Wang (2016). "Asynchronous distributed ADMM for large-scale optimization—Part I: Algorithm and convergence analysis." *IEEE Transactions on Signal Processing* **64**(12): 3118-3130.

Chang, X., Y. Xu, H. Sun and I. Khan "A distributed robust optimization approach for the economic dispatch of flexible resources." *International Journal of Electrical Power & Energy Systems* **124**: 106360.

Cherukuri, A. and J. Cortes (2016). "Initialization-free distributed coordination for economic dispatch under varying loads and generator commitment." *Automatica* **74**: 183-193.

Cohen, G. (1980). "Auxiliary problem principle and decomposition of optimization problems." *Journal of optimization Theory and Applications* **32**(3): 277-305.

Conejo, A. J., E. Castillo, R. Minguez and R. Garcia-Bertrand (2006). *Decomposition Techniques in Mathematical Programming*, Springer.

Costley, M. and S. Grijalva (2012). *Efficient distributed OPF for decentralized power system operations and electricity markets.* Innovative Smart Grid Technologies (ISGT), 2012 IEEE PES, IEEE.

Dall'Anese, E., H. Zhu and G. B. Giannakis (2013). "Distributed optimal power flow for smart microgrids." *IEEE Transactions on Smart Grid* **4**(3): 1464-1475.

DorMohammadi, S. and M. Rais-Rohani (2012). *Analytical target cascading framework using the exponential method of multipliers*. 12th AIAA conference on aviation technology, integration and operations.

DorMohammadi, S. and M. Rais-Rohani (2012). "Comparison of alternative strategies for multilevel optimization of hierarchical systems." *Applied Mathematics* **3**(10A): 1448-1462.

DorMohammadi, S. and M. Rais-Rohani (2013). "Exponential penalty function formulation for multilevel optimization using the analytical target cascading framework." *Structural and Multidisciplinary Optimization* **47**(4): 599-612.

Dwork, C., N. Lynch and L. Stockmeyer (1988). "Consensus in the presence of partial synchrony." *Journal of the ACM (JACM)* **35**(2): 288-323.

EE236C, L. V. "6. Proximal gradient method."

Engelmann, A., Y. Jiang, T. Mühlpfordt, B. Houska and T. Faulwasser (2019). "Toward Distributed OPF Using ALADIN." *IEEE Transactions on Power Systems* **34**(1): 584-594.

Erseghe, T. (2014). "Distributed optimal power flow using ADMM." *IEEE Transactions on Power Systems* **29**(5): 2370-2380.

Erseghe, T. (2015). "A distributed approach to the OPF problem." *EURASIP Journal on Advances in Signal Processing* **2015**(1): 45.

Ferrante, A., N. Constantinescu and J. A. Jackson (2015). "Lines of Convergence: R&D for Transmission and Distribution: Coordination and the Regulatory Challenge." *IEEE Power and Energy Magazine* **13**(1): 52-59.

Goldstein, T., B. O'Donoghue, S. Setzer and R. Baraniuk (2014). "Fast alternating direction optimization methods." *SIAM Journal on Imaging Sciences* **7**(3): 1588-1623.

Guo, J., G. Hug and O. Tonguz (2017). "Asynchronous ADMM for Distributed Non-Convex Optimization in Power Systems." *arXiv preprint arXiv:1710.08938.*

Guo, J., G. Hug and O. K. Tonguz (2017). "A case for nonconvex distributed optimization in large-scale power systems." *IEEE Transactions on Power Systems* **32**(5): 3842-3851.

Hestenes, M. R. (1969). "Multiplier and gradient methods." *Journal of optimization theory and applications* **4**(5): 303-320.

Huang, S., Y. Chen and C. Shen (2008). *Distributed OPF of larges scale interaction power systems.* Power System Technology and IEEE Power India Conference, 2008. POWERCON 2008. Joint International Conference on, IEEE.

Hug-Glanzmann, G. and G. Andersson (2009). "Decentralized optimal power flow control for overlapping areas in power systems." *IEEE Transactions on Power Systems* **24**(1): 327-336.

Hur, D., J.-K. Park and B. H. Kim (2002). "Evaluation of convergence rate in the auxiliary problem principle for distributed optimal power flow." *IEE Proceedings-Generation, Transmission and Distribution* **149**(5): 525-532.

Kar, S., G. Hug, J. Mohammadi and J. M. Moura (2014). "Distributed State Estimation and Energy Management in Smart Grids: A Consensus ${+} $ Innovations Approach." *IEEE Journal of Selected Topics in Signal Processing* **8**(6): 1022-1038.

Karagiannopoulos, S., P. Aristidou and G. Hug (2019). "Data-driven Local Control Design for Active Distribution Grids using off-line Optimal Power Flow and Machine Learning Techniques." *IEEE Transactions on Smart Grid.*

Kargarian, A. and Y. Fu (2014). "System of systems based security-constrained unit commitment incorporating active distribution grids." *IEEE Transactions on Power Systems* **29**(5): 2489-2498.

