Distributed resource allocation algorithm design and its application to economic dispatch in smart grids

Bai, Lu

2019

Bai, L. (2019). Distributed resource allocation algorithm design and its application to economic dispatch in smart grids. Master's thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/136854>

<https://doi.org/10.32657/10356/136854>

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

Downloaded on 26 Apr 2023 07:39:58 SGT

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

SINGAPORE

**DISTRIBUTED RESOURCE ALLOCATION
ALGORITHM DESIGN AND ITS
APPLICATION TO ECONOMIC DISPATCH IN
SMART GRIDS**

BAI LU

SCHOOL OF ELECTRICAL & ELECTRONIC ENGINEERING

2019

**DISTRIBUTED RESOURCE ALLOCATION
ALGORITHM DESIGN AND ITS
APPLICATION TO ECONOMIC DISPATCH IN
SMART GRIDS**

BAI LU

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirements for the degree of
Doctor of Philosophy

2019

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

1 Jan 2020

.....

Date

Bai Lu

.....

BAI LU

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

7 January 2020

.....

Date

.....

HU GUOQIANG

Authorship Attribution Statement

This thesis contains material from 3 papers published in the following peer-reviewed journals and conferences where I was the first author.

Chapter 3 and Chapter 4 are published as

L. Bai, M. Ye, C. Sun, and G. Hu, “Distributed Economic Dispatch Control via Saddle Point Dynamics and Consensus Algorithm”, *IEEE Transactions on Control Systems Technology*, vol. 27, no. 2, pp. 898-905, 2019.

L. Bai, M. Ye, C. Sun, and G. Hu, “Distributed Control for Optimal Economic Dispatch via Saddle Point Dynamics and Consensus Algorithms”, *IEEE Conference on Decision and Control (CDC)*, Las Vegas, NV, USA, pp. 6934-6393, Dec 2016.

The contributions of the co-authors are as follows:

- Prof. Hu supervised the project development, provided guidance during the project, and reviewed the manuscript drafts.
- I prepared the manuscript drafts.
- I performed the theoretical analysis and conducted the simulation.
- Dr. Ye and Dr. Sun assisted in providing guidance in related theories.

Chapter 5 is published as

L. Bai, C. Sun, Z. Feng, and G. Hu, “Distributed Continuous-time Resource Allocation with Time-varying Resources under Quadratic Cost Functions”, *IEEE Conference on Decision and Control (CDC)*, Miami Beach, FL, USA, Dec 2018.

The contributions of the co-authors are as follows:

- Prof. Hu supervised the project development, provided guidance during the project, and reviewed the manuscript drafts.
- I prepared the manuscript drafts.
- I performed the theoretical analysis and conducted the simulation.
- Together with Dr. Sun and Dr. Feng, we checked the theoretical analysis and improved the results.

1 Jan 2020

.....

Date

Bai Lu

.....

BAI LU

Acknowledgments

First and foremost, I would like to express my special and great appreciation to my supervisor, Prof. Guoqiang Hu, for his invaluable guidance and continuous support throughout my Ph.D study. His insightful suggestions and generous help have enlightened my research journey. Under his guidance, my thinking towards problems becomes deeper and broader. I also learned a lot from his high standard and rigorous attitude toward research, which will benefit me in all my life.

My sincere thanks to my lab mates for their friendship, helpful discussions, and encouragement. Thanks to my friends who play badminton and do sports together with me at NTU. They made my study and life more meaningful and enjoyable.

Last but not least, special thanks to my beloved parents and sister, for their endless love, encouragement, and support all the time.

Abstract

In nowadays, there is an increasing number of networked systems and the scale of which is becoming larger and larger. To improve the efficacy and safety of the large-scale networked system, distributed methods come to the fore as the time requires. The resource allocation (RA) problem is one of the most important problems in networked systems. It addresses how to economically assign the available resources to a set of users such that the overall objective is optimized. The RA problem has various applications, such as portfolio selection, production planning, queuing control, and economic dispatch (ED) in power systems. As a typical example of RA, ED produces optimal generation references for the distributed energy resources, such that the total generation cost is minimized under the power balance and generation capacity constraints. In this thesis, a basic RA problem with separable convex objective function and equality resource constraint is investigated. We focus on developing initialization-free distributed algorithms to solve the basic RA problem through a multi-agent framework. The algorithms developed mainly have their basis on consensus algorithms in multi-agent systems and gradient descent methods in optimization theory.

This thesis starts with the static RA problem where the optimal solution is fixed. Distributed approaches based on consensus protocols and the saddle point dynamics (SPD) are proposed to solve the RA problem under quadratic objective functions. By utilizing Lyapunov stability analysis and singular perturbation analysis, the stability of the proposed algorithms is proven. No global information is needed

in the proposed method and the requirement on initial conditions of the variables is mild. The algorithms are then extended to solve a more general RA problem where the objective functions are strictly convex instead of quadratic. Then, the ED problem in smart grids is explored. In addition to solving the basic ED problem, the algorithms are further extended to solve the ED problem with line flow limits. Simulations on the IEEE 9-bus system and the IEEE 118-bus system are conducted to verify the algorithms, demonstration platform based on Matlab GUI is developed to demonstrate the algorithms.

Noting that the resources in many problems exhibit time-varying nature, the RA problem with continuously time-varying resources (TVR) is then investigated. Since the resources are time-varying, the optimal solution changes over time. Thus, the algorithm should not only find the optimal solution but also have the ability to track the time-varying optimal solution trajectory. When the local allocation feasibility constraint is not considered and the objective function is quadratic, a distributed algorithm based on sign function and consensus algorithms is proposed. The tracking error is proven to converge to zero in finite time. With the local allocation feasibility constraint, a distributed algorithm based on singular perturbation theory and penalty function is proposed. The tracking error is proven to be uniformly ultimately bounded. The algorithms are applied to the ED problem with continuously time-varying loads. Simulation on the 24-bus system is conducted to illustrate the algorithms.

Considering the valve-point effect and multiple fuel options of the generators, the last part of this thesis proposes a distributed algorithm to solve the practical ED problem. The cost functions of the practical ED problem is non-convex and non-smooth. The Gaussian smoothing is used to approximate the non-smooth cost functions. Based on the approximated smoothed cost functions, randomized gradient and consensus algorithms are utilized to develop a distributed algorithm to solve the practical ED problem.

To summary, this thesis solves the RA problem in a distributed manner through a multi-agent framework. Two cases where the resources are static and time-varying are investigated, respectively. To demonstrate the practical relevance of our proposed algorithms, the ED problem in smart grids is investigated. Various simulation case studies are conducted to verify the algorithms. In addition, a randomized gradient and consensus based distributed algorithm is developed to solve the practical non-convex ED problem.

Symbols and Acronyms

Algebraic Operators

A^T	Transpose of matrix A
$\text{diag}\{a_i\}$	Diagonal matrix with i -th diagonal element being a_i
$[x_i]_{\text{vec}}$	Column vector with i -th element being x_i
$\mathbf{0}_n$	Column vector with n zeros
$\mathbf{1}_n$	Column vector with n ones
I_n	$n \times n$ identity matrix
λ^H	Conjugate transpose of λ
$\otimes$	Kronecker product
$\lambda_{\min}(A)$	Smallest eigenvalue of matrix A
sgn	Sign function
$\mathbb{E}$	Expectation

Sets

$\mathbb{Z}^+$	Set of positive integers
$\mathbb{R}$	Set of real numbers
$\mathbb{R}^n$	Set of n -dimensional real column vectors
$\mathbb{R}^{m \times n}$	Set of $m \times n$ real matrices

Acronyms

ED	Economic Dispatch
RA	Resource Allocation
SPD	Saddle Point Dynamics
KKT	Karush - Kuhn - Tucker
TVR	Time-varying Resources
TVO	Time-varying Optimization

Contents

Statement of Originality	ii
Supervisor Declaration Statement	iv
Authorship Attribution Statement	vii
Acknowledgments	viii
Abstract	xii
Symbols and Acronyms	xv
List of Contents	xix
List of Figures	xxiv
List of Tables	xxv
1 Introduction	1
1.1 Background, Motivation, and Objectives	1

1.2	Literature Review	4
1.2.1	Distributed Resource Allocation	5
1.2.2	Distributed Economic Dispatch	8
1.2.3	Distributed Time-varying Optimization	12
1.3	Contributions	13
1.4	Organization of the Thesis	15
2	Preliminaries	17
2.1	Graph Theory and Consensus Algorithms	17
2.2	Convex Optimization	19
2.2.1	Convex Optimality Condition	19
2.2.2	Karush-Kuhn-Tucker Conditions	19
2.2.3	Saddle Point Dynamics	20
2.3	Gaussian Smoothing, Randomized Gradient, and Projection	21
3	Distributed Resource Allocation via Saddle Point Dynamics and Consensus Algorithms	23
3.1	Introduction	23
3.2	Problem Formulation	26
3.3	Distributed RA with Quadratic Objective Functions	29
3.3.1	Distributed RA without Local Allocation Feasibility Constraint	29

3.3.2	Distributed RA with Local Allocation Feasibility Constraint	34
3.4	Distributed RA with Strictly Convex Objective Functions and Local Convex Inequality Constraint	40
3.5	Numerical Examples	42
3.6	Conclusion	46
4	Distributed Economic Dispatch in Smart Grids	49
4.1	Introduction	49
4.2	System Model and Problem Formulation	51
4.3	Distributed ED	53
4.3.1	Distributed Algorithm for the Basic ED Problem	53
4.3.2	Distributed Algorithm for the ED Problem with Line Flow Limits	54
4.4	Case Studies	57
4.4.1	Implementation on the IEEE 9-bus Test System	57
4.4.2	Implementation on the IEEE 118-bus Test System	68
4.4.3	Demonstration through Matlab GUI	69
4.5	Conclusion	70
5	Distributed Time-varying Resource Allocation and Its Application to Economic Dispatch with Time-varying Loads	71
5.1	Introduction	71

5.2	Problem Formulation	73
5.3	Distributed RA with Time-varying Resources	75
5.3.1	Without Local Allocation Feasibility Constraint under Quadratic Objective Functions	75
5.3.2	With Local Allocation Feasibility Constraint under Strongly Convex Objective Functions	82
5.3.3	Numerical Examples	94
5.4	Distributed ED with Time-varying Loads	100
5.4.1	System Model and Problem Formulation	100
5.4.2	Distributed Algorithm	101
5.4.3	Simulation on the 24-bus Test System	102
5.5	Conclusion	107
6	Distributed Practical Non-Convex Economic Dispatch	109
6.1	Introduction	109
6.2	Problem Formulation	110
6.3	Distributed Algorithm	112
6.3.1	Gaussian Smoothing and Randomized Gradient	112
6.3.2	Distributed Algorithm	114
6.4	Simulation	117
6.4.1	Numerical Example	117

6.4.2	Simulation on a Ten Generating Units System	120
6.5	Conclusion	121
7	Conclusions and Future Works	127
7.1	Conclusions	127
7.2	Recommendations for Further Works	129
	Author's Publications	133
	Bibliography	152

List of Figures

1.1	Structure of power system: (a) centralized structure with a central power generating unit; (b) distributed structure with distributed generating units and loads.	2
1.2	Comparison between centralized and distributed control architecture.	3
1.3	Hierarchical control in microgrid.	8
2.1	Saddle point (0,0) of function $S(x, y) = x^2 - y^2$	20
3.1	Communication topology among the five modules.	43
3.2	Evolution of the allocated time t without positivity constraint.	44
3.3	Evolution of λ and z without positivity constraint.	45
3.4	Evolution of the allocated time t with positivity constraint.	46
3.5	Evolution of λ and z with positivity constraint.	47
4.1	Structure of a virtual agent composed of local generator, load, computation unit, and communication unit.	51
4.2	Illustration of the distributed control architecture.	51

4.3	The Schematic outline of algorithm (4.3).	55
4.4	Illustration of the IEEE 9-bus system.	57
4.5	Communication topology of the IEEE 9-bus system.	58
4.6	Power generations of the generators and estimated incremental costs without capacity limits produced by (4.2). The dashed lines denote the optimal values.	59
4.7	Power generations and estimated incremental costs with capacity limits produced by (4.3). The dashed lines denote the optimal values.	60
4.8	Power generations and estimated incremental costs produced by (4.3) when capacity limits are contradicted. The dashed lines denote the new optimal values.	61
4.9	Power generations produced by (a) (Yang 2013), (b) (Xing 2015), (c) (Li 2015), (d) (Zhang 2016), (e) (Kar 2012), and (f) (Wang 2019). The solid lines represent the results when the initial conditions are satisfied, the dashed lines represent the results when the initial conditions are not satisfied.	62
4.10	Power generations and estimated incremental costs produced by (4.3) with load changes.	63
4.11	Power generation produced by (4.3) with non-quadratic cost function and fixed generation.	64
4.12	Power generation produced by (4.3) with one communication link loss.	65
4.13	Communication topology with one generator plugged-in.	65
4.14	Power generation produced by (4.3) with one generator plugged-in.	66

4.15	Power generations and power flows produced by (4.5) with line flow limits.	67
4.16	Power generations and estimated incremental costs produced by (4.3) tested on the IEEE 118-bus system.	68
4.17	User interface of the distributed ED developed by Matlab GUI. . . .	69
5.1	Communication graph for the five agents.	95
5.2	Evolution of x without local allocation feasibility constraint produced by (5.8)-(5.9). The upper figure shows that the decision variables track the optimal allocation of resource 1, and the lower figure shows that the decision variables track the optimal allocation of resource 2.	96
5.3	Evolution of λ without local allocation feasibility constraint produced by (5.8)-(5.9).	97
5.4	Evolution of x with local allocation feasibility constraint produced by (5.13). The upper figure shows that the decision variables track the optimal allocation of resource 1, and the lower figure shows that the decision variables track the optimal allocation of resource 2.	98
5.5	Evolution of λ with local allocation feasibility constraint produced by (5.13).	99
5.6	24-bus system and control areas. Gi represents generating unit, Li represents load, and Ai represents control area.	103
5.7	Continuously time-varying system load by spline interpolation. . . .	105
5.8	Communication topology among the nine control areas.	105

5.9	Power generation and evolution of λ without capacity limits produced by (5.19).	106
5.10	Power generation and evolution of λ with capacity limits produced by (5.20).	107
5.11	Power generation and evolution of λ with capacity limits under non-quadratic cost function produced by (5.20).	108
6.1	Flowchart of Algorithm 2.	123
6.2	Cost function curve of the numerical example.	124
6.3	Communication topology of the three generating units.	124
6.4	Updates of average values of P_{Gi} , z_i , and λ_i under different total load of the numerical example.	125
6.5	Communication topology of the ten generating units.	125
6.6	Generation P_G and z produced by Algorithm 2.	126

List of Tables

3.1	Parameters and optimal solution for the software-testing RA problem in [125]	43
4.1	Generator Parameters	58
5.1	Composition and capacity limit of each control area	103
5.2	24-hour hourly total system load profile from [137]	104
5.3	Control area distribution of the total system load from [137]	104
6.1	Cost function parameters of the numerical example	119
6.2	Comparison of the numerical example results of Algorithm 2 and Genetic Algorithm	119
6.3	Parameters of the cost function [43]	120
6.4	Comparison of the simulation results on the 10 generating units system	121

Chapter 1

Introduction

1.1 Background, Motivation, and Objectives

The resource allocation (RA) problem addresses how to economically assign the available resources to a set of users such that the total objective is optimized. It has diverse applications, such as portfolio selection in economics [1], batching policies for video-on-demand storage services in computer science [2], allocation of software-testing resources in reliability applications [3], economic dispatch (ED) in power systems [4], and so on [5, 6]. The basic RA problem has separable convex objective function and equality resource constraint. The importance of the RA problem not only lies in its economic efficiency, but also it is essential for the system to operate within the resource constraint. Such as in the power system, the mismatch between power generation and consumption will cause the frequency to deviate from the nominal value, and thus may damage synchronous machines and other appliances. As a typical example of the RA problem, the ED problem considers how to economically allocate the load demand to a group of generating units, such that the total generation cost is minimized under the power balance and generation capacity constraints. With ED, not only the generation requirements, which are the total

generation meets the total demand and the generation of each generating unit is within its capacity limit, are satisfied, but also the generation cost can be significantly reduced [7].

Nowadays, there is a rapid growth in the scale of the networked systems. Take the power system as an example, with the development of distributed generating units and the increase of the demand for electrical energy, the power system is transforming from the traditional centralized power system to a more distributed smart grid [8]. As shown in Figure. 1.1, in the centralized power system, there

Figure 1.1: Structure of power system: (a) centralized structure with a central power generating unit; (b) distributed structure with distributed generating units and loads.

usually has a large and distant central generating unit, which provides power to the customers through high voltage transmission lines; while in the distributed power system, there are distributed small generating units work collaboratively to serve the total load and provide power to nearby loads directly. The growing number of distributed generating units and loads leads to a larger and more complex networked power system. Thus, the economic dispatch problem investigated in this thesis will focus on the smart grid, where advanced communication and control technologies are integrated into the system to achieve a more efficient, reliable, and safe system [9]. People address this problem from optimization, control, communication, and

power systems perspectives. The results in these studies as well as this thesis provide important guidance on the realization of the smart grid.

In the meantime, the evolution of communication and computing technologies enables the development of distributed control algorithms which are more suitable for large-scale networked systems. The comparison between centralized and distributed control architectures is depicted in Figure. 1.2. In a centralized architecture, the

Figure 1.2: Comparison between centralized and distributed control architecture.

central unit, which is responsible to collect the information from each node, usually requires high quality communication and calculation infrastructures to handle the system-wide gathered information, gives rise to reliability concerns due to the presence of a central node [10], and is liable to modeling error [11]. Moreover, the constitutions of many cooperative systems, such as the smart grid and mobile robotic networks, are varying from time to time, which would reduce the efficiency of a centralized architecture. Alternatively, supporting each node with a local controller and a communication infrastructure to communicate with its neighbors, distributed algorithms which are more robust, scalable, and immune to topological variations can be performed. In addition, lower cost is expected to achieve since short-range wireless communication devices can be used to implement the network infrastructure and low-cost small local controllers can be used to do local calculations in contrast to a high quality centralized unit [12]. From the above, designing distributed algorithms to solve the large-scale networked RA problem is of both theoretical significance and practical relevance.

Motivated by theoretical significance and practical importance of solving the RA problem in a distributed manner, the objectives of this thesis are detailed as follows.

1. Develop distributed algorithms to solve the basic RA problem. The RA problem is a networked problem, without a central unit, the aggregate resource and the aggregate allocation are not available. By utilizing the multi-agent framework, consensus algorithms can be adopted for the users to estimate the mismatch between total resource and total allocation by communicating with their neighbors. Using the estimated information, optimization algorithms can be designed to solve the basic RA problem in a distributed manner.
2. Develop distributed algorithms to find and track the optimal solution trajectory of the RA problem with time-varying resources (TVR). The resources often exhibit time-varying characteristics in reality, which causes the optimal solution of the RA problem varying from time to time. Tracking the time-varying optimal solution will improve the economic efficiency comparing to solving the RA problem at certain time instances treating the resources to be static between time instances.
3. Investigate the ED problem in smart grids. The smart grid is a large-scale networked system with distributed generating units and loads. The network is variable due to the flexibility of various generating units as well as loads. Designing distributed methods to handle the ED problem in smart grids can improve the flexibility and reliability of the system.

1.2 Literature Review

Related works on distributed RA, distributed ED, and time-varying optimization (TVO) are reviewed in this section.

1.2.1 Distributed Resource Allocation

The RA aims to economically assign the available resources to a set of users. It has been widely investigated for its diverse applications. A general form of the RA problem is [6].

$$\begin{aligned} & \text{minimize } f(x) \\ & \text{subject to } \sum_{i=1}^n x_i = N, \end{aligned}$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is the objective function, n is the number of users, $x = [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^n$ is the decision variable, and $N \in \mathbb{R}$ is the total amount of the resource. According to the types of the variables, the structures of the objective function, and the constraints, the RA can be divided into different kinds of forms [6], such as continuous or discrete RA problem [13], separable or coupled objective function which arises from Lagrangian relaxation of coupling constraints [14]. One important and basic form is to optimize a separable convex function under the resource constraint that the total allocation equals the total amount of the available resource. In this thesis, we mainly focus on investigating the continuous separable convex RA problem with differentiable objective function, i.e.,

$$\text{minimize } f(x) = \sum_{i=1}^n f_i(x_i) \tag{1.1a}$$

$$\text{subject to } \sum_{i=1}^n x_i = N, \tag{1.1b}$$

where x_i is continuous, $f_i(x_i)$ is convex and differentiable. In addition, the local allocation feasibility constraint will be considered, which is

$$x_i^{\min} \leq x_i \leq x_i^{\max}, \quad i = 1, \dots, n,$$

where $x_i^{\min}$ and $x_i^{\max}$ are the lower and upper bounds on x_i . In the following, we will restrict our review to continuous separable convex RA.