Kargarian, A., Y. Fu and Z. Li (2015). "Distributed security-constrained unit commitment for large-scale power systems." *IEEE Transactions on Power Systems* **30**(4): 1925-1936.

Kargarian, A., Y. Fu and H. Wu (2016). "Chance-Constrained System of Systems Based Operation of Power Systems." *IEEE Transactions on Power Systems* **31**(5): 3404-3413.

Kargarian, A., M. Mehrtash and B. Falahati (2018). "Decentralized implementation of unit commitment with analytical target cascading: A parallel approach." *IEEE Transactions on Power Systems* **33**(4): 3981-3993.

Kargarian, A., J. Mohammadi, J. Guo, S. Chakrabarti, M. Barati, G. Hug, S. Kar and R. Baldick (2016). "Toward distributed/decentralized DC optimal power flow implementation in future electric power systems." *IEEE Transactions on Smart Grid* **9**(4): 2574-2594.

Kim, B. H. and R. Baldick (1997). "Coarse-grained distributed optimal power flow." IEEE *Transactions on Power Systems* **12**(2): 932-939.

Kim, B. H. and R. Baldick (2000). "A comparison of distributed optimal power flow algorithms." *IEEE Transactions on Power Systems* **15**(2): 599-604.

Kocuk, B., S. S. Dey and X. A. Sun (2016). "Strong SOCP relaxations for the optimal power flow problem." *Operations Research* **64**(6): 1177-1196.

Kristov, L., P. De Martini and J. D. Taft (2016). "A Tale of Two Visions: Designing a Decentralized Transactive Electric System." *IEEE Power and Energy Magazine* **14**(3): 63-69.

Kumar, S., R. Jain and K. Rajawat (2016). "Asynchronous optimization over heterogeneous networks via consensus admm." *IEEE Transactions on Signal and Information Processing over Networks* **3**(1): 114-129.

Lavaei, J. and S. H. Low (2012). "Zero duality gap in optimal power flow problem." *IEEE Transactions on Power Systems* **27**(1): 92-107.

Li, Y., Z. Lu and J. J. Michalek (2008). "Diagonal quadratic approximation for parallelization of analytical target cascading." *Journal of Mechanical Design* **130**(5): 051402.

Li, Z., Q. Guo, H. Sun and J. Wang (2016). "Coordinated economic dispatch of coupled transmission and distribution systems using heterogeneous decomposition." *IEEE Transactions on Power Systems* **31**(6): 4817-4830.

Li, Z., Q. Guo, H. Sun and J. Wang (2016). "Coordinated transmission and distribution AC optimal power flow." *IEEE Transactions on Smart Grid.*

Liu, J., M. Benosman and A. Raghunathan (2015). *Consensus-based distributed optimal power flow algorithm*. Innovative Smart Grid Technologies Conference (ISGT), 2015 IEEE Power & Energy Society, IEEE.

Lu, W., M. Liu, S. Lin and L. Li (2018). "Fully decentralized optimal power flow of multi-area interconnected power systems based on distributed interior point method." *IEEE Transactions on Power Systems* **33**(1): 901-910.

Malekpour, A. R. and A. Pahwa (2017). "Stochastic networked microgrid energy management with correlated wind generators." *IEEE Transactions on Power Systems* **32**(5): 3681-3693.

Malekpour, A. R., A. Pahwa and B. Natarajan (2016). "Hierarchical architecture for integration of rooftop PV in smart distribution systems." *IEEE Transactions on Smart Grid* **9**(3): 2019-2029.

Malhotra, A., G. Binetti, A. Davoudi and I. D. Schizas (2017). "Distributed power profile tracking for heterogeneous charging of electric vehicles." *IEEE Transactions on Smart Grid* **8**(5): 2090-2099.

Manshadi, S. D. and M. E. Khodayar (2016). "A hierarchical electricity market structure for the smart grid paradigm." *IEEE Transactions on Smart Grid* **7**(4): 1866-1875.

Marvasti, A. K., Y. Fu, S. DorMohammadi and M. Rais-Rohani (2014). "Optimal operation of active distribution grids: A system of systems framework." *IEEE Transactions on Smart Grid* **5**(3): 1228-1237.

Mbuwir, B. V., F. Spiessens and G. Deconinck (2020). "Distributed optimization for scheduling energy flows in community microgrids." *Electric Power Systems Research* **187**: 106479.

Mehrtash, M., A. Kargarian and A. Mohammadi (2019). "Distributed optimisation-based collaborative security-constrained transmission expansion planning for multi-regional systems." *IET Generation, Transmission & Distribution* **13**(13): 2819-2827.