A variety of centralized approaches where a central unit is responsible for collecting all the network data, calculating the optimal allocation, and then sending back to the users have been proposed to solve the RA problem. Many methods like gradient method, Newton's method, and quasi-Newton method [15] can be utilized to solve the RA problem in a centralized manner. More details can refer to [16], which gave a comprehensive review of the methods that solve the problem. Different from the centralized approach, a special approach named decomposition method divides an optimization problem into subproblems, and the subproblems are coordinated by the solution of a master problem [17]. The decomposition method is divided into price-based [18] and resource-based [19], which are also called dual decomposition and primal decomposition, respectively [20], based on the master problem. However, a central coordinator is still required similar to the centralized approaches, which is inefficient for a distributed large-scale system and has the risk of single node failure.

More recently, to cope with the large-scale networks, distributed optimization has attracted much attention. In a distributed mechanism, agents cooperatively find optimal solution using only local information. In light of the seminal work on distributed RA in [21], various distributed algorithms have been developed, see [22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32] and the references therein. Discrete-time iterative center-free algorithms based on the weighted gradients of neighbors have been widely studied. [22] investigated how to choose the scaling factors for the gradients to make the algorithm converge fast. Asynchronous gradient-descent was proposed in [23], which is suitable for a dynamic communication topology. A random block coordinate descent method was developed in [24] for large-scale convex problems to reduce the numerical complexity, proper probabilities were derived to let the algorithm converge fast. Faster convergence rate algorithms were developed in [25] and [26], where linear convergence rate and super-linear convergence rate were achieved based on a distributed multi-step gradient and a distributed Newton's

algorithm, respectively. In the aforementioned algorithms based on the weighted gradient descent, the starting-point must be feasible, i.e., the balance constraint must be satisfied initially. Recently, initialization-free continuous-time distributed methods were proposed to solve the RA problem. In [28], local allocation feasibility constraint was handled by projection or differentiated projection methods. [27] utilized non-smooth exact penalties to handle the local allocation feasibility constraint. While the above works mainly focused on undirected communication graphs, [30, 32] extended the distributed RA algorithms on weight-balanced directed communication graphs. Non-smooth objective functions were investigated in [29, 30], where sub-gradient based methods were developed to handle the non-smooth property of the objective functions. RA problem with uncertainties, including noisy gradient, noisy resource, and noisy communicating information, was solved in [31] by a stochastic distributed algorithm. Sub-optimal algorithm was developed in [32] to reduce the computational complexity or sensitivity to network topology for large-scale resource allocation problem.

Generally, the existing distributed algorithms that solve the RA problem can be divided into discrete-time algorithms and continuously-time algorithms. While discrete-time algorithms are more suitable for digital computers, continuous-time algorithms have attracted much attention in recent years due to the development of cyber-physical systems. Many practical systems such as robots and vehicles operate in continuous-time and the optimization problem is solved by the continuous-time control techniques. Furthermore, the well-developed continuous-time control techniques may facilitate the analysis and provide new design viewpoint and methods. The algorithms can also be distinguished by whether stringent requirements are imposed on the initial values. The initialization to fulfill the requirements, which include setting the initial values of the variables to satisfy stringent requirement and finding global information such as total resource in advance, in a flexible distributed system is restrictive since it involves global coordination and has to be repeatedly performed if the network configuration changes, resource changes, or users partic-

ipation changes [33]. In problems where constraints are handled, projection and penalty function are widely used. The projection operation usually requires the explicit lower and upper bounds on the decision variables or the computational cost of projection step is very expensive in many settings, especially in high-dimensional cases, the projection operator may become computational bottleneck [34]. Penalty method usually induces non-smooth analysis or approximation error.

1.2.2 Distributed Economic Dispatch

The microgrid which is the building block of the smart grid is controlled in a hierarchical manner in islanded mode [35, 36]. As depicted in Figure. 1.3, the control structure is usually separated into three layers, i.e., primary, secondary, and tertiary control. Primary control is the foundation of this hierarchy, it is implemented on

Figure 1.3: Hierarchical control in microgrid.

distributed generator. This control level mainly adjusts the frequency and voltage based on local power generation accomplished via droop control [37], which may cause frequency and voltage deviation. Secondary control is then applied to make them recover to their nominal values [38]. Tertiary control generates the operating point for the primary and secondary control. It is on the top layer of this hierarchy. The ED solves the problem of dispatching the load demands to the available power generating units in an economic, secure, and reliable way. The basic ED problem has the constraints that the total generation equals the total load demand and the generation of each generating unit is within its capacity limits. Besides

the basic constraints, some ED problem also takes into account the transmission loss [39, 11], power flow capacity limits [40, 41, 42], more practical cost functions include valve-point effect and multiple fuel options [43, 44, 45], combined economic emission dispatch [46, 47], dynamic ED with the generator ramp rate limit [48, 49], and combined ED with demand response [50, 51], etc.

In conventional operation of bulk power systems, the ED problem has been solved by various methods, including mathematical methods (e.g. λ -iteration [52], gradient method [53], Newton's method [4]), intelligent searching algorithms (e.g. genetic algorithm [54, 49], differential evolution [55, 56, 57], particle swarm algorithm [58, 59], simulated annealing [60, 61], bee colony optimization [62, 63]), and hybrid methods which combine two or more different methods (e.g. [64, 65]), etc. Generally speaking, mathematical methods usually have proven optimality and do not have problem-specific parameters to specify, however, they usually have restrictive assumptions, suffer from convergence to local optimum for non-convex ED problems, and are sensitive to initial points; intelligent searching methods can handle the complexity involved in practical problems and are not limited by restrictive assumptions, but the drawbacks are suffering from many problem-specific parameters selection and usually have high computational time; the hybrid method combines the best features of different algorithms, thus lead to superior performance than the stand-alone methods, however, they also suffer from long computational time.

Nowadays, power systems are moving from the traditional centralized structure to smart grids which have distributed nature. In a smart grid, integrated distributed generators work collaboratively to serve loads over common distribution networks [4], and distributed information and communication technology are integrated to facilitate efficient, reliable, and economic efficient electricity production and distribution [66]. With this distributed nature, especially in a large-scale power system, the centralized approaches are not suitable or effective due to heavy communication and calculation burden, unexpected single-node-failure, and inflexibility with respect to dynamic networks. To overcome the drawbacks of centralized approaches

and meet the requirements of the smart grids, distributed algorithms have been proposed to solve the ED problem.

In this thesis, we limit our attention on distributed mathematical methods for the basic ED problem where theoretical analysis are conducted. This kind of method provides a clear understanding of the principle and serves as the basis for further study of a more complicated problem. Although we have reviewed the distributed RA algorithms in Section 1.2.1 among which some are capable for a distributed ED, in the following, we give a brief review on works focus on solving the ED problem in a distributed way utilizing consensus algorithms. While not considering the capacity limits on the generators, the gradient of each generator's cost function, which is called the incremental cost, should equal at the optimal generations based on the Lagrange multiplier optimality condition. Based on this criterion, the incremental costs are often synchronized by consensus algorithms in a distributed ED. In [67], the incremental cost was updated based on a consensus algorithm and a leader was responsible to update the incremental cost according to the mismatch between total load and total generation. In [68], an innovation term with a vanishing learning gain was incorporated into the update of the incremental cost to accommodate the equality constraint. In [69], the innovation term was obtained by the collaboration of neighboring agents and the learning gain could be a constant. By finding the total load and total generation using consensus algorithms, a bisection approach on the incremental cost was proposed in [70]. In [11], the transmission losses were considered, by estimating and storing the information of neighboring nodes, the total mismatch and the number of nodes were discovered in a distributed manner, then, the incremental cost was updated based on a consensus algorithm and the total mismatch. Communication information losses were investigated for the ED problem in [71] by a gossip updating rule in finding the mismatch and a correction rule in estimating the incremental cost. In [72], the θ -logarithmic barrier method was utilized to handle the capacity limits and an event-triggered communication scheme was developed. With consideration of wind turbines, [73] proposed a projected gradient

and finite-time average consensus based strategy. A more comprehensive work on distributed control and optimization technologies in smart grid is later given in [74]. By incorporating the capacity limits constraint into the objective function using non-smooth penalties and using dynamic average consensus to estimate the mismatch, an initialization-free distributed Laplacian-nonsmooth-gradient algorithm was proposed in [27]. In [28], a proportional-integral type consensus and projection based continuous time algorithm was proposed to solve the distributed optimization problem in an initialization-free manner. [75] proposed a consensus-based algorithm for the ED of both generation and consumption of consumers, which was able to keep data privacy and reduce computational time. By using a multi-step update, a fast convergence rate was achieved in [76]. [77] considered different and time-varying gains of the mismatch between total load and total generation, which acted as a feedback to the updating of the incremental cost.

In the majority of the existing distributed methods, due to the utilization of consensus algorithms, the initial values of the incremental cost or other auxiliary variables need to meet stringent requirements to achieve the balance between total generation and total demand, or global information, such as the total number of the nodes, is needed or calculated. These requirements impose an initialization procedure in these algorithms, which is restrictive since it involves global coordination and has to be repeatedly performed if the network configuration changes, load changes, or any distributed generator plugs in or leaves off [33]. Moreover, an error in the initial values will result in power mismatch between the total generation and total load. To handle the capacity constraints, two commonly used approaches in continuous algorithms include projection method [28] and exact penalty method [27], which impose non-smooth property and require a non-smooth analysis.

1.2.3 Distributed Time-varying Optimization

In practice, many resources exhibit time-varying nature, such as the loads in power system, which can be treated as resources, are varying from time to time. The TVR make the RA problem a TVO problem with coupled constraints. In the aforementioned works, the resources are assumed to be constant. Compared with time-invariant optimization problems, solving TVO problems is more challenging since the algorithms should not only find but also track the optimal solution trajectory.

There are limited results on TVO on single-agent or with a centralized architecture [78, 79, 80, 81, 82, 83, 84, 85]. In [78], a discrete gradient method was investigated for the unconstrained TVO problems and the intervals between approximate and exact solutions were estimated. [80] derived a fast, augmented Lagrangian tracking algorithm for the TVO problem. [81] proposed iterative discrete-time sampling algorithms based on prediction and correction steps for unconstrained convex TVO problems. In [82], prediction-correction interior point method was developed for TVO problem with inequality and equality constraints. In [83], a sequence optimization problem with convex constraints was solved by a discrete-time algorithm. In [84], a framework for convex TVO was presented, and some common assumptions in existing algorithms were lifted by leveraging averaged operator theory. [85] solved a linearly constrained TVO problem by a prediction-correction method.

In addition to the centralized methods, designing distributed approaches to solve time-varying distributed convex optimization problems is more challenging since the agents can use only local information. In [86], a distributed non-model-based seeking algorithm was proposed to seek the time-varying Nash equilibrium. A multi-agent system with single-integrator and double-integrator dynamics and time-varying objective functions was studied in [87], control algorithms were designed based on consensus algorithms, sign function, and Hessian matrix. A gradient based method was proposed in [88] for a distributed optimization problem with time-varying quadratic objective functions. Furthermore, the case where the decision variables were subject

to local compact convex sets was studied in [89]. In [90], algorithms were derived from the prediction-correction methodology to solve a convex TVO problem with composed objective function consisted of local and network-related variables.

In the aforementioned distributed methods, a majority dealt with unconstrained optimization problems. Only a few considered special constraints, such as [89] considered lower and upper bounds on the decision variable and [90] considered a networked constraint, which is distributed in nature. The RA problem, which has a global resource constraint, cannot be directly solved by the existing distributed methods.

1.3 Contributions

This thesis mainly develops distributed algorithms for the basic RA problem, which has separable convex objective function and equality resource constraint, with static and time-varying resources, respectively. The main contributions of this thesis are listed as follows.

1. Distributed algorithms based on saddle point dynamics (SPD) and consensus algorithms are designed to solve the static basic RA problem in a distributed manner. The distributed algorithms are designed from the primal optimization viewpoint of the RA problem, which has a global resource constraint. Different from the distributed Nash equilibrium seeking, which aims to optimize each player's objective, the RA aims to optimize the sum of all the agents' objective functions. The stability of the proposed algorithms is proven by adopting singular perturbation theory and Lyapunov stability analysis. For the case without local allocation feasibility constraint, exponential stability is achieved under the proposed algorithm, for the case with local allocation feasibility constraint, practical stability is proven under the proposed algorithm. In the proposed algorithms, the consensus algorithms are employed to estimate the

global information in a distributed fashion, and the SPD are adopted to search for the optimal solution with the estimated information. No initialization procedure is required to implement the algorithm since the requirement on the initial values of the state variables is mild and no global information, such as the total number of agents, is required.

2. The basic RA problem with continuously TVR is investigated, and distributed algorithms are proposed to find and track the optimal solution trajectory. The RA problem with TVR is a kind of the TVO problems, which has a global resource constraint in nature. Comparing to unconstrained TVO, the constraint limits the solution space, which makes the optimization problem more challenging to handle. The existence of the solution to the RA problem with TVR is proven, and the condition which the optimal solution should satisfy is given. For the case under quadratic objective functions without local allocation feasibility constraint, a distributed algorithm based on primal dual gradient, consensus algorithms, and non-smooth sign function is proposed. The tracking error is proven to converge to zero in finite time based on the Lyapunov stability analysis. For the case under strongly convex objective functions with local allocation feasibility constraint, a distributed algorithm is designed based on saddle point dynamics and consensus algorithms. The tracking error is proven to be uniformly ultimately bounded based on the Lyapunov stability and singular perturbation analyses.
3. As an application, the ED problem in smart grids is investigated. The algorithms developed are applied and extended to handle the ED problem in a distributed manner. Simulations are conducted on the IEEE test systems to verify the distributed methods, and various case studies are conducted to show the effectiveness of the algorithms. Demonstration platform based on Matlab GUI is developed to demonstrate the ED of an IEEE 9-bus system in an interactive way. Considering valve-point effect and multiple fuel options, a more practical ED problem, which has non-convex and non-smooth cost functions,

is solved distributedly utilizing the randomized gradient and the consensus algorithms.

1.4 Organization of the Thesis

The rest of the thesis is organized as follows. Chapter 2 introduces preliminary knowledge related to this thesis, includes some basic knowledge of graph theory, consensus algorithms, convex optimization, Gaussian smoothing, randomized gradient, and projection operation. Chapter 3 solves the RA problem in a distributed way, two distributed algorithms based on the SPD and consensus algorithms for the case without and with the capacity limits are designed, respectively. The stability of the algorithms is proven through singular perturbation theory and Lyapunov stability analysis. As a typical example of the RA problem, the ED problem is investigated in Chapter 4 in detail. Various case studies are conducted on the IEEE 9-bus and the IEEE 118-bus systems to verify the proposed algorithms. Consider the time-varying feature of the resources, Chapter 5 investigated the RA problem with TVR. Distributed algorithms are developed to find and track the time-varying optimal solution. Chapter 6 solves a more practical non-convex ED problem in a distributed manner by utilizing randomized gradient and the consensus algorithms. Finally, conclusions and suggested future works are drawn in Chapter 7.

Chapter 2

Preliminaries

In this chapter, some basic concepts, definitions, and lemmas on graph theory, consensus algorithms in multi-agent systems, optimality conditions in convex optimization theory, Gaussian smoothing and randomized gradient are introduced.

2.1 Graph Theory and Consensus Algorithms

A graph is defined as $\mathcal{G} \triangleq (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} \triangleq \{1, 2, \dots, n\}$ is the node set and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the edge set. It is undirected if for every $(i, j) \in \mathcal{E}$, $(j, i) \in \mathcal{E}$. An undirected graph is connected if there exists a path between any two distinct nodes. The set of neighbors of node i is denoted by

$$\mathcal{N}_i(\mathcal{E}) \triangleq \{j \in \mathcal{V} | (i, j) \in \mathcal{E}\}.$$

The adjacency matrix A of a graph represents the connectivity of a graph. For an undirected graph, the elements a_{ij} in A are defined as $a_{ii} = 0$, $a_{ij} = a_{ji} > 0$ if node j is connected with node i , and $a_{ij} = 0$ otherwise. The degree matrix D of a graph is a diagonal matrix with the i th diagonal element being $\sum_{j=1}^n a_{ij}$. The Laplacian

matrix L of a graph is defined as $L \triangleq D - A$. For an undirected and connected graph, the eigenvalues of L are all positive except for one zero eigenvalue, $L = L^T$, and $\mathbf{1}_n^T L = \mathbf{0}_n^T$ [91].

The dynamic average consensus algorithm given below is often used to make the agents converge to the average value of a set of references.

Lemma 2.1. [92] *Let $\mathcal{G}$ be an undirected and connected graph, L be the Laplacian matrix of $\mathcal{G}$. Then, for any constant input $u \in \mathbb{R}^n$ and any initial states $x(0), \lambda(0) \in \mathbb{R}^n$, the trajectories of the system*

$$\begin{bmatrix} \dot{x} \\ \dot{\lambda} \end{bmatrix} = \begin{bmatrix} -I - L & -L \\ L & \mathbf{0}_{n \times n} \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix} + \begin{bmatrix} u \\ \mathbf{0}_n \end{bmatrix}$$

represented by $x(t)$ and $\lambda(t)$ converge to constant vectors, and more specifically, $x(t) \rightarrow \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^T u$ exponentially as $t \rightarrow \infty$.

The leader-following consensus algorithm enables the agent to converge to the leader's reference.

Lemma 2.2. [91] *Let $\mathcal{G}$ be an undirected and connected graph. Then, $x_i \rightarrow r_0$ as $t \rightarrow \infty$ if at least one $a_{i0} > 0$ for $i \in \{1, \dots, n\}$ under the following algorithm,*

$$\dot{x}_i = - \sum_{j=1}^n a_{ij}(x_i - x_j) - a_{i0}(x_i - r_0),$$

where r_0 is a time-invariant signal, $a_{i0} > 0$ if node i can get the information of r_0 and $a_{i0} = 0$, otherwise.

A related property on the following matrix is given below.

Lemma 2.3. [93] *Let $\mathcal{A} \in \mathbb{R}^{n \times n}$ and $\mathcal{B} \in \mathbb{R}^{n \times m}$. If $\mathcal{A}$ is Hurwitz and $\text{rank}(\mathcal{B}) = m$, then the matrix $\begin{bmatrix} \mathcal{A} & \mathcal{B} \\ -\mathcal{B}^T & \mathbf{0}_{m \times m} \end{bmatrix}$ is Hurwitz.*

2.2 Convex Optimization

2.2.1 Convex Optimality Condition

Lemma 2.4. [15] *For a convex optimization problem with differentiable objective function $f_0(x)$, denoting $\mathcal{F}$ the feasible region of the problem, then x^* is optimal if and only if $x^* \in \mathcal{F}$ and*

$$\nabla f_0(x^*)^T(x - x^*) \geq 0, \forall x \in \mathcal{F}.$$

2.2.2 Karush-Kuhn-Tucker Conditions

For an optimization problem,

$$\begin{aligned} & \text{minimize} && f_0(x), \\ & \text{subject to} && f_i(x) \leq 0, \quad i = 1, \dots, n, \\ & && h_i(x) = 0, \quad i = 1, \dots, m, \end{aligned}$$

where $f_i(x), i = 0, \dots, n$ and $h_i(x), i = 1, \dots, m$ are differentiable. Let x^* and (λ^*, ν^*) be the primal and dual optimal solution, and the duality gap is zero. Then, they satisfy the following Karush-Kuhn-Tucker (KKT) conditions [15]

$$\nabla f_0(x^*) + \sum_{i=1}^n \lambda_i^* \nabla f_i(x^*) + \sum_{i=1}^m \nu_i^* \nabla h_i(x^*) = 0, \quad (2.1a)$$

$$f_i(x^*) \leq 0, i = 1, \dots, n, \quad (2.1b)$$

$$h_i(x^*) = 0, i = 1, \dots, m, \quad (2.1c)$$

$$\lambda_i^* \geq 0, i = 1, \dots, n, \quad (2.1d)$$

$$\lambda_i^* f_i(x^*) = 0, i = 1, \dots, n. \quad (2.1e)$$

Furthermore, if $f_i(x), i = 0, \dots, n$ are convex and $h_i(x), i = 1, \dots, m$ are affine, $\tilde{x}, \tilde{\lambda}$, and $\tilde{\nu}$ satisfy the KKT conditions (2.1), then $\tilde{x}$ and $(\tilde{\lambda}, \tilde{\nu})$ are the primal and dual optimal solution.

2.2.3 Saddle Point Dynamics

Definition 2.1. [15] A pair (x^*, y^*) is a min-max saddle point of $S(x, y)$ if for all (x, y) , the following inequality is satisfied:

$$S(x^*, y) \leq S(x^*, y^*) \leq S(x, y^*).$$

Figure. 2.1 shows the saddle point $(0, 0)$ of function $S = x^2 - y^2$ for $x, y \in \mathbb{R}$.

Figure 2.1: Saddle point $(0, 0)$ of function $S(x, y) = x^2 - y^2$.

The SPD are used to seek the saddle point of a continuously differentiable function $S(x, y)$, which are given by the following equations [94]:

$$\dot{x} = -k_1 \frac{\partial S(x, y)}{\partial x}, \quad (2.2a)$$

$$\dot{y} = k_2 \frac{\partial S(x, y)}{\partial y}, \quad (2.2b)$$

where k_1 and k_2 are positive constants.

Consider the following convex optimization problem

$$\begin{aligned} & \text{minimize } f(x), \\ & \text{subject to } h(x) = 0, \end{aligned}$$

where $f(x)$ is twice continuously differentiable and strictly convex in x and $h(x)$ is an affine function of x . The Lagrangian function is $S(x, y) = f(x) + \nu h(x)$, where ν is the Lagrange multiplier corresponding to the equality constraint. Suppose the optimal solution of this problem exists and is finite. Let x^* and ν^* be the primal and dual optimal solution, respectively. Then, the SPD (2.2) enables (x, ν) to converge to (x^*, ν^*) asymptotically [95].

2.3 Gaussian Smoothing, Randomized Gradient, and Projection

The Gaussian approximation of a function $f(x) : \mathbb{R} \rightarrow \mathbb{R}$ is [96]

$$f_\mu(x) = \frac{1}{\kappa} \int_{\mathbb{R}} f(x + \mu\xi) e^{-\frac{1}{2}\xi^2} d\xi,$$

where $\kappa = \int_{\mathbb{R}} e^{-\frac{1}{2}\xi^2} d\xi = \sqrt{2\pi}$ and $\mu \geq 0$ is the smoothing parameter. In general, $f_\mu(x)$ has better properties than $f(x)$. For $\mu > 0$, $f_\mu(x)$ is differentiable no matter $f(x)$ is differentiable or not. The gradient of $f_\mu(x)$ is [96]

$$\nabla f_\mu = \frac{1}{\kappa} \int_{\mathbb{R}} \frac{f(x + \mu\xi) - f(x)}{\mu} e^{-\frac{1}{2}\xi^2} \xi d\xi. \quad (2.3)$$

The randomized gradient oracle is defined as [96]

$$g_\mu(x) = \frac{f(x + \mu\xi) - f(x)}{\mu} \xi, \quad (2.4)$$

where ξ is a standard normally distributed random variable. Thus, we have

$$\mathbb{E}[g_\mu(x)] = \nabla f_\mu(x),$$

where $\mathbb{E}$ is the expectation on ξ .

The projection of x onto a closed convex set $\Omega \in \mathbb{R}^n$ is defined by $P_\Omega(x) = \arg \min_{y \in \Omega} \|x - y\|$. The non-expansiveness property of the projection operation is given below.

Lemma 2.5. [97] *For any $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^n$,*

$$\|P_\Omega(x) - P_\Omega(y)\| \leq \|x - y\|.$$

Chapter 3

Distributed Resource Allocation via Saddle Point Dynamics and Consensus Algorithms

3.1 Introduction

Due to the rapid growth in the scale of networked systems and the development of communication and computing technologies, developing distributed algorithms to handle the networked problems is of great importance. In this chapter, we focus on developing distributed algorithms to solve the RA problem. Specifically, we aim to minimize a separable convex objective function subject to the resource constraint that the total allocation equals the total amount of the available resources, and in addition, the local allocation feasibility constraint.

Since the RA problem is a special kind of optimization problem, distributed RA falls into the scope of distributed optimization. The works on distributed optimization

mainly focus on solving a sum of global objective functions as follows [98].

$$\text{minimize } f(x) = \sum_{i=1}^n f_i(x),$$

where $x = [x_i]_{\text{vec}}, i = 1, \dots, n$. Allowing a set of distributed agents collaborate with each other, each agent can get an estimation of the global variable x . Thus, the problem above can be reformulated as a constrained optimization problem [99, 98, 100, 101, 102, 103, 104, 74, 105]

$$\begin{aligned} \text{minimize } f(y) &= \sum_{i=1}^n f_i(y_i) \\ \text{subject to } y_i &= y_j, \quad i, j \in \{1, \dots, n\}, \end{aligned} \quad (3.1)$$

where $y = [y_i]_{\text{vec}}$. The connection between the problem in (3.1) and the RA problem (see (3.2)) has been pointed out in [106] from the Lagrange dual problem of the RA problem. A more special relation between these two problems was given in [107] from the optimality conditions of these two problems. The connection between these two problems indicates that a distributed method solves one problem may also provide possible solution to the other one. While most methods are designed to solve the problem in (3.1), in this chapter, we solve the RA problem from the primal optimization viewpoint, which has a global resource constraint $\sum_{i=1}^n x_i = \sum_{i=1}^n d_i$.

Another related problem is Nash equilibrium seeking among a network of players. In Nash equilibrium seeking in non-cooperative games, each player aims to maximize its own payoff by adjusting its own action given other players' actions [108, 109]. To achieve a distributed manner, each player generates estimation of other players' actions through consensus protocols [110, 111, 112, 113]. However, the object functions are different for Nash equilibrium seeking and distributed optimization. The players in the game intend to maximize their own objective function while the agents in a distributed optimization problem and resource allocation problem aim to optimize the sum of all the agents' objective functions.

In the existing distributed algorithms solving RA problems, a few considered coupled constraints, which are either using an iterative update scheme [26, 114, 115, 116] or a continuous-time scheme [27, 28, 117, 118, 119]. In these works, a lot have an initial value requirement, such as $\sum_{i=1}^n x_i(0) = \sum_{i=1}^n d_i$. This kind of requirement is not straightforward to fulfill in a distributed setting and the initialization error, which may due to some perturbations, will result in a steady-state error. Moreover, if there is any user leaves or joins the networked system or a change in the resource, a re-initialization is required to be performed.

The results in this chapter contribute to the distributed continuous-time scheme solving the RA problem without a stringent initial requirement. The main contributions of this chapter are in three folds.

1. Distributed algorithms based on the consensus algorithms and the SPD are proposed to solve the RA problem. In the distributed algorithms, each node only needs to know its own and its neighbors' information and no global information, such as the total number of the nodes, is needed. The consensus algorithms are employed to estimate the global information in a distributed fashion. With the estimated information, the SPD are adopted to search the optimal solution. The stability of the closed loop system is proven through Lyapunov stability analysis and the singular perturbation theory.
2. No strict requirement is enforced on the initial values of the variables. The initial values of the variables in the algorithm can be arbitrarily set except that the initial values of the Lagrange multipliers corresponding to the local allocation feasibility constraint are required to be positive. Compared with existing initialization-free works in [27, 28], we use a different design scheme comes from the saddle-point dynamics, and the local allocation feasibility constraint is handled based on a more general method, which can be easily extended to handle a more general inequality constraint.
3. The privacy is kept within each node. Only estimated values are exchanged

with neighboring nodes. No information which may be confidential, such as local resource and local incremental cost, is required to be communicated with other nodes.

The rest of this chapter is organized as follows. In Section 3.2, the problem formulation is given and basic assumptions on the problem are introduced. Section 3.3 gives the distributed algorithms that solve the RA problem under quadratic objective functions and box-like local allocation feasibility constraint. Section 3.4 extends the results to a more general problem with strictly convex objective functions and local convex inequality constraints. A numerical example of a software-testing RA problem is solved in Section 3.5 to illustrate the proposed algorithms. Finally, Section 3.6 gives the conclusion.

3.2 Problem Formulation

Consider a RA problem over a network of n agents. Each agent is associated with a decision variable $x_i \in \mathbb{R}$, a corresponding objective function $f_i(x_i) : \mathbb{R} \rightarrow \mathbb{R}$, and a local resource d_i . The objective of RA is to assign the available resource to each agent such that the overall objective, which is $\sum_{i=1}^n f_i(x_i)$, is optimized under the resource constraint $\sum_{i=1}^n x_i = \sum_{i=1}^n d_i$. In some problems, the allocation of agent i is required to satisfy the local allocation feasibility constraint $x_i^{\min} \leq x_i \leq x_i^{\max}$, where $x_i^{\min} \in \mathbb{R}$ is the lower bound and $x_i^{\max} \in \mathbb{R}$ is the upper bound of the allocation variable x_i . Denote $x = [x_i]_{\text{vec}}$, $x^{\min} = [x_i^{\min}]_{\text{vec}}$ and $x^{\max} = [x_i^{\max}]_{\text{vec}}$. The RA problem is formulated as [6]

$$\text{minimize } f(x) = \sum_{i=1}^n f_i(x_i) \quad (3.2a)$$

$$\text{subject to } \sum_{i=1}^n x_i = \sum_{i=1}^n d_i, \quad (3.2b)$$

$$x_i^{\min} \leq x_i \leq x_i^{\max}, \quad i = 1, \dots, n. \quad (3.2c)$$

For the purpose of simplicity, the decision variables and parameters are assumed to be one-dimensional. The algorithms developed in this chapter can be directly applied to solve the multi-dimensional variable case.

In a distributed control architecture, there is no central node to collect all the agents' information. Each agent calculates its own allocation by using its own and its neighbors' information. First, the RA problem without the local allocation feasibility constraint (3.2c) is investigated. The problem is formally described below.

Problem 3.1. (Distributed RA without Local Allocation Feasibility Constraint) *Consider the RA problem over a network of n agents without local allocation feasibility constraint*

$$\text{minimize } f(x) = \sum_{i=1}^n f_i(x_i) \quad (3.3a)$$

$$\text{subject to } \sum_{i=1}^n x_i = \sum_{i=1}^n d_i. \quad (3.3b)$$

Suppose the optimal solution exists and is denoted as $x_i^, i = 1, \dots, n$. The agents communicate with each other via an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Design algorithms $u_i(x_i, x_j), j \in \mathcal{N}_i$, such that $x_i \rightarrow x_i^*$ under the dynamics $\dot{x}_i = u_i(x_i, x_j)$.*

Then, the local allocation feasibility constraint is taken into consideration. The problem is defined as

Problem 3.2. (Distributed RA with Local Allocation Feasibility Constraint) *Consider the RA problem (3.2) over a network of n agents. Suppose the optimal solution exists and is denoted as $\bar{x}_i^*, i = 1, \dots, n$. The agents communicate with each other via an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Design algorithms $u_i(x_i, x_j), j \in \mathcal{N}_i$, such that $x_i \rightarrow \bar{x}_i^*$ under the dynamics $\dot{x}_i = u_i(x_i, x_j)$.*

Remark 3.1. *In some problems, the aggregate resource is known instead of distributed local resources, such as the total asset in portfolio selection problem. In such situation, any $d_j, j \in \mathcal{V}$ can be set as the aggregate resource, and other $d_i, i \in$*

$\{1, \dots, j-1, j+1, \dots, n\}$ are set as zero. However, if in problems where only local resources are available, such as the local loads in smart grids, it is difficult to obtain the aggregate resource in a distributed architecture, especially in a large-scale networked system.

Throughout this chapter, the following assumptions hold.

Assumption 3.1. *The communication graph $\mathcal{G}$ among the agents is undirected and connected.*

The connectivity assumption of the communication graph is commonly used in networked multi-agent system [120, 121, 122], which is used to ensure the information can be propagated throughout the system. This assumption can be easily satisfied when designing the communication topology.

Assumption 3.2. *The RA problem in (3.2) is feasible, i.e.,*

$$\sum_{i=1}^n x_i^{\min} \leq \sum_{i=1}^n d_i \leq \sum_{i=1}^n x_i^{\max}.$$

The feasibility assumption is a necessary condition for the existence of the solution. If this assumption is not satisfied, some preprocessing should be performed, such as improving the resource or reducing the users. The satisfaction of the feasibility can be checked by solving the corresponding feasibility problem [15], where the objective is a constant and thus the feasibility problem can be solved by simple linear programming.

3.3 Distributed Resource Allocation under Quadratic Objective Functions

First, the objective function $f_i(x_i)$ is assumed to be quadratic. The convex quadratic objective function has the following form

$$f_i(x_i) = a_i x_i^2 + b_i x_i + c_i, \quad (3.4)$$

where $a_i > 0$, b_i , and c_i are the parameters.

3.3.1 Distributed Resource Allocation without Local Allocation Feasibility Constraint

In this section, Problem 3.1 under quadratic objective function (3.4) is investigated.

3.3.1.1 Centralized Approach Based on the SPD

Based on the SPD, a centralized approach is first introduced. The Lagrangian function of the problem in (3.3) is

$$L_1(x, \lambda_0) = f(x) + \lambda_0 \left(\sum_{i=1}^n d_i - \sum_{i=1}^n x_i \right),$$

where $\lambda_0 \in \mathbb{R}$ is the Lagrange multiplier corresponding to the resource constraint. Since the problem in (3.3) is convex, there is a λ_0^* which is the dual optimal value such that (x^*, λ_0^*) , where $x^* = [x_i^*]_{\text{vec}}$, is the saddle point of $L_1(x, \lambda_0)$ [15]. Furthermore, (x^*, λ_0^*) satisfies the following Lagrange multiplier optimality condition

$$\frac{\partial L_1}{\partial x}(x^*, \lambda_0^*) = \frac{\partial f(x)}{\partial x}(x^*) - \mathbf{1}_n \lambda_0^* = 0, \quad (3.5a)$$

$$\frac{\partial L_1}{\partial \lambda_0}(x^*, \lambda_0^*) = \sum_{i=1}^n d_i - \sum_{i=1}^n x_i^* = 0. \quad (3.5b)$$

Inspired by the SPD (2.2), the following dynamics

$$\dot{x}_i = -\frac{\partial L_1}{\partial x_i} = -\left(\frac{df_i(x_i)}{dx_i} - \lambda_0\right), \quad i \in \{1, \dots, n\}, \quad (3.6a)$$

$$\dot{\lambda}_0 = \frac{\partial L_1}{\partial \lambda_0} = \sum_{i=1}^n d_i - \sum_{i=1}^n x_i, \quad (3.6b)$$

can be utilized to search the saddle point (x^*, λ_0^*) .

However, λ_0 in (3.6a) and the values of $\sum_{i=1}^n d_i$ and $\sum_{i=1}^n x_i$ in (3.6b) are global information. Hence, a central unit is needed to implement the algorithm in (3.6). In the following, a distributed approach is developed.

3.3.1.2 Distributed Approach

The distributed updating laws for x_i and λ_0 are designed as

$$\dot{x}_i = -\bar{k}_i \left(\frac{df_i(x_i)}{dx_i} - \lambda_i \right), \quad i = 1, \dots, n, \quad (3.7a)$$

$$\dot{\lambda}_0 = \bar{m}_0 z_q, \quad (3.7b)$$

where $\bar{k}_i = \delta k_i$, $\bar{m}_0 = \delta m_0$, δ is a small positive parameter to be designed, k_i and m_0 are fixed positive parameters, λ_i is agent i 's estimation of λ_0 , and z_q is agent q 's estimation of $\frac{1}{n} \sum_{i=1}^n (d_i - x_i)$. Any z_q , $q \in \{1, \dots, n\}$ can be selected to calculate λ_0 . The estimation algorithms on λ_i and z_q are designed based on the dynamic average consensus algorithm in Lemma 2.1 and the leader-following consensus algorithm in Lemma 2.2 as follows

$$\dot{z}_i = -z_i - \sum_{j=1}^n a_{ij}(z_i - z_j) - \sum_{j=1}^n a_{ij}(\theta_i - \theta_j) + d_i - x_i, \quad (3.8a)$$

$$\dot{\theta}_i = \sum_{j=1}^n a_{ij}(z_i - z_j), \quad (3.8b)$$

$$\dot{\lambda}_i = - \sum_{j=1}^n a_{ij}(\lambda_i - \lambda_j) - a_{i0}(\lambda_i - \lambda_0), \quad (3.8c)$$

where θ_i is an intermediate variable.

Theorem 3.1. *Under Assumption 3.1, there exists a positive constant δ^* such that for every $0 < \delta < \delta^*$, x converges to x^* exponentially under algorithms (3.7) and (3.8).*

Proof. Let $U = [U_1 \ U_2] \in \mathbb{R}^{n \times n}$ with $U_1 \in \mathbb{R}^{n \times (n-1)}$ and $U_2 \in \mathbb{R}^n$ be an orthogonal matrix such that $U_2^T L = 0$. Define $[\theta_a^T \ \theta_b]^T \triangleq U^T \theta$ where $\theta = [\theta_i]_{\text{vec}}$, $\theta_a \in \mathbb{R}^{n-1}$, and $\theta_b \in \mathbb{R}$. From (3.8a) and (3.8b) it can be derived that

$$\begin{bmatrix} \dot{z} \\ \dot{\theta}_a \end{bmatrix} = \begin{bmatrix} -(I + L) & -LU_1 \\ U_1^T L & \mathbf{0}_{(n-1) \times (n-1)} \end{bmatrix} \begin{bmatrix} z \\ \theta_a \end{bmatrix} + \begin{bmatrix} d - x \\ \mathbf{0}_{n-1} \end{bmatrix}, \quad (3.9a)$$

$$\dot{\theta}_b = 0, \quad (3.9b)$$

where $z = [z_i]_{\text{vec}}$ and $d = [d_i]_{\text{vec}}$. Hence, $\theta_b(t) = \theta_b(0)$, which is a stable state. Thus, in the following analysis, θ_b will be omitted.

Let $\tau = \delta t$. Equations (3.7), (3.8c) and (3.9a) in the τ -time scale can be written as

$$\frac{dx}{d\tau} = -K \left(\frac{\partial f(x)}{\partial x} - \lambda \right), \quad (3.10a)$$

$$\frac{d\lambda_0}{d\tau} = m_0 z_q, \quad (3.10b)$$

$$\delta \frac{d\lambda}{d\tau} = -(L + A_0)\lambda + A_0 \mathbf{1}_n \lambda_0, \quad (3.10c)$$

$$\begin{bmatrix} \delta \frac{dz}{d\tau} \\ \delta \frac{d\theta_a}{d\tau} \end{bmatrix} = \begin{bmatrix} -(I + L) & -LU_1 \\ U_1^T L & \mathbf{0}_{(n-1) \times (n-1)} \end{bmatrix} \begin{bmatrix} z \\ \theta_a \end{bmatrix} + \begin{bmatrix} d - x \\ \mathbf{0}_{n-1} \end{bmatrix}, \quad (3.10d)$$

where $K = \text{diag}\{k_i\}$. The system given by (3.10) attains a singular perturbed form given in Chapter 11 in [123]. The small parameter δ separates the dynamics into

two-time-scale models, i.e., a reduced (slow) and a boundary-layer (fast) models. By analyzing the properties of the separated models, the properties of the whole system can be obtained.

Let λ^e , z^e and θ_a^e be the solution of the following equations.

$$\begin{aligned} & -(L + A_0)\lambda^e + A_0\mathbf{1}_n\lambda_0 = \mathbf{0}_n, \\ & \begin{bmatrix} -(I + L) & -LU_1 \\ U_1^T L & \mathbf{0}_{(n-1) \times (n-1)} \end{bmatrix} \begin{bmatrix} z^e \\ \theta_a^e \end{bmatrix} + \begin{bmatrix} d - x \\ \mathbf{0}_{n-1} \end{bmatrix} = \mathbf{0}_{2n-1}, \end{aligned}$$

where $A_0 = \text{diag}\{a_{i0}\}$. Since $L + A_0$ is positive definite, $\lambda^e = \mathbf{1}_n\lambda_0$. By Lemma 2.1 and Lemma 2.1 in [124], $z^e = \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T(d - x)$, and θ_a^e is a linear function of x .

By setting $\lambda = \lambda^e$ and $z_q = z_q^e$ in (3.10a) and (3.10b), we get the reduced system

$$\begin{bmatrix} \frac{dx}{d\tau} \\ \frac{d\lambda_0}{d\tau} \end{bmatrix} = H_2 \begin{bmatrix} x \\ \lambda_0 \end{bmatrix} + \mathbf{b}_1,$$

where

$$H_2 = \begin{bmatrix} -\text{diag}\{2k_i a_i\} & [k_i]_{\text{vec}} \\ -\frac{m_0}{n}\mathbf{1}_n^T & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{b}_1 = \begin{bmatrix} -[k_i b_i]_{\text{vec}} \\ \frac{m_0}{n}\mathbf{1}_n^T d \end{bmatrix}.$$

Define

$$P \triangleq \begin{bmatrix} \text{diag}\{\sqrt{k_i}\} & \mathbf{0}_n \\ \mathbf{0}_n^T & \sqrt{\frac{m_0}{n}} \end{bmatrix}.$$

It can be derived that

$$P^{-1}H_2P = \begin{bmatrix} -\text{diag}\{2k_i a_i\} & \sqrt{\frac{m_0}{n}}[\sqrt{k_i}]_{\text{vec}} \\ -\sqrt{\frac{m_0}{n}}[\sqrt{k_i}]_{\text{vec}}^T & 0 \end{bmatrix}.$$

$P^{-1}H_2P$ is Hurwitz by Lemma 2.3. As a result, H_2 is Hurwitz due to similarity

transformation. Define $\tilde{\phi} \triangleq \begin{bmatrix} x - x^* \\ \lambda_0 - \lambda_0^* \end{bmatrix}$, by noticing that $\begin{bmatrix} x^* \\ \lambda_0^* \end{bmatrix} = -H_2^{-1}\mathbf{b}_1$ from the Lagrange multiplier optimality condition (3.5), it can be derived that

$$\frac{d\tilde{\phi}}{d\tau} = H_2\tilde{\phi}. \quad (3.12)$$

Thus, the reduced system is exponentially stable.