Mhanna, S., G. Verbič and A. C. Chapman (2018). *Accelerated Methods for the SOCP-relaxed Component-based Distributed Optimal Power Flow*. 2018 Power Systems Computation Conference (PSCC), IEEE.

Michelena, N., H. Park and P. Y. Papalambros (2003). "Convergence properties of analytical target cascading." *AIAA journal* **41**(5): 897-905.

Mohammadi, A. and A. Kargarian (2020). "Accelerated and Robust Analytical Target Cascading for Distributed Optimal Power Flow." *IEEE Transactions on Industrial Informatics.*

Mohammadi, A., M. Mehrtash and A. Kargarian (2018). "Diagonal quadratic approximation for decentralized collaborative TSO+ DSO optimal power flow." *IEEE Transactions on Smart Grid* **10**(3): 2358-2370.

Mohri, M., A. Rostamizadeh and A. Talwalkar (2018). *Foundations of machine learning*, MIT press.

Molzahn, D. K., F. Dörfler, H. Sandberg, S. H. Low, S. Chakrabarti, R. Baldick and J. Lavaei (2017). "A survey of distributed optimization and control algorithms for electric power systems." *IEEE Transactions on Smart Grid* **8**(6): 2941-2962.

Momentum Extrapolation Prediction-Based Asynchronous Distributed Optimization for Power Systems (2020). from https://sites.google.com/site/aminkargarian/test-system-data/momentum-extrapolation-a-admm.

Nesterov, Y. (1983). "A method of solving a convex programming problem with convergence rate o(1=k2)." *Soviet Math. Dokl.* **27**: 372-376.

Nguyen, H. K., H. Mohsenian-Rad, A. Khodaei and Z. Han (2017). "Decentralized reactive power compensation using nash bargaining solution." *IEEE Transactions on Smart Grid* **8**(4): 1679-1688.

Palomar, D. P. and M. Chiang (2006). "A tutorial on decomposition methods for network utility maximization." *IEEE Journal on Selected Areas in Communications* **24**(8): 1439-1451.

Peng, Z., Y. Xu, M. Yan and W. Yin (2016). "Arock: an algorithmic framework for asynchronous parallel coordinate updates." *SIAM Journal on Scientific Computing* **38**(5): A2851-A2879.

Ramanan, P., M. Yildirim, E. Chow and N. Gebraeel (2019). "An Asynchronous, Decentralized Solution Framework for the Large Scale Unit Commitment Problem." *IEEE Transactions on Power Systems.*

Ruszczyński, A. (1995). "On convergence of an augmented Lagrangian decomposition method for sparse convex optimization." *Mathematics of Operations Research* **20**(3): 634-656.

Sun, H., Q. Guo, B. Zhang, Y. Guo, Z. Li and J. Wang (2015). "Master–slave-splitting based distributed global power flow method for integrated transmission and distribution analysis." *IEEE Transactions on Smart Grid* **6**(3): 1484-1492.

Tosserams, S., L. Etman, P. Papalambros and J. Rooda (2006). "An augmented Lagrangian relaxation for analytical target cascading using the alternating direction method of multipliers." *Structural and multidisciplinary optimization* **31**(3): 176-189.

Wang, Y., S. Wang and L. Wu (2017). "Distributed optimization approaches for emerging power systems operation: A review." *Electric Power Systems Research* **144**: 127-135.

Wang, Y., L. Wu and S. Wang (2016). "A fully-decentralized consensus-based ADMM approach for DC-OPF with demand response." *IEEE Transactions on Smart Grid* **8**(6): 2637-2647.

Yi, P., Y. Hong and F. Liu (2016). "Initialization-free distributed algorithms for optimal resource allocation with feasibility constraints and application to economic dispatch of power systems." *Automatica* **74**: 259-269.

Zhang, L., J. Zhao and W. Li (2019). "Online and Unsupervised Anomaly Detection for Streaming Data Using an Array of Sliding Windows and PDDs." *IEEE Transactions on Cybernetics.*

Zhang, Q., K. Dehghanpour and Z. Wang (2018). "Distributed CVR in Unbalanced Distribution Systems with PV Penetration." *IEEE Transactions on Smart Grid.*

Zhang, R. and J. Kwok (2014). *Asynchronous distributed ADMM for consensus optimization.* International Conference on Machine Learning.

Zimmerman, R. D., C. E. Murillo-Sánchez and R. J. Thomas (2011). "MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education." *IEEE Transactions on power systems* **26**(1): 12-19.

VITA

Ali Mohammadi received his bachelor’s degree of Electrical Engineering (Telecommunication) from Shahid Bahonar university of Kerman, Kerman, Iran, in 2012, and his MS degree in Electrical Engineering (Telecommunication) from Shiraz University of Technology, Shiraz, Iran, in 2014. He is currently pursuing his Ph.D. degree at the Department of Electrical and Computer Engineering, Louisiana State University, LA, USA. His research interests include power system optimization, machine learning, and distributed optimization.