Define $\psi \triangleq [\lambda^T, z^T, \theta_a^T]^T$, $\psi^e \triangleq [\lambda^{eT}, z^{eT}, \theta_a^{eT}]^T$, and $\tilde{\psi} \triangleq \psi - \psi^e$. Then we have the boundary layer model

$$\delta \frac{d\tilde{\psi}}{d\tau} = H_1\tilde{\psi}, \quad (3.13)$$

where

$$H_1 = \begin{bmatrix} -(L + A_0) & \mathbf{0}_{n \times n} & \mathbf{0}_{n \times (n-1)} \\ \mathbf{0}_{n \times n} & -(I + L) & -LU_1 \\ \mathbf{0}_{(n-1) \times n} & U_1^T L & \mathbf{0}_{(n-1) \times (n-1)} \end{bmatrix}.$$

Since $\begin{bmatrix} -(I + L) & -LU_1 \\ U_1^T L & \mathbf{0}_{(n-1) \times (n-1)} \end{bmatrix}$ is Hurwitz due to Lemma 2.3 and $-(L + A_0)$ is Hurwitz, H_1 is Hurwitz. Thus, the boundary-layer system is exponentially stable.

Therefore, from Theorem 11.4 in [123] and noticing that the closed loop system is linear, we can conclude that there exists a positive constant δ^* such that for every $0 < \delta < \delta^*$, x converges to x^* exponentially. $\square$

Remark 3.2. *In this algorithm, agent q serves as a virtual leader which generates the signal λ_0 . Only a subset of the agents are required to be connected with agent q . Compared to centralized approaches, failure of the virtual leader can be easily recovered by appointing any other agent, for example, agent r , to calculate λ_0 using its own z_r , while in centralized approaches, the failure of the central node is disastrous.*

3.3.2 Distributed Resource Allocation with Local Allocation Feasibility Constraint

In this section, Problem 3.2 under quadratic objective function (3.4) is investigated. With the local allocation feasibility constraint (3.2c) considered, we introduce an additional assumption, which is used to ensure the uniqueness of the dual optimal value.

Assumption 3.3. *There is at least one agent whose optimal allocation is strictly within its bound. That is, $\exists i$, such that $x_i^{\min} < \bar{x}_i^* < x_i^{\max}$.*

Assumption 3.3 is used to ensure the uniqueness of the optimal Lagrange multiplier. If this assumption is not satisfied, the optimal allocation of each agent lies on its boundary. This will happen when the total resource equals the summation of lower bounds or upper bounds, where the optimal solution can be obtained directly by feasibility test, or when the incremental costs for some agents have no intersection with others within their local allocation feasibility constraints, this situation can also be checked easily by looking into the property of the cost functions.

The local allocation feasibility constraint is critical for some practical problems, such as the ED in power system. Because the KKT conditions are different from the Lagrange multiplier optimality condition (3.5), the consideration of local allocation feasibility constraint introduces additional challenge to distributed methods compared with those without local allocation feasibility constraint. With the local allocation feasibility constraint, the Lagrangian function of the problem in (3.2) is

$$L_2(x, \gamma) = f(x) + \lambda_0 \left(\sum_{i=1}^n d_i - \sum_{i=1}^n x_i \right) + \sum_{i=1}^n \alpha_i (x_i - x_i^{\max}) + \sum_{i=1}^n \beta_i (x_i^{\min} - x_i),$$

where $\gamma = [\lambda_0, \alpha_1, \beta_1, \dots, \alpha_n, \beta_n]^T$ and α_i and β_i are the Lagrange multipliers corresponding to the local allocation feasibility constraint. Since the problem in (3.2) is convex and the constraints are all linear, there exists $\gamma^* = [\lambda_0^*, \alpha_1^*, \beta_1^*, \dots, \alpha_n^*, \beta_n^*]^T$

such that $(\bar{x}^*, \gamma^*)$, where $\bar{x}^* = [\bar{x}_i^*]_{\text{vec}}$, is the saddle point of $L_2(x, \gamma)$. In the meantime, $(\bar{x}^*, \gamma^*)$ satisfies the following KKT conditions,

$$\begin{aligned}
\frac{df_i}{dx_i}(\bar{x}_i^*) - \lambda_0^* + \alpha_i^* - \beta_i^* &= 0, \\
\sum_{i=1}^n d_i - \sum_{i=1}^n \bar{x}_i^* &= 0, \\
\bar{x}_i^* - x_i^{\max} &\leq 0, \\
x_i^{\min} - \bar{x}_i^* &\leq 0, \\
\alpha_i^*(\bar{x}_i^* - x_i^{\max}) &= 0, \\
\beta_i^*(x_i^{\min} - \bar{x}_i^*) &= 0, \\
\alpha_i^* \geq 0, \beta_i^* &\geq 0.
\end{aligned} \tag{3.14}$$

Motivated by [93], the distributed updating law for agent i is designed as

$$\dot{x}_i = -\bar{k}_i \left(\frac{df_i(x_i)}{dx_i} - \lambda_i + \alpha_i - \beta_i \right), \tag{3.15a}$$

$$\dot{\alpha}_i = \bar{m}_{i1} \alpha_i (x_i - x_i^{\max}), \tag{3.15b}$$

$$\dot{\beta}_i = \bar{m}_{i2} \beta_i (x_i^{\min} - x_i), \tag{3.15c}$$

where $\bar{m}_{ij} = \delta m_{ij}$, m_{ij} for $j \in \{1, 2\}$ are fixed positive constants, $\alpha_i(0) > 0$, and $\beta_i(0) > 0$. For agent q , in addition to (3.15), it needs to calculate λ_0 the same as that in Section 3.3.1 as

$$\dot{\lambda}_0 = \bar{m}_0 z_q. \tag{3.16}$$

The estimation algorithm is the same as that in Section 3.3.1, i.e.,

$$\dot{z}_i = -z_i - \sum_{j=1}^n a_{ij}(z_i - z_j) - \sum_{j=1}^n a_{ij}(\theta_i - \theta_j) + d_i - x_i, \tag{3.17a}$$

$$\dot{\theta}_i = \sum_{j=1}^n a_{ij}(z_i - z_j), \tag{3.17b}$$

$$\dot{\lambda}_i = - \sum_{j=1}^n a_{ij}(\lambda_i - \lambda_j) - a_{i0}(\lambda_i - \lambda_0). \quad (3.17c)$$

Define $o = [x^T \ \alpha^T \ \beta^T \ \lambda_0 \ \lambda^T \ z^T \ \theta_a^T]^T$ and $o^* = [\bar{x}^{*T} \ \alpha^{*T} \ \beta^{*T} \ \lambda_0^* \ \lambda_0^* \mathbf{1}_n^T \ \frac{1}{n} \mathbf{1}_n^T (\bar{x}^* - d) \ \theta_a^{eT}(\bar{x}^*)]^T$. Then, the following result can be obtained.

Theorem 3.2. *Suppose that Assumptions 3.1-3.3 hold. Then, for each pair of positive numbers (Δ, ζ) , there exists a positive constant $\delta^*(\Delta, \zeta)$, such that for every $0 < \delta < \delta^*$, under algorithms (3.15), (3.16), and (3.17) with $\alpha(0) > 0, \beta(0) > 0$, there exists a $T > 0$ such that*

$$\|o - o^*\| \leq \zeta, \quad \forall t > T,$$

$$\text{for } \|o(0) - o^*\| \leq \Delta.$$

Proof. By the similar variable transformation on θ in the proof of Theorem 3.1, the definition $\psi = [\lambda^T, z^T, \theta_a^T]^T$, $\psi^e = [\lambda^{eT}, z^{eT}, \theta_a^{eT}]^T$ with $\lambda^e = \mathbf{1}_n \lambda_0$ and $z^e = \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^T (d - x)$, and $\tilde{\psi} = \psi - \psi^e$, according to (3.17), we get

$$\dot{\tilde{\psi}} = H_1 \tilde{\psi} - \dot{\psi}^e. \quad (3.18)$$

Since H_1 is Hurwitz, there exists a symmetric positive definite matrix P_1 such that

$$H_1 P_1 + P_1 H_1^T = -Q_1$$

for any given symmetric positive definite matrix Q_1 . Writing (3.15) and (3.16) as functions of $\tilde{\lambda} = \lambda - \lambda^e$ and $\tilde{z} = z - z^e$ gives

$$\dot{x}_i = -\delta k_i \left(\frac{df_i(x_i)}{dx_i} - \lambda_0 + \alpha_i - \beta_i \right) + \delta k_i \tilde{\lambda}_i, \quad (3.19a)$$

$$\dot{\alpha}_i = \delta m_{i1} \alpha_i (x_i - x_i^{\max}), \quad (3.19b)$$

¹ θ_a is a function of θ .

² $\theta_a^{eT}(x^*)$ is a function of x^* . The details can be found in the proof of Theorem 3.2.

$$\dot{\beta}_i = \delta m_{i2} \beta_i (x_i^{\min} - x_i), \quad (3.19c)$$

$$\dot{\lambda}_0 = \frac{\delta m_0}{n} \left(\sum_{i=1}^n d_i - \sum_{i=1}^n x_i \right) + \delta m_0 \tilde{z}_q. \quad (3.19d)$$

First, we check the stability of the following reduced system,

$$\dot{x}_i = -k_i \left(\frac{df_i(x_i)}{dx_i} - \lambda_0 + \alpha_i - \beta_i \right), \quad (3.20a)$$

$$\dot{\alpha}_i = m_{i1} \alpha_i (x_i - x_i^{\max}), \quad (3.20b)$$

$$\dot{\beta}_i = m_{i2} \beta_i (x_i^{\min} - x_i), \quad (3.20c)$$

$$\dot{\lambda}_0 = \frac{m_0}{n} \left(\sum_{i=1}^n d_i - \sum_{i=1}^n x_i \right). \quad (3.20d)$$

Consider the Lyapunov function inspired by [88],

$$\begin{aligned} V_r = & \sum_{i=1}^n \frac{1}{2k_i} (x_i - \bar{x}_i^*)^2 + \frac{n}{2m_0} (\lambda_0 - \lambda_0^*)^2 + \sum_{i=1}^n \left(\frac{1}{m_{i1}} (\alpha_i - \alpha_i^* - \alpha_i^* \log(\alpha_i)) \right. \\ & \left. + \alpha_i^* \log(\alpha_i^*)) + \frac{1}{m_{i2}} (\beta_i - \beta_i^* - \beta_i^* \log(\beta_i) + \beta_i^* \log(\beta_i^*)) \right). \end{aligned}$$

Denoting the derivative of V_r along (3.20) as $\dot{V}_{rr}$. Then we have

$$\begin{aligned} \dot{V}_{rr} & \leq L_2(\bar{x}^*, \gamma) - L_2(x, \gamma^*) \\ & = L_2(\bar{x}^*, \gamma) - L_2(\bar{x}^*, \gamma^*) + L_2(\bar{x}^*, \gamma^*) - L_2(x, \gamma^*) \\ & \leq 0. \end{aligned}$$

Because $(\bar{x}^*, \gamma^*)$ is the saddle point of $L_2(x, \gamma)$, $\dot{V}_{rr} = 0$ if and only if

$$L_2(\bar{x}^*, \gamma) - L_2(\bar{x}^*, \gamma^*) = 0, \quad L_2(\bar{x}^*, \gamma^*) - L_2(x, \gamma^*) = 0.$$

Since $L_2(x, \gamma)$ is strictly convex in x , $L_2(\bar{x}^*, \gamma^*) - L_2(x, \gamma^*) = 0$ if and only if $x = \bar{x}^*$. When $x = \bar{x}^*$, $\bar{x}_i^* - x_i^{\max} \leq 0$, $x_i^{\min} - \bar{x}_i^* \leq 0$, and $\sum_{i=1}^n d_i - \sum_{i=1}^n \bar{x}_i^* = 0$.

$L_2(\bar{x}^*, \gamma) - L_2(\bar{x}^*, \gamma^*) = 0$ implies

$$\begin{aligned} & \sum_{i=1}^n \alpha_i(\bar{x}_i^* - x_i^{\max}) + \sum_{i=1}^n \beta_i(x_i^{\min} - \bar{x}_i^*) \\ &= \sum_{i=1}^n \alpha_i^*(\bar{x}_i^* - x_i^{\max}) + \sum_{i=1}^n \beta_i^*(x_i^{\min} - \bar{x}_i^*) = 0. \end{aligned}$$

Since $\bar{x}_i^* - x_i^{\max} \leq 0$, $x_i^{\min} - \bar{x}_i^* \leq 0$, and $\alpha_i \geq 0$, $\beta_i \geq 0$ (which can be easily seen from (3.15b) and (3.15c) with $\alpha_i(0) > 0$ and $\beta_i(0) > 0$), it can be derived that

$$\begin{aligned} \alpha_i(\bar{x}_i^* - x_i^{\max}) &= 0, \\ \beta_i(x_i^{\min} - \bar{x}_i^*) &= 0. \end{aligned} \tag{3.21}$$

When $x = \bar{x}^*$,

$$\dot{x}_i = -k_i(2a_i\bar{x}_i^* + b_i - \lambda_0 + \alpha_i - \beta_i) = 0. \tag{3.22}$$

Above all, γ satisfies the KKT conditions (3.14), which indicates $\gamma = \gamma^*$. As a result, $\dot{V}_{rr} = 0$ only when $(x, \gamma) = (\bar{x}^*, \gamma^*)$. Thus, (x, γ) converges to the saddle point of $L_2(x, \gamma)$ asymptotically.

When $x_i^{\min} < \bar{x}_i^* < x_i^{\max}$, $\alpha_i = \beta_i = 0$ at $\dot{V}_{rr} = 0$ from (3.21). For $i \in \{1, \dots, n\}$, define $N_{\text{in}} = \{i | x_i^{\min} < \bar{x}_i^* < x_i^{\max}\}$, $N_{\text{max}} = \{i | \bar{x}_i^* - x_i^{\max} = 0\}$ and $N_{\text{min}} = \{i | x_i^{\min} - \bar{x}_i^* = 0\}$. Since any two of $x_i^{\min} < \bar{x}_i^* < x_i^{\max}$, $\bar{x}_i^* - x_i^{\max} = 0$, and $x_i^{\min} - \bar{x}_i^* = 0$ cannot be satisfied simultaneously, $N_{\text{in}} \cap N_{\text{max}} = \emptyset$, $N_{\text{in}} \cap N_{\text{min}} = \emptyset$, and $N_{\text{max}} \cap N_{\text{min}} = \emptyset$. Let $e_i \in \mathbb{R}^n$ denote an n dimensional column vector whose i th element is equal to 1 and other elements are equal to zero. Define a matrix $A_f \in \mathbb{R}^{n \times n}$ whose i th column is

$$\text{col}[A_f]_i = \begin{cases} -e_i, & i \in N_{\text{max}} \\ e_i, & i \in N_{\text{min}} \\ \mathbf{0}_n, & i \in N_{\text{in}} \end{cases}.$$

Remove the zero columns of A_f and define the remaining matrix as $A_a \in \mathbb{R}^{n \times (n - |N_{\text{in}}|)}$, where $|N_{\text{in}}|$ is the number of elements in N_{in} . For γ , if $i \in N_{\text{in}}$, remove α_i and β_i ; if $i \in N_{\text{max}}$, remove β_i ; if $i \in N_{\text{min}}$, remove α_i . Define the remaining vector of γ as γ_a . From (3.22) it can be derived that

$$[\mathbf{1}_n \ A_a] \gamma_a = \text{diag} \{2a_i\} \bar{x}^* + [b_i]_{\text{vec}}.$$

Since $[\mathbf{1}_n \ A_a]$ has full column rank by Assumption 3.3, γ_a is unique. As a result, γ^* is unique. Define $\bar{\phi} = [(x - \bar{x}^*)^T (\gamma - \gamma^*)^T]^T$. From Lemma 4.3 in [123], there exists a $\mathcal{K}$ function $W(\bar{\phi})$ such that

$$\dot{V}_{rr} \leq -W(\bar{\phi}).$$

Let

$$V = \tilde{\psi}^T P_1 \tilde{\psi} + V_r.$$

For every (x, λ_0) that belongs to a compact set, calculating the derivative of V along (3.18) and (3.19) gives

$$\begin{aligned} \dot{V} &= -\tilde{\psi}^T Q_1 \tilde{\psi} - 2\tilde{\psi}^T P_1 \dot{\psi}^e + \delta \dot{V}_{rr} + \delta(x - \bar{x}^*)^T \tilde{\lambda} + \delta n(\lambda_0 - \lambda^*) \tilde{z}_q \\ &\leq -\lambda_{\min}(Q_1) \|\tilde{\psi}\|^2 + \delta d_1 \|\tilde{\psi}\|^2 + \delta d_2 \|\tilde{\psi}\| \|\tilde{\phi}\| - \delta W(\bar{\phi}) + \delta d_3 \|\tilde{\psi}\| \|\tilde{\phi}\| \\ &= -(\lambda_{\min}(Q_1) - \delta d_1) \|\tilde{\psi}\|^2 - \delta W(\bar{\phi}) + \delta d_4 \|\tilde{\psi}\| \|\tilde{\phi}\| \\ &\leq -(\lambda_{\min}(Q_1) - \delta d_1 - \frac{\delta d_4 \epsilon}{2}) \|\tilde{\psi}\|^2 - \delta W(\bar{\phi}) + \frac{\delta d_4}{2\epsilon} \|\tilde{\phi}\|^2, \end{aligned}$$

where $d_1, d_2, d_3, d_4, \epsilon$ are positive constants. By selecting $\delta < \frac{2\lambda_{\min}(Q_1)}{2d_1 + d_4 \epsilon} \triangleq \delta^*$, then

$$\dot{V} \leq -\frac{1}{2} W'(\tilde{\psi}, \bar{\phi}), \quad \forall \|\tilde{\psi}^T \bar{\phi}^T\| > W'^{-1}(2d_5),$$

where $W'(\tilde{\psi}, \bar{\phi})$ is a $\mathcal{K}$ function, and $d_5 = \frac{\delta d_4}{2\epsilon} \sup\{\|\tilde{\phi}\|^2\}$. Thus, from Theorem 4.18 in [123], the conclusion is obtained. $\square$

3.4 Distributed Resource Allocation with Strictly Convex Objective Functions and Local Con- vex Inequality Constraint

In this section, the result obtained in Section 3.3.2 is extended to a more general case, where the objective function is assumed to be strongly convex instead of quadratic and the local allocation feasibility constraint (3.2c) is assumed to take a more general form instead of lower and upper limits.

Consider the following problem formulation

$$\text{minimize } F(x) = \sum_{i=1}^n F_i(x_i), \quad (3.23a)$$

$$\text{subject to } \sum_{i=1}^n x_i = \sum_{i=1}^n d_i, \quad (3.23b)$$

$$g_{ij}(x_i) \leq 0, \quad i \in \{1, 2, \dots, n\}, j \in \{1, \dots, m_i\}, \quad (3.23c)$$

where m_i is the number of local inequality constraints on x_i . For the problem in (3.23), we introduce the following assumption.

Assumption 3.4. $F_i(x_i)$ is twice continuously differentiable and strictly convex, $g_{ij}(x_i)$ is twice continuously differentiable and convex. There exists a finite optimal solution x_g^* . The Slater's condition is satisfied. There exists at most one $j \in \{1, \dots, m_i\}$ such that $g_{ij}(x_{gi}^*) = 0 \forall i$; $\exists i$, such that $g_{ij}(x_{gi}^*) < 0$, $j = 1, \dots, m_i$.

The differentiability and convexity of the objective function and inequality constraint function in Assumption 3.4 is general for convex optimization. The Slater's condition ensures that the gap between primal and dual optimal values is zero. Other conditions are similar to Assumption 3.1 discussed before.

With slight revision, the distributed updating law that solves the problem in (3.23)

for agent i is designed as

$$\dot{x}_i = -\bar{k}_i \left(\frac{dF_i(x_i)}{dx_i} - \lambda_i + \sum_{j=1}^{m_i} \alpha_{ij} \frac{dg_{ij}(x_i)}{dx_i} \right), \quad (3.24a)$$

$$\dot{\lambda}_0 = \bar{m}_0 z_q, \quad (3.24b)$$

$$\dot{\alpha}_{ij} = \bar{m}_{ij} \alpha_{ij} g_{ij}(x_i), \quad (3.24c)$$

$$\dot{z}_i = -z_i - \sum_{j=1}^n a_{ij}(z_i - z_j) - \sum_{j=1}^n a_{ij}(\theta_i - \theta_j) + d_i - x_i, \quad (3.24d)$$

$$\dot{\theta}_i = \sum_{j=1}^n a_{ij}(z_i - z_j), \quad (3.24e)$$

$$\dot{\lambda}_i = -\sum_{j=1}^n a_{ij}(\lambda_i - \lambda_j) - a_{i0}(\lambda_i - \lambda_0). \quad (3.24f)$$

Theorem 3.3. *Under Assumptions 3.1 and 3.4, for each pair of positive numbers (Δ, ζ) , there exists a positive constant $\delta^*(\Delta, \zeta)$, such that for every $0 < \delta < \delta^*$, under algorithm (3.24) with $\alpha_{ij}(0) > 0$, there exists a $T > 0$ such that*

$$\|o - o^*\| \leq \zeta, \quad \forall t > T,$$

for $\|o(0) - o^*\| \leq \Delta$, where $o = [x^T \ [\alpha_{ij}]_{vec}^T \ \lambda_0 \ \lambda^T \ z^T \ \theta_1^T]^T$ and $o^* = [x_g^* \ [\alpha_{ij}^*]_{vec}^T \ \lambda_0^* \ \lambda_0^{*T} \ \frac{1}{n} \mathbf{1}_n^T (x_g^* - d) \mathbf{1}_n^T \ \theta_1^{eT}(x_g^*)]^T$.

Proof. The proof is similar to that of Theorem 3.2, hence is omitted here. $\square$

Remark 3.3. *In this algorithm, (3.24c) are utilized to make α and β satisfy the KKT conditions, such that x satisfies the inequality constraints. Different from other methods, such as projection method or using non-smooth penalty function to incorporate the box constraints into the objective function, this algorithm directly utilizes the function of the inequality constraints, which does not require explicit lower and upper bounds on the decision variables.*

Remark 3.4. *In Theorems 3.1-3.3, the variables x_i and resources d_i are assumed to be scalar. Actually, the results in these theorems also hold for higher-dimensional*

case. Suppose $x_i, d_i, x_i^{\min}, x_i^{\max} \in \mathbb{R}^d$, and $f_i(x_i) : \mathbb{R}^d \rightarrow \mathbb{R}$ with the cost function coefficients has proper dimensions. Then, the algorithms remain similar except that the derivative of $\frac{df_i(x_i)}{dx_i}$ in (3.7a) and (3.15a) change to $\frac{\partial f_i(x_i)}{\partial x_i}$, $\frac{dF_i(x_i)}{dx_i}$ in (3.24a) changes to $\frac{\partial F_i(x_i)}{\partial x_i}$, $\dot{\alpha}_i = \bar{m}_{i1}\alpha_i(x_i - x_i^{\max})$, $\dot{\beta}_i = \bar{m}_{i2}\beta_i(x_i^{\min} - x_i)$ in (3.15b) and (3.15c) change to $\dot{\alpha}_i = \bar{m}_{i1}\text{diag}\{\alpha_i\}(x_i - x_i^{\max})$, $\dot{\beta}_i = \bar{m}_{i2}\text{diag}\{\beta_i\}(x_i^{\min} - x_i)$, and $\dot{\alpha}_{ij} = \bar{m}_{ij}\alpha_{ij}g_{ij}(x_i)$ in (3.24c) changes to $\dot{\alpha}_{ij} = \bar{m}_{ij}\text{diag}\{\alpha_{ij}\}g_{ij}(x_i)$, where $\text{diag}\{\alpha_i\}$ is the diagonal matrix with the j -th diagonal element being the j -th element of α_i . With Kronecker product, the proofs are similar.

3.5 Numerical Examples

In this section, a RA problem which aims to allocate software-testing resource to improve software reliability [6] is solved by the proposed algorithms. Consider a software system composed of n modules which are tested independently. The objective is to identify as many errors as possible in a given time period T . The problem is formulated as [125]

$$\text{minimize} \quad \sum_{i=1}^n w_i N_i e^{-t_i/\theta_i} \quad (3.25a)$$

$$\text{subject to} \quad \sum_{i=1}^n t_i = T, \quad (3.25b)$$

$$t_i \geq 0, \quad i = 1, \dots, n, \quad (3.25c)$$

where t_i is the testing time of module i to be allocated, w_i is the weight measuring the criticality of a fault removal from module i , N_i is the mean number of errors remaining in module i , and θ_i is the mean detection time of a failure in module i .

Consider the example in [125] where there are five modules, and the amount of time to be allocated is $T = 5.5556$ h. The parameters adopted from [125] and the optimal solution are shown in Table 3.1.

Table 3.1: Parameters and optimal solution for the software-testing RA problem in [125]

Module	N_i	w_i	θ_i (h)	t_i^* (h)
1	21	1	1.6667	2.6215
2	15	1	1.2500	1.9051
3	5	1	1.3889	0.4446
4	10	1	3.3333	0.4593
5	6	1	2.1667	0.1251

To perform the distributed algorithm, a communication topology is imposed on the 5 modules. The communication graph is depicted in Figure. 3.1. The weights

Figure 3.1: Communication topology among the five modules.

associated with the edges of the communication graph are $a_{ij} = 1$ if $(v_i, v_j) \in \mathcal{E}$, otherwise, $a_{ij} = 0$ in the simulation.

Without considering the positivity constraint $t_i \geq 0$, algorithms (3.7) and (3.8) are used to solve the problem in (3.25). The control parameters are set as $\bar{k}_i = 100$ and $\bar{m}_0 = 0.1$. Module 1 is selected to calculate λ_0 . Figure. 3.2 plots the evolution of the allocated time t_i . It can be seen that the allocated time of each module converges to its optimal solution and the total allocation converges to the total time. Figure. 3.3 plots the evolution of λ_i s and z_i s. It can be seen that the λ_i s converge to a common value, which is the optimal Lagrange multiplier corresponding to the resource constraint, and the z_i s, which estimate the mismatch between total time and total allocation, converge to zero.

Taking the positivity constraint $t_i \geq 0$ into consideration, algorithms (3.15), (3.16), and (3.17) are used. In this simulation, the total time T is revised to $T = 4.4444$ h to show the effectiveness of local allocation feasibility constraint. As can be seen from [125], the solution without positivity constraint has negative t_5 . The control

Figure 3.2: Evolution of the allocated time t without positivity constraint.

Figure 3.3: Evolution of λ and z without positivity constraint.

Figure 3.4: Evolution of the allocated time t with positivity constraint.

parameters are set as $\bar{k}_i = 5$, $\bar{m}_0 = 0.2$, and $\bar{m}_{i1} = 0.01$. Figure. 3.4 shows that the allocated time of the modules converge to the new optimal solution, which are positive.

3.6 Conclusion

In this chapter, distributed algorithms based on dynamic average consensus, leader-following consensus, and the SPD are proposed to solve the RA problem. The proposed algorithms do not require any initialization procedure to design the initial values of the variables or find global information such as the total resource, total number of agents, and the entire communication topology. Based on the singular perturbation theory and the Lyapunov stability analysis, global exponential stability

Figure 3.5: Evolution of λ and z with positivity constraint.

is derived if the local resource allocation feasibility constraint is not considered, and practical stability is proven when the local allocation feasibility constraint is considered. A software reliability test resource allocation problem is adopted to verify the proposed algorithms. Simulation results show the effectiveness of the proposed algorithms.

Chapter 4

Distributed Economic Dispatch in Smart Grids

4.1 Introduction

The RA has a lot of applications in practical systems. As a typical example, the ED problem is investigated in this chapter. The ED problem addresses how to allocate the load demand to a set of generators in an economically efficient way while satisfying the generation load balance constraint and the generator capacity constraint. Nowadays, the traditional centralized power generation system is transforming to more distributed smart grids. A smart grid is generally defined as a power grid that integrates advanced monitoring technologies, information and communication technologies, and control methods to improve the efficiency and reliability of the power system [126]. In a smart grid, small generating units distributed in different locations work collaboratively to serve the load. Instead of using a central unit to coordinate the distributed generating units, using small controllers on each generating unit by communicating with their neighbors to achieve the overall goal is more promising. The distributed control architecture can improve the efficiency

and reliability of the system, usually has lower cost, and is more flexible with the plug-and-play operation of the grid [127].

To achieve the distributed coordination of the distributed generating units, various distributed algorithms have been proposed to solve the ED problem (e.g., [27, 28, 67, 68, 69, 70, 71, 72, 73, 75, 76, 77]). Among the proposed algorithms, a majority requires the balance between total load and total generation to be satisfied initially or an exact assign of the initial values of some variables. In this chapter, the algorithms developed in Chapter 3 are applied and extended to achieve a distributed ED without stringent initialization requirement. The main contributions of this chapter are summarized as follows.

1. The basic ED problem is solved in a distributed and initialization-free manner. No stringent requirement is enforced on the initial values of the variables. No private confidential information, such as the local loads and the incremental cost of the generating units, is required to be communicated with others. Furthermore, the cost functions of the generating units is not restricted to be quadratic, which can model a broader scope of generating units. In addition, the ED problem with line flow limits is solved in a distributed manner.
2. Numerous case studies on the IEEE 9-bus system are conducted to illustrate the proposed algorithms and simulation on the large-scale IEEE 118-bus system is performed to show that the algorithms are applicable to large-scale systems. Demonstration platform based on Matlab GUI is developed to demonstrate the distributed ED in an interactive way.

The rest of this chapter is organized as follows. Section 4.2 gives the system model of the distributed ED problem. The distributed algorithms developed in Chapter 3 are applied to solve the ED problem in Section 4.3. Furthermore, the algorithms are extended to solve the ED problem with line flow limits. Section 4.4 gives various cases studies conducted on the IEEE 9-bus system and the IEEE 118-bus system,

and demonstration platform is developed to demonstrate the distributed ED. Finally, the conclusion is given in Section 4.5.

4.2 System Model and Problem Formulation

Consider a power grid with distributed generators and distributed loads. Suppose there are n buses, without loss of generality, assume that each bus contains one generator (G) and one local load (L). Each bus is equipped with a local agent (A) which has the ability of communication and computation. The structure of a virtual agent is shown in Figure. 4.1. Each agent can communicate with its neighboring

Figure 4.1: Structure of a virtual agent composed of local generator, load, computation unit, and communication unit.

agents and is able to calculate the optimal generation reference for the generator. The distributed control architecture is illustrated in Figure. 4.2.

Figure 4.2: Illustration of the distributed control architecture.

The solid line indicates the physical connection among the buses, and the dashed line among the agents indicates the communication topology. The local agent gets

the local load and parameters of the local generator, does the calculation while communicating with its neighbors, and then sends the generation reference to the generator after finding the optimal generation.

Let P_{Gi} denote the active power generated by the generator in bus i and P_{Di} denote the active power needed by the load in bus i . The basic ED problem is formulated as

$$\begin{aligned} & \text{minimize } f(P_G) = \sum_{i=1}^n f_i(P_{Gi}), \\ & \text{subject to } \sum_{i=1}^n P_{Gi} = \sum_{i=1}^n P_{Di}, \\ & P_{Gi}^{\min} \leq P_{Gi} \leq P_{Gi}^{\max}, \quad i = 1, 2, \dots, n, \end{aligned} \tag{4.1}$$

where $P_G = [P_{Gi}]_{\text{vec}}$, $f_i(P_{Gi})$ is the cost function of the generator in bus i , and $P_{Gi}^{\min}$ and $P_{Gi}^{\max}$ are the lower and upper bounds of P_{Gi} with $P_{Gi}^{\min} < P_{Gi}^{\max}$. The cost function $f_i(P_{Gi})$ is assumed to be strictly convex. In practice, the cost function for a thermal generator is often modeled as a quadratic function given by [4]

$$f_i(P_{Gi}) = a_i P_{Gi}^2 + b_i P_{Gi} + c_i,$$

where $a_i > 0$, $b_i \geq 0$ and $c_i \geq 0$ are the cost parameters.

Remark 4.1. *If in bus i , there is no load, then $P_{Di} = 0$; if there are more than one load, they can be aggregated into one single load. If there is no generator in bus i , then $P_{Gi}(t) = 0 \forall t$; if there are more than one generator in one bus, they can be modeled as several buses with one generator located in each bus. For generators which provide constant generations, they can be treated as negative loads.*

4.3 Distributed Economic Dispatch

In this section, we apply the algorithms developed in Chapter 3 to solve the ED problem in (4.1). First, the basic ED problem is solved where the capacity limits of the generators are ignored first and then taken into consideration. Then, the ED problem with line flow limits is investigated. The distributed algorithms are given based on the algorithms developed in Chapter 3 with slight revision.

4.3.1 Distributed Algorithm for the Basic Economic Dispatch Problem

For the ED problem in (4.1), when the capacity limits are not considered, the algorithms (3.7) and (3.8) can be directly applied to derive the distributed algorithm that solves the ED problem as follows.

$$\dot{P}_{Gi} = -\bar{k}_i \left(\frac{df_i(P_{Gi})}{dP_{Gi}} - \lambda_i \right), \quad (4.2a)$$

$$\dot{\lambda}_0 = \bar{m}_0 z_q, \quad (4.2b)$$

$$\dot{z}_i = -z_i - \sum_{j=1}^n a_{ij}(z_i - z_j) - \sum_{j=1}^n a_{ij}(\theta_i - \theta_j) + P_{Di} - P_{Gi}, \quad (4.2c)$$

$$\dot{\theta}_i = \sum_{j=1}^n a_{ij}(z_i - z_j), \quad (4.2d)$$

$$\dot{\lambda}_i = -\sum_{j=1}^n a_{ij}(\lambda_i - \lambda_j) - a_{i0}(\lambda_i - \lambda_0). \quad (4.2e)$$

With the capacity limits of the generators considered, the distributed algorithm is derived based on (3.15) as

$$\dot{P}_{Gi} = -\bar{k}_i \left(\frac{df_i(P_{Gi})}{dP_{Gi}} - \lambda_i + \alpha_i - \beta_i \right), \quad (4.3a)$$

$$\dot{\lambda}_0 = \bar{m}_0 z_q, \quad (4.3b)$$

$$\dot{\alpha}_i = \bar{m}_{i1} \alpha_i (P_{Gi} - P_{Gi}^{\max}), \quad (4.3c)$$

$$\dot{\beta}_i = \bar{m}_{i2} \beta_i (P_{Gi}^{\min} - P_{Gi}), \quad (4.3d)$$

$$\dot{z}_i = -z_i - \sum_{j=1}^n a_{ij}(z_i - z_j) - \sum_{j=1}^n a_{ij}(\theta_i - \theta_j) + P_{Di} - P_{Gi}, \quad (4.3e)$$

$$\dot{\theta}_i = \sum_{j=1}^n a_{ij}(z_i - z_j), \quad (4.3f)$$

$$\dot{\lambda}_i = - \sum_{j=1}^n a_{ij}(\lambda_i - \lambda_j) - a_{i0}(\lambda_i - \lambda_0). \quad (4.3g)$$

In algorithm (4.3), the dynamic average consensus algorithm (4.3e) and (4.3f) is utilized to find the average value of the mismatch between total load and total generation. One agent q , which can be arbitrarily selected, calculates λ_0 by (4.3b) using z_q , and each agent calculates λ_i by the leader-following consensus algorithm (4.3g). With the estimated information, each agent updates P_{Gi} by (4.3a) using λ_i and other local information, and updates α_i and β_i by (4.3c) and (4.3d) to make P_{Gi} located within its capacity limits. The schematic outline of algorithm (4.3) is given in Figure. 4.3.

4.3.2 Distributed Algorithm for the Economic Dispatch Problem with Line Flow Limits

In this section, the line flow limits of the transmission lines are considered upon the basic ED problem. Suppose the transmission network is represented by graph $\mathcal{G}_t = (\mathcal{N}, \mathcal{E}_t)$ with $\mathcal{E}_t = \{1, \dots, m\}$ being the transmission lines set. $(i, j) \in \mathcal{E}_t$ means bus i and bus j are connected by a transmission line $l \in \mathcal{E}_t$. Arbitrarily assign direction to each line l as the reference power flow direction, define the incidence matrix $B \in \mathbb{R}^{n \times m}$ with $B_{il} = 1$ if line l origins from bus i , $B_{il} = -1$ if line l goes to

Figure 4.3: The Schematic outline of algorithm (4.3).

bus i , and $B_{il} = 0$, otherwise. The problem is formulated as [28]

$$\begin{aligned}
 & \text{minimize } f(P_G) = \sum_{i=1}^n f_i(P_{Gi}) \\
 & \text{subject to } P_{Gi} - \sum_{l=1}^m B_{il} P_{Ll} - P_{Di} = 0, \quad i = 1, \dots, n, \\
 & \quad P_{Gi}^{\min} \leq P_{Gi} \leq P_{Gi}^{\max}, \quad i = 1, \dots, n, \\
 & \quad |P_{Ll}| \leq P_{Ll}^{\max}, \quad l = 1, \dots, m,
 \end{aligned} \tag{4.4}$$

where P_{Ll} is the power flow on transmission line l , $P_{Ll}^{\max} > 0$ is the line flow limit of transmission line l . The solution of the problem in (4.4) is denoted by (P_G^*, P_L^*) with $P_L^* = [P_{Ll}^*]_{\text{vec}}$. Note that P_G^* is unique.

The augmented Lagrangian function of the problem in (4.4) is

$$L_3 = \sum_{i=1}^n (f_i(P_{Gi}) + \lambda_i(P_{Di} + \sum_{l=1}^m B_{il}P_{Ll} - P_{Gi}) + \alpha_i(P_{Gi} - P_{Gi}^{\max}) + \beta_i(P_{Gi}^{\min} - P_{Gi})) \\ + \sum_{l=1}^m (u_l(P_{Ll} - P_{Ll}^{\max}) + v_l(-P_{Ll} - P_{Ll}^{\max})) + \frac{1}{2} \sum_{i=1}^n (P_{Di} + \sum_{l=1}^m B_{il}P_{Ll} - P_{Gi})^2,$$

where $\lambda_i, \alpha_i, \beta_i, u_l, v_l \in \mathbb{R}$ are the Lagrange multipliers. By setting the communication graph $\mathcal{G}$ the same as the transmission graph $\mathcal{G}_t$, the distributed algorithm is designed as follows:

$$\dot{P}_{Gi} = -\kappa_{Gi} \left(\frac{df_i(P_{Gi})}{dP_{Gi}} - \lambda_i + \alpha_i - \beta_i - (P_{Di} + \sum_{l=1}^m B_{il}P_{Ll} - P_{Gi}) \right), \quad i = 1, \dots, n, \quad (4.5a)$$

$$\dot{P}_{Ll} = -\kappa_{Ll} \left(\sum_{i=1}^n B_{li}\lambda_i + u_l - v_l + \sum_{i=1}^n B_{li}(P_{Di} + \sum_{l=1}^m B_{il}P_{Ll} - P_{Gi}) \right), \quad l = 1, \dots, m, \quad (4.5b)$$

$$\dot{\lambda}_i = \kappa_{\lambda i} (P_{Di} + \sum_{l=1}^m B_{il}P_{Ll} - P_{Gi}), \quad i = 1, \dots, n, \quad (4.5c)$$

$$\dot{\alpha}_i = \kappa_{\alpha i} \alpha_i (P_{Gi} - P_{Gi}^{\max}), \quad i = 1, \dots, n, \quad (4.5d)$$

$$\dot{\beta}_i = \kappa_{\beta i} \beta_i (P_{Gi}^{\min} - P_{Gi}), \quad i = 1, \dots, n, \quad (4.5e)$$

$$\dot{u}_l = \kappa_{ul} u_l (P_{Ll} - P_{Ll}^{\max}), \quad l = 1, \dots, m, \quad (4.5f)$$

$$\dot{v}_l = \kappa_{vl} v_l (-P_{Ll} - P_{Ll}^{\max}), \quad l = 1, \dots, m, \quad (4.5g)$$

where $\kappa_{Gi}, \kappa_{Ll}, \kappa_{\lambda i}, \kappa_{\alpha i}, \kappa_{\beta i}, \kappa_{ul}, \kappa_{vl}$ are positive constants, $\alpha_i(0) > 0$, $\beta_i(0) > 0$, $u_l(0) > 0$, and $v_l(0) > 0$.

Assuming that the problem in (4.4) is feasible and the transmission graph is connected, under algorithm (4.5), P_G converges to P_G^* asymptotically.

Remark 4.2. Noticing that (x^*, y^*) , where $x^* = (P_G^*, P_L^*)$ and $y^* = (\lambda^*, \alpha^*, \beta^*, [u_l^*]_{vec}, [v_l^*]_{vec})$ being the optimal values of the Lagrange multipliers, is a saddle point of L_3 and $L_3(x, y^*) = L_3(x^*, y^*)$ implies (x, y^*) is a saddle point of L_3 [94]. By following

the proof of the stability of the reduced system in Theorem 3.2, the conclusion is obtained.

4.4 Case Studies

4.4.1 Implementation on the IEEE 9-bus Test System

To verify the proposed algorithms, simulation is conducted on the IEEE 9-bus system shown in Figure. 4.4. Various case studies include load change, non-quadratic cost function and constant generation, communication link loss, and plug and play test are conducted.

Figure 4.4: Illustration of the IEEE 9-bus system.

In the IEEE 9-bus system, there are 9 buses, 3 generators, and 3 loads. The power demand of loads located in bus 5, 6, 8 are 200 MW, 550 MW and 100 MW, respectively. The parameters of the cost functions and capacity limits of the generators are given in Table 4.1 [71]. By centralized calculation, the optimal generations can be obtained as $P_{G1}^* = 393.1698$ MW, $P_{G2}^* = 334.6038$ MW, $P_{G3}^* = 122.2264$ MW,

Table 4.1: Generator Parameters

P_G	a_i	b_i	c_i	$P_{Gi}^{\min}(\text{MW})$	$P_{Gi}^{\max}(\text{MW})$
P_{G1}	0.001562	7.92	561	150	600
P_{G2}	0.00194	7.85	310	100	400
P_{G3}	0.00482	7.97	78	50	200

and $\lambda_0^* = 9.1483$. The communication topology is set the same as the physical connection of the buses shown in Figure. 4.5.

Figure 4.5: Communication topology of the IEEE 9-bus system.

The squares represent agents contain generators, the triangles represent agents contain loads, and the circles represent agents contains no generators or loads, which are used to calculate intermediate variables and propagate information. Agent 4 is selected to calculate λ_0 . The initial values of the variables are randomly generated except that $\alpha(0) > \mathbf{0}_n$ and $\beta(0) > \mathbf{0}_n$.

4.4.1.1 Without Capacity Limits

In this simulation, the capacity limits are not considered. The parameters are set as $\bar{k}_i = 1000$ and $\bar{m}_0 = 0.003$. The simulation results produced by (4.2) are shown in Fig. 4.6.

The upper figure shows the updating of the generators' power generations, which converge to the optimal generations. The lower figure shows the updating of the

Figure 4.6: Power generations of the generators and estimated incremental costs without capacity limits produced by (4.2). The dashed lines denote the optimal values.

estimated incremental costs, which converge to λ_0^* .

4.4.1.2 With Capacity Limits

In this simulation and thereafter, the generator capacity limits are considered and the updating law used is (4.3). The parameters are set as $\bar{k}_i = 1000$, $\bar{m}_0 = 0.003$, and $\bar{m}_{i1} = \bar{m}_{i2} = 1$. The results given in Fig. 4.7 show that the solutions converge to their optimal values.

To check the effectiveness of the proposed algorithm, the upper capacity limit of generator 3 is revised to 80 MW, which is lower than $P_{G3}^* = 122.2264$ MW. In this situation, the new optimal solutions are $P_{G1}^* = 416.5620$ MW, $P_{G2}^* = 353.4380$ MW,

Figure 4.7: Power generations and estimated incremental costs with capacity limits produced by (4.3). The dashed lines denote the optimal values.

$P_{G3}^* = 80.0000$ MW, and $\lambda_0^* = 9.2213$ by direct calculation. The simulation results are shown in Fig. 4.8. The upper figure shows that generator 3's final generation decreases and equals 80MW, which is its upper bound. The lower figure shows that the incremental costs converge to the new optimal incremental cost, which increases due to lower power generation of the generator 3.

4.4.1.3 Comparison with Methods Require Initialization

In this section, the algorithms proposed in [69] (Yang 2013), [70] (Xing 2015), [72] (Li 2015), [71] (Zhang 2016), [68] (Kar 2012), and [77] (Wang 2019) are tested on the IEEE 9-bus system to show the importance of initial condition required in these algorithms. Since we care about the initial condition, the control parameters in these

Figure 4.8: Power generations and estimated incremental costs produced by (4.3) when capacity limits are contradicted. The dashed lines denote the new optimal values.

algorithms are select only to ensure the convergence regardless of the convergence rate. Figure. 4.9 shows the simulation results. The solid lines represent the results when the initial conditions are satisfied and the dashed lines represent the results when the initial conditions are not satisfied. It can be seen that when the initial conditions of these methods are not satisfied, there will be a steady-state error between the total power generation and the total load demand.

4.4.1.4 Load Change

In this simulation, the loads located in bus 8 is increased from 100 MW to 300 MW at $t = 50s$ and recovered to 100 MW at $t = 100s$.

(a) (Yang 2013)

(b) (Xing 2015)

(c) (Li 2015)

(d) (Zhang 2016)

(e) (Kar 2012)

(f) (Wang 2019)

Figure 4.9: Power generations produced by (a) (Yang 2013), (b) (Xing 2015), (c) (Li 2015), (d) (Zhang 2016), (e) (Kar 2012), and (f) (Wang 2019). The solid lines represent the results when the initial conditions are satisfied, the dashed lines represent the results when the initial conditions are not satisfied.

Figure. 4.10 shows that after the load increases at $t = 50s$, the power generations asymptotically increase to the new optimal generations, and the total generation

Figure 4.10: Power generations and estimated incremental costs produced by (4.3) with load changes.

converges to the new total load. After the load recovers at $t = 100s$, the generations recover to their previous values. The estimated incremental costs change as the load changes.

4.4.1.5 Non-quadratic Cost Function and Constant Generation

In this section, the generator located in bus 2 is assumed to have a non-quadratic but strictly convex cost function, which is given by [4]

$$f_2(P_{G2}) = a_2 P_{G2}^2 + b_2 P_{G2} + c_2 + 100e^{\frac{P_{G2}}{100}}.$$

The generator located in bus 3 is assumed to have a fixed generation of $P_{G3} = 150$ MW. In this case, the optimal generations are $P_{G1}^* = 600$ MW, $P_{G2}^* = 100$ MW and

$P_{G3}^*=150$ MW by direct calculation. The result is shown in Figure. 4.11.

Figure 4.11: Power generation produced by (4.3) with non-quadratic cost function and fixed generation.

4.4.1.6 Communication Link Loss

In this part, loss of communication link which does not break the connectivity of the communication graph is tested. Figure. 4.12 compares the updating of the generations without and with loss of communication link between bus 6 and bus 9. The result shows that loss of a communication link, which does not break the connectivity of the communication graph, does not influence the convergency of the algorithm. However, the converge rate is different under different communication topologies.

Figure 4.12: Power generation produced by (4.3) with one communication link loss.

4.4.1.7 Plug and Play Test

To test the plug and play adaptability, one generator is plugged into the IEEE 9-bus system at $t = 50s$. The new communication topology is given in Figure. 4.13. The

Figure 4.13: Communication topology with one generator plugged-in.

new plugged generator is located in bus 10, which is connected to bus 3 and bus 6. The parameters of the generator located in bus 10 are set the same as those of the generator located in bus 1. The simulation result is shown in Figure. 4.14. It can be seen that after the new generator is plugged in, the generations of the

Figure 4.14: Power generation produced by (4.3) with one generator plugged-in.

original generators drop down, and the total generation equals the total load. Since the generator located in bus 10 has the same parameters as those of the generator located in bus 1, they have the same optimal generations as shown in Figure. 4.14.

4.4.1.8 With Line Flow Limits

In this section, the line flow limits are imposed on the IEEE 9-bus system. From the previous calculation, the optimal generations of the three generators located in buses 1, 2, and 3 are $P_{G1}^* = 393.1698$ MW, $P_{G2}^* = 334.6038$ MW, and $P_{G3}^* = 122.2264$

Figure 4.15: Power generations and power flows produced by (4.5) with line flow limits.

MW, respectively. To see the influence of the line flow limits, we impose a line flow limit of 250 MW (smaller than P_{G2}^*) on the transmission line which connects bus 2 and bus 7, and 500 MW on other transmission lines. By direct calculation, the optimal generations of the three generators with line flow limits are $P_{G1}^* = 457.0673$ MW, $P_{G2}^* = 250.0$ MW, and $P_{G3}^* = 142.9327$ MW, respectively. By using algorithm (4.5) and setting the control gains as $\kappa_{Gi} = \kappa_{Li} = 1000$, $\kappa_{\lambda i} = 1$, and $\kappa_{\alpha i} = \kappa_{\beta i} = \kappa_{ul} = \kappa_{vl} = 0.02$, we get the simulation results shown in Fig. 4.15. The upper figure shows that the generations converge to the new optimal generations. The lower figure shows that the power flow on all transmission lines are within their line flow limits, especially P_{L27} , which is the power flow on line connected bus 2 and 7, reaches its upper limit as expected.

4.4.2 Implementation on the IEEE 118-bus Test System

In this section, simulation on the IEEE 118-bus system is conducted to test the performance of the proposed algorithm. There are 118 buses and 54 generators. The data used is adopted from [128] and the communication graph is set the same as the physical connection of the buses. Agent 49 is selected to calculate λ_0 . The parameters are set as $\bar{k}_i = 2300$ and $\bar{m}_0 = \bar{m}_{i1} = \bar{m}_{i2} = 0.15$. The simulation results in Fig. 4.16 show that the algorithm is still valid for this large-scale power system.

Figure 4.16: Power generations and estimated incremental costs produced by (4.3) tested on the IEEE 118-bus system.

Figure 4.17: User interface of the distributed ED developed by Matlab GUI.

4.4.3 Demonstration through Matlab GUI

To demonstrate the proposed algorithms, we developed platform based on Matlab GUI to show the distributed ED algorithm (4.3) tested on the IEEE 9-bus system with a 24-hour hourly load profile.

The user interface is shown in Figure. 4.17. In the platform, the initial 24-hour hourly load profile is set based on the prediction of the load from historical load data. Due to some circumstance change, such as sudden weather change, the prediction of the load may be inaccurate. An adjustment should be allowed as the time approaches to give a more accurate load prediction. In the platform, the load is updated at the beginning of each hour, and it can be revised manually before the time arrives. By receiving the updated load, the algorithm calculates the optimal generation. Meanwhile, the parameters of the generator cost functions can be revised to represent different kind of generators or different work status.

Furthermore, when there is a high load burden, one generator G4 can be plugged-in to relieve the system pressure. When the load decreases, the generator can be plugged-off.

4.5 Conclusion

In this chapter, we investigated the ED problem in smart grids. The distributed algorithms developed in Chapter 3 are applied to solve the ED problem in a distributed manner. Furthermore, with slight revision, the algorithm is capable to solve the ED problem with line flow limits in a distributed way under the condition that the communication topology is set the same as the physical connection of the buses. Numerous case studies are conducted on the IEEE 9-bus system and the IEEE 118-bus system to show the effectiveness of the proposed algorithms. Demonstration platforms are developed based on Matlab GUI and R, respectively, on the IEEE 9-bus system to demonstrate the proposed algorithms in an interactive way.

Chapter 5

Distributed Time-varying Resource Allocation and Its Application to Economic Dispatch with Time-varying Loads

5.1 Introduction

The resources often have time-varying nature in practice. Take the loads in the ED problem as an example, they are varying from time to time due to the consumption and renewable energy sources, which can be seen by historical data of loads profile [129]. In our previous work handling ED problem, the loads are treated static constants during a time period [130]. To improve the economic efficiency, we focus on solving the RA problem with continuously TVR in a distributed manner in this chapter.

The RA problem with TVR is fundamentally a TVO problem. With the TVR, the optimal solution varies from time to time. Compared with time-invariant op-

timization problems, solving TVO problems is more challenging since the optimal solution is time-varying, which requires the algorithm to not only find but also track it. Furthermore, unlike the general tracking problem, the error signal is not directly available to design the algorithm.

There are limited results on TVO. The problem was first regarded as non-stationary optimization problems. For an optimization problem with a time-varying objective function, an updating law based on the inverse of the Hessian matrix and the derivative of the gradient with optimal starting point is able to find and track the optimal solution [131]. Recently, a series of prediction-correction methods which depend on the Hessian matrix, the gradient, and the derivative of the gradient were proposed to solve the single-agent based TVO problems [79, 80, 81]. Among these works, no constraint is considered. For constrained TVO problems, an approximated solution was obtained in [84] by an iterative algorithm which were refined every time a new sampled time-invariant problem is available. [85] solved linear constrained TVO problem with discrete-time setting. [82] utilized interior-point method to handle the constraints and the TVO problem is solved with a prediction-correction scheme.

Different from single-agent or centralized TVO, designing distributed algorithm is more challenging since only local information can be used, which restricts the access to the inverse of the Hessian matrix. Distributed TVO were proposed for networked problems in [88, 87, 89, 90], where no global constraint is considered. In this chapter, distributed algorithms are developed for the RA problem with continuously TVR, which is a TVO problem with global constraint. The contributions are summarized as follows.

1. The RA problem with TVR is investigated. The existence of the solution to the RA problem with continuously TVR is proven and the condition which the optimal solution should satisfy is given.
2. Distributed approaches are developed to solve the problem. The RA problem inherently has a global resource constraint. In the existing distributed meth-

ods that deal with TVO problems, only a few literatures consider constraints. Local allocation feasibility constraint and networked constraint are studied in [89, 90], which are relatively easy to handle in a distributed way compared with the global resource constraint. For the case under quadratic objective functions without local allocation feasibility constraint, a distributed algorithm is developed from the primal dual viewpoint, and the sign function is utilized to tracking the time-varying optimal solution. The tracking error is proven to converge to zero in finite time. For the case under strongly convex objective functions with local allocation feasibility constraint, a distributed algorithm is developed based on the singular perturbation theory, the tracking error is proven to be uniformly ultimately bounded.

3. The ED problem with continuously time-varying loads is solved by the proposed distributed algorithms. Simulation on a 24-bus system shows the efficacy of the proposed algorithms.

The rest of this chapter is organized as follows. Section 5.2 gives the problem formulation and some general assumptions. Section 5.3 proposes the distributed algorithms for the RA problem with continuously time-varying resources and gives a numerical example. Application on the ED problem with time-varying loads is given in Section 5.4, where a simulation on the 24-bus system is conducted. Section 5.5 gives the conclusion.

5.2 Problem Formulation

Consider a RA problem with time-varying local resources over n agents. Each agent is associated with a continuous decision variable $x_i \in \mathbb{R}$, a corresponding strongly convex cost function $f_i(x_i) \in \mathbb{R}$, and a local time-varying resource $d_i(t) \in \mathbb{R}$. The resource constraint is $\sum_{i=1}^n x_i = \sum_{i=1}^n d_i(t)$. In some problems, the allocation of agent i is required to satisfy the local allocation feasibility constraint $x_i^{\min} \leq$

$x_i \leq x_i^{\max}$, where $x_i^{\min} \in \mathbb{R}$ and $x_i^{\max} \in \mathbb{R}$ are the lower and upper bounds of x_i , respectively. Denote $x = [x_i]_{\text{vec}}$, $x^{\min} = [x_i^{\min}]_{\text{vec}}$ and $x^{\max} = [x_i^{\max}]_{\text{vec}}$. The RA problem with TVR is formulated as [6]

$$\text{minimize } f(x) = \sum_{i=1}^n f_i(x_i) \quad (5.1a)$$

$$\text{subject to } \sum_{i=1}^n x_i = \sum_{i=1}^n d_i(t), \quad (5.1b)$$

$$x^{\min} \leq x \leq x^{\max}. \quad (5.1c)$$

The communication network among the n agents is modeled as an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Throughout this chapter, the following assumptions hold.

Assumption 5.1. *The communication graph is undirected and connected.*

The connectivity of the undirected communication graph guarantees that the information can be propagated throughout the system.

Assumption 5.2. *The derivative of $d_i(t)$ exists, and $d_i(t)$ and its derivative are bounded for all $i = 1, \dots, n$.*

Assumption 5.2 ensures that the optimal solution of the problem in (5.1) and its derivative are bounded, which is trivial in tracking problems [132, 133]. The smoothness of the time-varying resources can be ensured by preprocessing the resources with some smoothing methods.

Assumption 5.3. *The problem in (5.1) is feasible all the time, i.e.,*

$$\sum_{i=1}^n x_i^{\min} \leq \sum_{i=1}^n d_i(t) \leq \sum_{i=1}^n x_i^{\max}, \quad \forall t \geq 0.$$

5.3 Distributed Resource Allocation with Time-varying Resources

5.3.1 Without Local Allocation Feasibility Constraint under Quadratic Objective Functions

In this section, the problem in (5.1) without local allocation feasibility constraint (5.1c) under quadratic objective functions is investigated. The objective function $f_i(x_i)$ is assumed to be quadratic defined as

$$f_i(x_i) = a_i x_i^2 + b_i x_i + c_i, \quad (5.2)$$

where $a_i > 0$, b_i and c_i are the objective function parameters. The optimal solution mentioned in this section refers to the optimal solution of (5.1) without (5.1c) under (5.2).

For constant d_i , the optimal solution exists, is unique, and satisfies the Lagrange multiplier optimality condition. For time-varying variable $d_i(t)$, the following lemma provides the existence and uniqueness of the solution and gives the condition that the optimal solution satisfies, which is similar to the Lagrange multiplier optimality condition.

Lemma 5.1. *Under Assumption 5.2, for the problem in (5.1) without local allocation feasibility constraint under quadratic objective functions, the optimal solution $x^*(t)$ exists, and there exists $\lambda_c^*(t) \in \mathbb{R}$ such that the following equations are satisfied,*

$$\nabla_x f(x^*(t)) - \mathbf{1}_n \lambda_c^*(t) = \mathbf{0}_n, \quad (5.3a)$$

$$\sum_{i=1}^n d_i(t) - \sum_{i=1}^n x_i^*(t) = 0. \quad (5.3b)$$

Furthermore, $(x^*(t), \lambda_c^*(t))$ is unique.

Proof. Define

$$L_1(x, \lambda_c, t) = f(x) + \lambda_c \left(\sum_{i=1}^n d_i(t) - \sum_{i=1}^n x_i \right),$$

where $\lambda_c \in \mathbb{R}$. For fixed time $t_f \in [0, t]$, it is well known that there exists a unique pair (x_f^*, λ_{cf}^*) , where x_f^* is the optimal solution and λ_{cf}^* is the optimal value of the Lagrange multiplier corresponding to the equality constraint at t_f , such that [15]

$$\begin{aligned} \nabla_x f(x_f^*) - \mathbf{1}_n \lambda_{cf}^* &= \mathbf{0}_n, \\ \sum_{i=1}^n d_i(t_f) - \sum_{i=1}^n x_{fi}^* &= 0. \end{aligned}$$

Define $\phi = [x^T \ \lambda_c]^T$ and $\phi_f^* = [x_f^{*T} \ \lambda_{cf}^*]^T$. The above equations are equivalent to

$$\nabla_\phi L_1(\phi_f^*, t_f) = \mathbf{0}_{n+1},$$

It can be easily verified that the following conditions are satisfied: (1) $\nabla_\phi L_1(\phi, t)$ is continuous with respect to (ϕ, t) in a neighborhood of (ϕ_f^*, t_f) ; (2) $\nabla_\phi L_1(\phi, t)$ is differentiable in ϕ in a neighborhood of (ϕ_f^*, t_f) ; (3) $\nabla_\phi^2 L_1(\phi, t)$ is continuous at (ϕ_f^*, t_f) and is nonsingular. By the implicit function theorem [131], there exists a unique differentiable function $\phi^*(t)$ in a neighborhood of t_f such that $\phi^*(t_f) = \phi_f^*$ and

$$\nabla_\phi L_1(\phi^*(t), t) = \mathbf{0}_{n+1},$$

which is equivalent to

$$\begin{aligned} \nabla_x f(x^*(t)) - \mathbf{1}_n \lambda_c^*(t) &= \mathbf{0}_n, \\ \sum_{i=1}^n d_i(t) - \sum_{i=1}^n x_i^*(t) &= 0. \end{aligned}$$

Thus, $x^*(t)$ is in the feasible region of the problem in (5.1) without local allocation

feasibility constraint, and

$$\nabla_x f(x^*(t)) = \mathbf{1}_n \lambda_c^*(t).$$

Selecting $y(t) = [y_i(t)]_{\text{vec}} \in \mathbb{R}^n$ such that $\sum_{i=1}^n d_i(t) - \sum_{i=1}^n y_i(t) = 0$, we have

$$\begin{aligned} & [\nabla_x f(x^*(t))]^T (y(t) - x^*(t)) \\ &= \lambda_c^*(t) \mathbf{1}_n^T (y(t) - x^*(t)) \\ &= 0. \end{aligned}$$

Thus, the first-order optimality condition is satisfied, which indicates that $x^*(t)$ is the optimal solution. By the uniqueness of ϕ_f^* and $\phi^*(t)$ in a neighborhood of t_f , $(x^*(t), \lambda_c^*(t))$ is unique. $\square$

In (5.3), $\lambda_c^*(t)$ is common information shared by all the agents. If we can find $\lambda_i^*(t)$ for agent i such that $\lambda_i^*(t) = \lambda_c^*(t)$, $i \in \{1, \dots, n\}$, then each agent can use its local $\lambda_i^*(t)$.

Lemma 5.2. *Under Assumptions 5.1-5.2, if $(x^*(t), \lambda^*(t), \mu^*(t))$ is the solution of the following equations*

$$2ax^*(t) + b - \lambda^*(t) = \mathbf{0}_n, \quad (5.4a)$$

$$d(t) - x^*(t) - L\mu^*(t) = \mathbf{0}_n, \quad (5.4b)$$

$$L\lambda^*(t) = \mathbf{0}_n, \quad (5.4c)$$

where $a = \text{diag}\{a_i\}$, $b = [b_i]_{\text{vec}}$, $\lambda^*(t) = [\lambda_i^*(t)]_{\text{vec}}$, and $d(t) = [d_i(t)]_{\text{vec}}$, then, $x^*(t)$ is the optimal solution.

Proof. Since the graph $\mathcal{G}$ is undirected and connected, (5.4c) implies

$$\lambda_1^*(t) = \dots = \lambda_n^*(t). \quad (5.5)$$

Multiplying both sides of (5.4b) by $\mathbf{1}_n^T$ on the left gives

$$\mathbf{1}_n^T(d(t) - x^*(t)) = \mathbf{1}_n^T L \mu^*(t) = 0. \quad (5.6)$$

Solving (5.4a) under the constraints of (5.5) and (5.6) gives the same solution with (5.3), which indicates $\lambda_i^* = \lambda_c^*$ and $x^*(t)$ is the optimal solution. $\square$

The algorithm (5.8)-(5.9) is designed based on the primal dual viewpoint. The Lagrangian dual problem for the problem in (5.1) without (5.1c) is [15]

$$\text{maximize } g(\lambda_c) = \inf_x L_1(x, \lambda_c, t).$$

As we mentioned above, if we can find $\lambda_1 = \dots = \lambda_n = \lambda_c$, then, the objective function can be decoupled as the summation of local objective functions. The Lagrangian dual problem is equivalent to the following constrained optimization problem,

$$\text{maximize } g(\lambda) = \inf_x L_1(x, \lambda, t), \quad (5.7a)$$

$$\text{subject to } L\lambda = \mathbf{0}_n. \quad (5.7b)$$

The Lagrangian function of the constrained dual problem (5.7) is

$$L_d(x, \lambda, \mu, t) = \inf_x L_1(x, \lambda, t) + \mu^T L\lambda,$$

where $\mu \in \mathbb{R}^n$ is the Lagrange multiplier corresponding to (5.7b). Based on the gradient of L_d and to track the time-varying solution, the distributed algorithm is designed as

$$\begin{aligned} \dot{x}_i &= -k_a(2a_i x_i + b_i - \lambda_i) - k_b(2a_i \text{sgn}_{1i} + \text{sgn}_{2i}), \\ \dot{\lambda}_i &= k_a(d_i(t) - x_i - \sum_{j=1}^n a_{ij}(\mu_i - \mu_j)) + k_b(\text{sgn}_{1i} + \sum_{j=1}^n a_{ij}(\text{sgn}_{3i} - \text{sgn}_{3j})), \\ \dot{\mu}_i &= k_a \sum_{j=1}^n a_{ij}(\lambda_i - \lambda_j) - k_b \sum_{j=1}^n a_{ij}(\text{sgn}_{2i} - \text{sgn}_{2j}), \end{aligned}$$

with

$$\begin{aligned}\text{sgn}_{1i} &= \text{sgn}(2a_i x_i + b_i - \lambda_i), \\ \text{sgn}_{2i} &= \text{sgn}\left(x_i + \sum_{j=1}^n a_{ij}(\mu_i - \mu_j) - d_i(t)\right), \\ \text{sgn}_{3i} &= \text{sgn}\left(-\sum_{j=1}^n a_{ij}(\lambda_i - \lambda_j)\right),\end{aligned}$$

where k_a and k_b are positive constants. Writing the algorithm in a compact form gives

$$\dot{x} = -k_a(2ax + b - \lambda) - k_b(2a\text{sgn}_1 + \text{sgn}_2), \quad (5.8a)$$

$$\dot{\lambda} = k_a(d(t) - x - L\mu) + k_b(\text{sgn}_1 + L\text{sgn}_3), \quad (5.8b)$$

$$\dot{\mu} = k_a L\lambda - k_b L\text{sgn}_2, \quad (5.8c)$$

with

$$\text{sgn}_1 = \text{sgn}(2ax + b - \lambda), \quad (5.9a)$$

$$\text{sgn}_2 = \text{sgn}(x + L\mu - d(t)), \quad (5.9b)$$

$$\text{sgn}_3 = \text{sgn}(-L\lambda). \quad (5.9c)$$

The pseudocode of (5.8)-(5.9) is given below. The stability of the algorithm is given

Algorithm 1: Pseudocode of Algorithm (5.8)-(5.9)

Input : Coefficients of cost functions a_i, b_i , and local resource $d_i(t)$ for $i = 1, \dots, n$.

Output : Allocated resource x_i^* for $i = 1, \dots, n$.

Initialize: Generate random initial values of $x(0)$, $\lambda(0)$, and $\mu(0)$. Set control parameters k_a and k_b , and weights of the communication graph a_{ij} .

```

1 while not converge do
2   | Calculate  $\text{sgn}_1, \text{sgn}_2$ , and  $\text{sgn}_3$  according to (5.9);
3   | Update  $x, \lambda$ , and  $\mu$  according to (5.8);
4 end
```

as follows.

Theorem 5.1. *Under Assumptions 5.1-5.2, if the agents adopt (5.8)-(5.9) to update the decision variables and k_b is set to be sufficiently large, then $x \rightarrow x^*(t)$ in finite time.*

Proof. Let $U = [U_1 \ U_2] \in \mathbb{R}^{n \times n}$ with $U_1 \in \mathbb{R}^{n \times (n-1)}$ and $U_2 \in \mathbb{R}^n$ be an orthogonal matrix such that $U_2^T L = \mathbf{0}_n^T$. Define $\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} = U^T \mu = \begin{bmatrix} U_1^T \mu \\ U_2^T \mu \end{bmatrix}$, and thus, $\mu = U \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} = U_1 \mu_1 + U_2 \mu_2$. Then, (5.8)-(5.9) can be written as

$$\begin{aligned} \dot{x} &= -k_a(2ax + b - \lambda) - k_b(2\text{sgn}(2ax + b - \lambda) + \text{sgn}(x + LU_1 \mu_1 - d(t))), \\ \dot{\lambda} &= k_a(d(t) - x - LU_1 \mu_1) + k_b(\text{sgn}(2ax + b - \lambda) + L\text{sgn}(-L\lambda)), \\ \dot{\mu}_1 &= k_a U_1^T L \lambda - k_b U_1^T L \text{sgn}(x + LU_1 \mu_1 - d(t)), \\ \dot{\mu}_2 &= 0. \end{aligned}$$

Let $\phi = [x^T \ \lambda^T \ \mu_1^T]^T$. Then, we have

$$\dot{\phi} = k_a(A\phi + e) - k_b \bar{A}^T \text{sgn}(\bar{A}\phi + \begin{bmatrix} e \\ 0 \end{bmatrix}),$$

where

$$\begin{aligned} A &= \begin{bmatrix} -2a & I & \mathbf{0}_{n \times (n-1)} \\ -I & \mathbf{0}_{n \times n} & -LU_1 \\ \mathbf{0}_{(n-1) \times n} & U_1^T L & \mathbf{0}_{(n-1) \times (n-1)} \end{bmatrix}, \\ \bar{A} &= \begin{bmatrix} -2a & I & \mathbf{0}_{n \times (n-1)} \\ -I & \mathbf{0}_{n \times n} & -LU_1 \\ \mathbf{0}_{n \times n} & L & \mathbf{0}_{n \times (n-1)} \end{bmatrix}, \end{aligned}$$

$$e = \begin{bmatrix} -b \\ d(t) \\ \mathbf{0}_{n-1} \end{bmatrix}.$$

Since $\begin{bmatrix} -2a & I \\ -I & \mathbf{0}_{n \times n} \end{bmatrix}$ is Hurwitz and $\begin{bmatrix} \mathbf{0}_{n \times (n-1)} \\ -LU_1 \end{bmatrix}$ has full column rank, according to Lemma 2.3, A is Hurwitz. Since $\bar{A}$ has full column rank, $\bar{A}^T \bar{A}$ is positive definite.

Let $(x^*(t), \lambda^*(t), \mu^*(t))$ be the solution of (5.4). By Lemma 5.2, $x^*(t)$ is the optimal solution. Let $\begin{bmatrix} \mu_1^*(t) \\ \mu_2^*(t) \end{bmatrix} = U^T \mu^*(t)$, and $\phi^* = [x^{*T}(t) \ \lambda^{*T}(t) \ \mu_1^{*T}(t)]^T$. By (5.4), it can be derived that

$$\begin{aligned} A\phi^* + e &= \mathbf{0}_{3n-1}, \\ \bar{A}\phi^* + \begin{bmatrix} e \\ 0 \end{bmatrix} &= \mathbf{0}_{3n}. \end{aligned}$$

Since A is nonsingular, ϕ^* is unique and is a linear function of e . According to Assumption 5.2, $\|\dot{\phi}^*\|$ is bounded. Denote the bound of $\|\dot{\phi}^*\|$ as γ . Let $\tilde{\phi} = \phi - \phi^*$. Then, it can be derived that

$$\dot{\tilde{\phi}} = k_a A \tilde{\phi} - k_b \bar{A}^T \text{sgn}(\bar{A} \tilde{\phi}) - \dot{\phi}^*. \quad (5.10)$$

Define $V = \frac{1}{2} \tilde{\phi}^T \tilde{\phi}$. Calculating the derivative of V along (5.10) gives

$$\begin{aligned} \dot{V} &= \tilde{\phi}^T \dot{\tilde{\phi}} \\ &= \frac{1}{2} k_a \tilde{\phi}^T (A^T + A) \tilde{\phi} - k_b \tilde{\phi}^T \bar{A}^T \text{sgn}(\bar{A} \tilde{\phi}) - \tilde{\phi}^T \dot{\phi}^* \\ &\leq -2k_a \tilde{x}^T a \tilde{x} - k_b \|\bar{A} \tilde{\phi}\| + \gamma \|\tilde{\phi}\| \\ &\leq -2k_a \tilde{x}^T a \tilde{x} - (k_b \sqrt{\lambda_{\min}(\bar{A}^T \bar{A})} - \gamma) \|\tilde{\phi}\|. \end{aligned}$$

By selecting k_b sufficiently large such that $k_b\sqrt{\lambda_{\min}(\bar{A}^T\bar{A})} - \gamma \triangleq \delta > 0$, we have

$$\begin{aligned}\dot{V} &\leq -2k_a\tilde{x}^T a\tilde{x} - \delta\|\tilde{\phi}\| \\ &< -\delta\sqrt{2}V^{\frac{1}{2}}.\end{aligned}$$

Thus, $\tilde{\phi} \rightarrow \mathbf{0}_{3n-1}$ in finite time [134], which indicates $x \rightarrow x^*$ in finite time, and the settling time T can be estimated as $T < \frac{1}{\delta}\|\tilde{\phi}(0)\|$. $\square$

Remark 5.1. *The value of k_b only needs to satisfy $k_b\sqrt{\lambda_{\min}(\bar{A}^T\bar{A})} - \gamma > 0$, where $\bar{A}$ and γ are defined in the proof of Theorem 5.1.*

5.3.2 With Local Allocation Feasibility Constraint under Strongly Convex Objective Functions

For quadratic convex objective functions, the second derivative of the objective functions are positive constants, which are relatively easier to handle [135]. However, in some problems, the object of the agent is formulated by non-quadratic function. In this section, the objective function is assumed to be strongly convex instead of quadratic and the local allocation feasibility constraint is considered. The following assumption is imposed.

Assumption 5.4. *The objective functions $f_i(x_i)$ for $i = 1, \dots, n$ are twice differentiable and strongly convex.*

Since $f_i(x_i)$ is twice differentiable and strongly convex, there exists positive constant m_i such that $\forall x_i$

$$\nabla^2 f_i(x_i) \geq m_i, \quad i \in \{1, \dots, n\}.$$

Penalty functions are often used to deal with inequality constraints by incorporating the inequality constraints into the objective function. To deal with the local

allocation feasibility constraint and gain access to the gradient based minimization techniques, a smooth penalty function $p_i : \mathbb{R} \rightarrow \mathbb{R}$ is used to punish the violation on the constraint $y \leq 0$ [136]

$$p_i(y) = \begin{cases} 0, & \text{if } y \leq 0, \\ \kappa \frac{y^2}{2\epsilon}, & \text{if } 0 < y \leq \epsilon, \\ \kappa(y - \frac{\epsilon}{2}), & \text{if } y > \epsilon, \end{cases}$$

where κ is the penalty parameter and ϵ is the smoothing parameter, which are all positive constants.

Reformulate the problem in (5.1) as

$$\begin{aligned} & \text{minimize } f_p(x) = \sum_{i=1}^n f_i(x_i) + \sum_{i=1}^n (p_i(x_i - x_i^{\max}) + p_i(x_i^{\min} - x_i)) \\ & \text{subject to } \sum_{i=1}^n x_i = \sum_{i=1}^n d_i(t). \end{aligned} \quad (5.11)$$

Correspondingly, each agent has the modified local cost

$$f_{pi}(x_i) = f_i(x_i) + p_i(x_i - x_i^{\max}) + p_i(x_i^{\min} - x_i).$$

Since $f_i(x_i)$ is twice differentiable and strongly convex, f_{pi} is twice differentiable and strongly convex in x_i . Denote the optimal solution of the problem in (5.11) as x_p^* . Similar to Lemma 5.1 and Lemma 5.2, there exists a λ^* such that the following conditions are satisfied

$$\begin{aligned} \nabla f_p(x_p^*) - \lambda^* &= \mathbf{0}_n, \\ \mathbf{1}_n^T (d(t) - x_p^*) &= 0, \\ L\lambda^* &= \mathbf{0}_n. \end{aligned} \quad (5.12)$$

For the problem in (5.1), the Lagrangian function is

$$\begin{aligned} L_2(x, \lambda_c, t, \alpha, \beta) = & f(x) + \lambda_c \left(\sum_{i=1}^n d_i(t) - \sum_{i=1}^n x_i \right) + \sum_{i=1}^n \alpha_i (x_i - x_i^{\max}) \\ & + \sum_{i=1}^n \beta_i (x_i^{\min} - x_i), \end{aligned}$$

where $\alpha = [\alpha_i]_{\text{vec}}$ and $\beta = [\beta_i]_{\text{vec}}$ with $\alpha_i \in \mathbb{R}$ and $\beta_i \in \mathbb{R}$ being the Lagrange multipliers corresponding to the local allocation feasibility constraint. Let α_i^* and β_i^* be the optimal values. Since the problem in (5.1) is convex and the constraints are linear, the optimal values of the Lagrange multipliers are bounded [15]. If $\kappa > \max\{\alpha_i^*, \beta_i^*\}$ for $i = 1, \dots, n$ and $t \geq 0$, the following condition hold [136]

$$f(x^*) - n\kappa\epsilon \leq f_p(x_p^*, \kappa, \epsilon) \leq f(x^*),$$

where x^* is the optimal solution of the problem in (5.1).

5.3.2.1 Algorithm with a Virtual Leader

Based on the penalized problem in (5.11), the distributed algorithm is designed as follows.

$$\begin{aligned} \delta_0 \dot{\lambda}_0 &= z_q, \\ \delta_1 \dot{x}_i &= -\nabla f_{pi}(x_i) + \lambda_i, \\ \delta_2 \dot{z}_i &= -z_i - \sum_{j=1}^n a_{ij}(z_i - z_j) - \sum_{j=1}^n a_{ij}(\theta_i - \theta_j) + d_i(t) - x_i, \\ \delta_2 \dot{\theta}_i &= \sum_{j=1}^n a_{ij}(z_i - z_j), \\ \delta_2 \dot{\lambda}_i &= -\sum_{j=1}^n a_{ij}(\lambda_i - \lambda_j) - a_{i0}(\lambda_i - \lambda_0), \end{aligned}$$

where δ_0 , δ_1 , and δ_2 are positive constants to be determined, z_q for $q \in \{1, \dots, n\}$ is the q th component of z . Write the algorithm in a compact form gives

$$\delta_0 \dot{\lambda}_0 = z_q, \quad (5.13a)$$

$$\delta_1 \dot{x} = -\nabla f_p(x) + \lambda, \quad (5.13b)$$

$$\delta_2 \dot{z} = -(I + L)z - L\theta + d(t) - x, \quad (5.13c)$$

$$\delta_2 \dot{\theta} = Lz, \quad (5.13d)$$

$$\delta_2 \dot{\lambda} = -L\lambda - A_0(\lambda - \mathbf{1}_n \lambda_0), \quad (5.13e)$$

where $\nabla f_p(x) = [\nabla f_{pi}(x_i)]_{\text{vec}}$ and $A_0 = \text{diag}\{a_{i0}\}$. In this algorithm, (5.13a) and (5.13b) utilize the SPD to find the saddle point of $f_p(x)$, (5.13c) and (5.13d) are designed based on the dynamic average consensus algorithm in Lemma 2.1 to estimate the average value of the mismatch between total resource and total allocation, and (5.13e) utilizes the leader-following consensus algorithm in Lemma 2.2 to estimate the value of λ_0 .

Theorem 5.2. *Under Assumptions 5.1-5.4, if the agents adopt (5.13) to update the decision variables, then, there exist positive constants δ_0^* , $\delta_1^*(\delta_0^*)$, and $\delta_2^*(\delta_0^*, \delta_1^*)$ such that for every $0 < \delta_0 < \delta_0^*$, $0 < \delta_1 < \delta_1^*$, and $0 < \delta_2 < \delta_2^*$, $\|x - x_p^*(t)\|$ is uniformly ultimately bounded as $t \rightarrow \infty$. Furthermore, if $\kappa > \max\{\alpha_i^*, \beta_i^*\}$ for $i \in \{1, \dots, n\}$ and $t \geq 0$,*

$$f(x^*) - n\kappa\epsilon \leq f_p(x_p^*, \kappa, \epsilon) \leq f(x^*).$$

Proof. Similar to the variable transformation in the proof of Theorem 3.1, define $[\theta_a^T \ \theta_b]^T \triangleq U^T \theta$ where $\theta_a \in \mathbb{R}^{n-1}$ and $\theta_b \in \mathbb{R}$. From (5.13c) and (5.13d), it can be derived that

$$\begin{aligned} \delta_2 \begin{bmatrix} \dot{z} \\ \dot{\theta}_a \end{bmatrix} &= \begin{bmatrix} -(I + L) & -LU_1 \\ U_1^T L & \mathbf{0}_{(n-1) \times (n-1)} \end{bmatrix} \begin{bmatrix} z \\ \theta_a \end{bmatrix} + \begin{bmatrix} d(t) - x \\ \mathbf{0}_{n-1} \end{bmatrix}, \\ \dot{\theta}_b &= 0. \end{aligned} \quad (5.14a)$$

Hence, $\theta_b(t) = \theta_2(0)$, which is stable. Let $\bar{z}, \bar{\theta}_a, \bar{\lambda}$ be the solution of the following equations

$$\begin{aligned} -(L + A_0)\bar{\lambda} + A_0\mathbf{1}_n\lambda_0 &= \mathbf{0}_n, \\ \begin{bmatrix} -(I + L) & -LU_1 \\ U_1^T L & \mathbf{0}_{(n-1) \times (n-1)} \end{bmatrix} \begin{bmatrix} \bar{z} \\ \bar{\theta}_a \end{bmatrix} + \begin{bmatrix} d(t) - x \\ \mathbf{0}_{n-1} \end{bmatrix} &= \mathbf{0}_{2n-1}. \end{aligned} \quad (5.15)$$

Solving these equations we can get $\bar{\lambda} = \mathbf{1}_n\lambda_0$, $\bar{z} = \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T(d(t) - x)$, and $\bar{\theta}_a$ is a linear combination of the elements in x and $d(t)$. Define $\psi \triangleq [\lambda^T, z^T, \theta_a^T]^T$, $\bar{\psi} \triangleq [\bar{\lambda}^T, \bar{z}^T, \bar{\theta}_a^T]^T$, and $\tilde{\psi} \triangleq \psi - \bar{\psi}$. Then we have

$$\delta_2 \dot{\tilde{\psi}} = H\tilde{\psi} - \delta_2 \dot{\bar{\psi}},$$

where

$$H = \begin{bmatrix} -(L + A_0) & \mathbf{0}_{n \times n} & \mathbf{0}_{n \times (n-1)} \\ \mathbf{0}_{n \times n} & -(I + L) & -LU_1 \\ \mathbf{0}_{(n-1) \times n} & U_1^T L & \mathbf{0}_{(n-1) \times (n-1)} \end{bmatrix}.$$

Since $\begin{bmatrix} -(I + L) & -LU_1 \\ U_1^T L & \mathbf{0}_{(n-1) \times (n-1)} \end{bmatrix}$ is Hurwitz due to Lemma 2.3 and $-(L + A_0)$ is Hurwitz, H is Hurwitz. Thus, there exists a symmetric positive definite matrix P such that

$$H^T P + PH = -Q$$

for any given symmetric positive definite matrix Q .

Define $g_i = (\nabla f_{pi})^{-1}$, $\bar{x}_i = g_i(\lambda_0)$, $\bar{x} = [\bar{x}_i]_{\text{vec}}$, $\tilde{x} = x - \bar{x}$, and $\hat{x}$ is between x and $\bar{x}$. Then, according to the mean value theorem, we have

$$\delta_1 \dot{\tilde{x}} = \delta_1 \dot{x} - \delta_1 \dot{\bar{x}}$$

$$\begin{aligned}
&= -\nabla f_p(x) + \lambda - \delta_1 \dot{\tilde{x}} \\
&= -(\nabla f_p(\bar{x}) + \nabla^2 f_p(\hat{x})\tilde{x}) + \tilde{\lambda} + \mathbf{1}_n \lambda_0 - \delta_1 \dot{\tilde{x}} \\
&= -\nabla^2 f_p(\hat{x})\tilde{x} + \tilde{\lambda} - \delta_1 \dot{\tilde{x}}.
\end{aligned}$$

Since $f_{pi}(x_i)$ is strngly convex, $\nabla^2 f_p(\hat{x})$ is positive definite $\forall \hat{x}$ and $\nabla^2 f_p(\hat{x}) \geq m_0 I_n$, where $m_0 = \min_{i \in \{1, \dots, n\}} m_i$.

Define $\tilde{\lambda}_0 = \lambda_0 - \lambda_0^*$, where λ_0^* is the optimal Lagrange multiplier corresponding to the equality constraint in (5.11), and $g(\lambda_0^*) = [g_i(\lambda_0^*)]_{\text{vec}}$. Similar to Lemma 5.1, it can be easily verified that λ_0^* exists, satisfies $\mathbf{1}_n^T(d - g(\lambda_0^*)) = 0$, and $\|\dot{\lambda}_0^*\|$ is bounded. Then, we have

$$\begin{aligned}
\delta_0 \dot{\tilde{\lambda}}_0 &= \tilde{z}_q + \bar{z}_q - \delta_0 \dot{\lambda}_0^* \\
&= \tilde{z}_q + \frac{1}{n} \mathbf{1}_n^T(d(t) - x) - \delta_0 \dot{\lambda}_0^* \\
&= \tilde{z}_q + \frac{1}{n} \mathbf{1}_n^T(d(t) - \tilde{x} - \bar{x}) - \delta_0 \dot{\lambda}_0^* \\
&= \tilde{z}_q - \frac{1}{n} \mathbf{1}_n^T \tilde{x} + \frac{1}{n} \mathbf{1}_n^T(d(t) - g(\lambda_0)) - \delta_0 \dot{\lambda}_0^* \\
&= \tilde{z}_q - \frac{1}{n} \mathbf{1}_n^T \tilde{x} - \frac{1}{n} \mathbf{1}_n^T \nabla g(\hat{\lambda}_0) \tilde{\lambda}_0 - \delta_0 \dot{\lambda}_0^*,
\end{aligned}$$

where $\hat{\lambda}_0$ is a value between λ_0^* and λ_0 . Since $f_{pi}(x_i)$ is strongly convex, its gradient $\nabla f_{pi}(x_i)$ is strictly increasing. Thus, the inverse function $g_i(\lambda_0)$ of $\nabla f_{pi}(x_i)$ is strictly increasing. As a result, $\nabla g_i(\hat{\lambda}_0) > 0$ and $\mathbf{1}_n^T \nabla g(\hat{\lambda}_0)$ is positive. In addition, $\nabla g_i(\hat{\lambda}_0)$ is upper bounded since $\nabla^2 f_{pi}(x_i) \geq m_i \forall x_i$.

Define $V = \tilde{\psi}^T P \tilde{\psi} + \frac{1}{2} \tilde{x}^T \tilde{x} + \frac{1}{2} \tilde{\lambda}_0^2$. It can be derived that

$$\begin{aligned}
\dot{V} &= 2\tilde{\psi}^T P \dot{\tilde{\psi}} + \tilde{x}^T \dot{\tilde{x}} + \tilde{\lambda}_0 \dot{\tilde{\lambda}}_0 \\
&= -\frac{1}{\delta_2} \tilde{\psi}^T Q \tilde{\psi} - \frac{1}{\delta_1} \tilde{x}^T \nabla^2 f_p(\hat{x}) \tilde{x} - \frac{1}{\delta_0 n} \mathbf{1}_n^T \nabla g(\hat{\lambda}_0) \tilde{\lambda}_0^2 \\
&\quad - 2\tilde{\psi}^T P \dot{\tilde{\psi}} + \frac{1}{\delta_1} \tilde{x}^T \tilde{\lambda} - \tilde{x}^T \dot{\tilde{x}} + \frac{1}{\delta_0} \tilde{\lambda}_0 \tilde{z}_q - \frac{1}{\delta_0 n} \tilde{\lambda}_0 \mathbf{1}_n^T \tilde{x} - \tilde{\lambda}_0 \dot{\lambda}_0^*.
\end{aligned}$$

Since $\tilde{\psi}$ is a linear combination of λ_0 , $d(t)$, and x , there exist positive constants d_1 ,

d_2 , and d_3 such that

$$-2\tilde{\psi}^T P \dot{\tilde{\psi}} \leq \frac{d_1}{\delta_0} (\|\tilde{\psi}\|^2 + \|\tilde{\psi}\| \|\tilde{x}\| + \|\tilde{\psi}\| \|\tilde{\lambda}_0\|) + \frac{d_2}{\delta_1} (\|\tilde{\psi}\| \|\tilde{x}\| + \|\tilde{\psi}\|^2) + d_3 \|\tilde{\psi}\|.$$

Since $\bar{x} = g(\lambda_0)$, there exists a positive constant d_4 such that

$$-\tilde{x}^T \dot{\tilde{x}} \leq \frac{d_4}{\delta_0} (\|\tilde{x}\| \|\tilde{\psi}\| + \|\tilde{x}\|^2 + \|\tilde{x}\| \|\tilde{\lambda}_0\|).$$

Thus,

$$\begin{aligned} \dot{V} &\leq -\frac{\lambda_{\min}(Q)}{\delta_2} \|\tilde{\psi}\|^2 - \frac{\lambda_{\min}(\nabla^2 f_p(\hat{x}))}{\delta_1} \|\tilde{x}\|^2 - \frac{1}{\delta_0 n} \mathbf{1}_n^T \nabla g(\hat{\lambda}_0) \tilde{\lambda}_0^2 + \left(\frac{d_1}{\delta_0} + \frac{d_2}{\delta_1}\right) \|\tilde{\psi}\|^2 \\ &\quad + \frac{d_4}{\delta_0} \|\tilde{x}\|^2 + \frac{1+d_1}{\delta_0} \|\tilde{\psi}\| \|\tilde{\lambda}_0\| + \left(\frac{1+d_2}{\delta_1} + \frac{d_1+d_4}{\delta_0}\right) \|\tilde{\psi}\| \|\tilde{x}\| \\ &\quad + d_3 \|\tilde{\psi}\| + \left(\frac{d_4}{\delta_0} + \frac{\sqrt{n}}{\delta_0 n}\right) \|\tilde{x}\| \|\tilde{\lambda}_0\| + \|\dot{\lambda}_0^*\| \|\tilde{\lambda}_0\| \\ &\leq -\left(\frac{\lambda_{\min}(Q)}{\delta_2} - \frac{d_1}{\delta_0} - \frac{d_2}{\delta_1} - \frac{1}{4\epsilon_1}\right) \|\tilde{\psi}\|^2 - \left(\frac{m_0}{\delta_1} - \frac{d_4}{\delta_0}\right) \|\tilde{x}\|^2 - \left(\frac{\mathbf{1}_n^T \nabla g(\hat{\lambda}_0)}{\delta_0 n} - \frac{1}{4\epsilon_2}\right) \|\tilde{\lambda}_0\|^2 \\ &\quad + \left(\frac{d_5}{\delta_1} + \frac{d_6}{\delta_0}\right) \|\tilde{\psi}\| \|\tilde{x}\| + \frac{d_7}{\delta_0} \|\tilde{\psi}\| \|\tilde{\lambda}_0\| + \frac{d_8}{\delta_0} \|\tilde{x}\| \|\tilde{\lambda}_0\| + \epsilon_1 d_3^2 + \epsilon_2 d_9^2 \\ &= -[\|\tilde{\psi}\| \|\tilde{x}\|] H_1 \begin{bmatrix} \|\tilde{\psi}\| \\ \|\tilde{x}\| \end{bmatrix} - [\|\tilde{\psi}\| \|\tilde{\lambda}_0\|] H_2 \begin{bmatrix} \|\tilde{\psi}\| \\ \|\tilde{\lambda}_0\| \end{bmatrix} - [\|\tilde{x}\| \|\tilde{\lambda}_0\|] H_3 \begin{bmatrix} \|\tilde{x}\| \\ \|\tilde{\lambda}_0\| \end{bmatrix} \\ &\quad + \epsilon_1 d_3^2 + \epsilon_2 d_9^2, \end{aligned}$$

where $d_5 = 1 + d_2$, $d_6 = d_1 + d_4$, $d_7 = 1 + d_1$, $d_8 = d_4 + \frac{\sqrt{n}}{n}$, $d_9 = \sup \|\dot{\lambda}_0^*\|$, and

$$\begin{aligned} H_1 &= \frac{1}{2} \begin{bmatrix} \frac{\lambda_{\min}(Q)}{\delta_2} - \frac{d_1}{\delta_0} - \frac{d_2}{\delta_1} - \frac{1}{4\epsilon_1} & -\left(\frac{d_5}{\delta_1} + \frac{d_6}{\delta_0}\right) \\ -\left(\frac{d_5}{\delta_1} + \frac{d_6}{\delta_0}\right) & \frac{m_0}{\delta_1} - \frac{d_4}{\delta_0} \end{bmatrix}, \\ H_2 &= \frac{1}{2} \begin{bmatrix} \frac{\lambda_{\min}(Q)}{\delta_2} - \frac{d_1}{\delta_0} - \frac{d_2}{\delta_1} - \frac{1}{4\epsilon_1} & -\frac{d_7}{\delta_0} \\ -\frac{d_7}{\delta_0} & \frac{\mathbf{1}_n^T \nabla g(\hat{\lambda}_0)}{\delta_0 n} - \frac{1}{4\epsilon_2} \end{bmatrix}, \\ H_3 &= \frac{1}{2} \begin{bmatrix} \frac{m_0}{\delta_1} - \frac{d_4}{\delta_0} & -\frac{d_8}{\delta_0} \\ -\frac{d_8}{\delta_0} & \frac{\mathbf{1}_n^T \nabla g(\hat{\lambda}_0)}{\delta_0 n} - \frac{1}{4\epsilon_2} \end{bmatrix}. \end{aligned}$$

By setting δ_0 , δ_1 , and δ_2 satisfy the following conditions

$$\begin{aligned}\delta_0 &< \frac{4\epsilon_2 \mathbf{1}_n^T \nabla g(\hat{\lambda}_0)}{n} \triangleq \delta_0^*, \\ \delta_1 &< \frac{m_0 \delta_0^* (\frac{1}{n} \mathbf{1}_n^T \nabla g(\hat{\lambda}_0) - \frac{\delta_0^*}{4\epsilon_2})}{d_8^2 + (\frac{1}{n} \mathbf{1}_n^T \nabla g(\hat{\lambda}_0) - \frac{\delta_0^*}{4\epsilon_2}) d_4} \triangleq \delta_1^*, \\ \delta_2 &< \frac{\lambda_{\min}(Q)}{\zeta + \frac{d_1}{\delta_0^*} + \frac{d_2}{\delta_1^*} + \frac{1}{4\epsilon_1}} = \delta_2^*,\end{aligned}$$

where $\zeta = \max\{\frac{d_7^2}{\frac{\delta_0^*}{n} \mathbf{1}_n^T \nabla g(\hat{\lambda}_0) - \frac{\delta_0^*}{4\epsilon_2}}, \frac{(d_5/\delta_1^* + d_6/\delta_0^*)^2}{\frac{m_0}{\delta_1^*} - \frac{d_4}{\delta_0^*}}\}$, H_1 , H_2 , and H_3 are positive definite.

Let $\underline{\lambda} = \min\{\lambda_{\min}(H_1), \lambda_{\min}(H_2), \lambda_{\min}(H_3)\}$ and $v = [\tilde{\psi}^T, \tilde{x}^T, \tilde{\lambda}_0]^T$, we have

$$\begin{aligned}\dot{V} &\leq -2\underline{\lambda} \|v\|^2 + \epsilon_1 d_3^2 + \epsilon_2 d_9^2 \\ &\leq -\underline{\lambda} \|v\|^2, \quad \forall \|v\| \geq \sqrt{\frac{\epsilon_1 d_3^2 + \epsilon_2 d_9^2}{\underline{\lambda}}}.\end{aligned}$$

Thus, $\|v\|$ is uniformly ultimately bounded. Since $\bar{x}$ can be bounded by a linear function of λ_0 and $\bar{\psi}$ is a linear function of x and λ_0 , $\|x - x_p^*\|$ is uniformly ultimately bounded.

By selecting κ large enough such that $\kappa > \max\{\alpha_i^*, \beta_i^*\}$ for $i = 1, \dots, n$ and $t \geq 0$, we have [136]

$$f(x^*) - n\kappa\epsilon \leq f_p(x_p^*, \kappa, \epsilon) \leq f(x^*).$$

□

Remark 5.2. *The optimal solution of the penalized problem in (5.11) does not necessarily coincide with the optimal solution of the original problem in (5.1). However, the approximation error can be tuned small enough by tuning the penalty parameter κ and smoothing parameter ϵ .*

5.3.2.2 Algorithm without Leader

In algorithm (5.13), agent q which is selected to calculate the update of λ_0 serves as a virtual leader. In this section, the virtual leader is not required, each agent calculates its λ_i based on its estimation of the mismatch instead of tracking λ_0 of a virtual leader.

Before giving the algorithm, we give the following lemma on the eigenvalues of a matrix. λ^H represents the conjugate transpose of λ .

Lemma 5.3. *For a real matrix $A \in \mathbb{R}^{n \times n}$, if*

$$\begin{aligned} A\nu_1 &= \lambda_1^H \nu_1, \\ A^T \nu_2 &= \lambda_2 \nu_2, \\ &\dots \\ A^T \nu_n &= \lambda_n \nu_n, \end{aligned}$$

and ν_i for $i \in \{1, \dots, n\}$ are orthogonal, then the eigenvalues of A are $\lambda_1, \dots, \lambda_n$.

Proof. Without loss of generality, assume that v_i s are orthonormal.

$$\nu_1^H A^T \nu_1 = (A\nu_1)^H \nu_1 = \lambda_1 \nu_1^H \nu_1 = \lambda_1.$$

There exist $s_i \in \mathbb{R}, i \in \{1, \dots, n\}$, such that

$$A^T \nu_1 = s_1 \nu_1 + s_2 \nu_2 + \dots + s_n \nu_n.$$

Multiplying ν_1^H on the left gives

$$\nu_1^H A^T \nu_1 = s_1 \nu_1^H \nu_1 + s_2 \nu_1^H \nu_2 + \dots + s_n \nu_1^H \nu_n = s_1.$$

Thus, $s_1 = \lambda_1$. Let $V = [v_1 \ v_2 \ \dots \ v_n]$. There exist a matrix B , such that

$$A^T V = V B. \quad (5.16)$$

From the above analysis, the first column of B is $[\lambda_1 \ s_2 \ \dots \ s_n]^T$. Since $A^T \nu_k = \lambda_k \nu_k$ for $k \in \{2, \dots, n\}$, the remaining columns of B have $\lambda_i, i \in \{2, \dots, n\}$ on the main diagonal and zero otherwise. Thus, B is a lower triangular matrix with $\lambda_i, i \in \{1, \dots, n\}$ being its eigenvalues. Since A and B are similar due to (5.16), the eigenvalues of A are $\lambda_1, \dots, \lambda_n$. $\square$

Based on the gradient and the singular perturbation theory, the distributed control algorithm without a virtual leader is designed as follows:

$$\begin{aligned} \delta_0 \dot{\lambda}_i &= z_i - \sum_{j=1}^n a_{ij}(\lambda_i - \lambda_j), \\ \delta_1 \dot{x}_i &= -\nabla f_{pi}(x_i) + \lambda_i, \\ \delta_2 \dot{z}_i &= -z_i - \sum_{j=1}^n a_{ij}(z_i - z_j) - \sum_{j=1}^n a_{ij}(\theta_i - \theta_j) + d_i(t) - x_i, \\ \delta_2 \dot{\theta}_i &= \sum_{j=1}^n a_{ij}(z_i - z_j), \end{aligned}$$

where δ_0, δ_1 , and δ_2 are positive constants to be determined. Write the algorithm in a compact form gives

$$\delta_0 \dot{\lambda} = z - L\lambda, \quad (5.17a)$$

$$\delta_1 \dot{x} = -\nabla f_p(x) + \lambda, \quad (5.17b)$$

$$\delta_2 \dot{z} = -(I + L)z - L\theta + d(t) - x, \quad (5.17c)$$

$$\delta_2 \dot{\theta} = Lz. \quad (5.17d)$$

Theorem 5.3. *Under Assumptions 5.1-5.4, if the agents adopt (5.17) to update the*

decision variables, then, there exist positive constants δ_0^* , $\delta_1^*(\delta_0^*)$, and $\delta_2^*(\delta_0^*, \delta_1^*)$ such that for every $0 < \delta_0 < \delta_0^*$, $0 < \delta_1 < \delta_1^*$, and $0 < \delta_2 < \delta_2^*$, $\|x - x_p^*(t)\|$ is uniformly ultimately bounded as $t \rightarrow \infty$. Furthermore, if $\kappa > \max\{\alpha_i^*, \beta_i^*\}$ for $i \in \{1, \dots, n\}$ and $t \geq 0$,

$$f(x^*) - n\kappa\epsilon \leq f_p(x_p^*, \kappa, \epsilon) \leq f(x^*).$$

Proof. Define $\tilde{\lambda} = \lambda - \lambda^*$ and $\hat{\lambda}$ is between λ and λ^* . According to condition (5.12), $L\lambda^* = \mathbf{0}_n$ and $\mathbf{1}_n^T(d(t) - g(\lambda^*)) = 0$ with $g(\lambda^*) = [g_i(\lambda_i^*)]_{\text{vec}}$. Thus, we have

$$\begin{aligned} \delta_0 \dot{\tilde{\lambda}} &= \delta_0 \dot{\lambda} - \delta_0 \dot{\lambda}^* \\ &= \tilde{z} + \bar{z} - L\tilde{\lambda} - L\lambda^* - \delta_0 \dot{\lambda}^* \\ &= \tilde{z} + \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^T (d(t) - x) - L\tilde{\lambda} - \delta_0 \dot{\lambda}^* \\ &= \tilde{z} + \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^T (d(t) - \tilde{x} - \bar{x}) - L\tilde{\lambda} - \delta_0 \dot{\lambda}^* \\ &= \tilde{z} - \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^T \tilde{x} + \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^T (d(t) - g(\lambda)) - L\tilde{\lambda} - \delta_0 \dot{\lambda}^* \\ &= \tilde{z} - \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^T \tilde{x} + \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^T (d(t) - g(\lambda^*) - \nabla g(\hat{\lambda})\tilde{\lambda}) - L\tilde{\lambda} - \delta_0 \dot{\lambda}^* \\ &= \tilde{z} - \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^T \tilde{x} - H_2 \tilde{\lambda} - \delta_0 \dot{\lambda}^*, \end{aligned}$$

where $H_2 = \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^T \nabla g(\hat{\lambda}) + L$. Since $\nabla g(\hat{\lambda})$ is a diagonal matrix, $\frac{1}{n} \mathbf{1}_n \mathbf{1}_n^T \nabla g(\hat{\lambda})$ is a matrix with column space of $\text{span}\{\mathbf{1}_n\}$. Let $\lambda_{L1} = 0 < \lambda_{L2} \leq \dots \leq \lambda_{Ln}$ be the eigenvalues of L , and $\nu_1, \dots, \nu_n$ be the corresponding eigenvectors. It is known that $\nu_1 \in \text{span}\{\mathbf{1}_n\}$, thus, we have

$$(\frac{1}{n} \mathbf{1}_n \mathbf{1}_n^T \nabla g(\hat{\lambda}) + L)\nu_1 = \frac{1}{n} \mathbf{1}_n^T \nabla g(\hat{\lambda}) \mathbf{1}_n \nu_1.$$

As a result, $\frac{1}{n} \mathbf{1}_n^T \nabla g(\hat{\lambda}) \mathbf{1}_n$ is an eigenvalue of $\frac{1}{n} \mathbf{1}_n \mathbf{1}_n^T \nabla g(\hat{\lambda}) + L$ and $\frac{1}{n} \mathbf{1}_n^T \nabla g(\hat{\lambda}) \mathbf{1}_n$ is positive. Since L is symmetric and λ_{L1} is the only zero eigenvalue, $\nu_2, \dots, \nu_n$ are orthogonal to ν_1 . Thus, $\nu_2, \dots, \nu_n$ belong to the null space of $(\frac{1}{n} \mathbf{1}_n \mathbf{1}_n^T \nabla g(\hat{\lambda}))^T$. Then,

for $i = 2, \dots, n$,

$$\left(\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\nabla g(\hat{\lambda}) + L\right)^T\nu_i = L\nu_i = \lambda_{Li}\nu_i.$$

According to Lemma 5.3, the eigenvalues of the matrix $(\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\nabla g(\hat{\lambda}) + L)$ are $\frac{1}{n}\mathbf{1}_n^T\nabla g(\hat{\lambda})\mathbf{1}_n, \lambda_{L2}, \dots, \lambda_{Ln}$, which have positive real parts. Thus, there exists a symmetric positive definite matrix P_2 such that

$$\left(\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\nabla g(\hat{\lambda}) + L\right)^TP_2 + P_2\left(\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\nabla g(\hat{\lambda}) + L\right) = Q_2$$

for any given symmetric positive definite matrix Q_2 . Similarly, we have

$$\begin{aligned}\delta_1\dot{\tilde{x}} &= -\nabla f_p(x) + \lambda - \delta_1\dot{\tilde{x}} \\ &= -(\nabla f_p(\bar{x}) + \nabla^2 f_p(\hat{x})\tilde{x}) + \lambda - \delta_1\dot{\tilde{x}} \\ &= -\nabla^2 f_p(\hat{x})\tilde{x} - \delta_1\dot{\tilde{x}}.\end{aligned}$$

Define $V = \tilde{\psi}^TP_1\tilde{\psi} + \frac{1}{2}\tilde{x}^T\tilde{x} + \tilde{\lambda}^TP_2\tilde{\lambda}$, following a similar proof of that in Theorem 5.2, we can get

$$\begin{aligned}\dot{V} &\leq -\frac{\lambda_{\min}(Q_1)}{\delta_2}\|\tilde{\psi}\|^2 - \frac{m_0}{\delta_1}\|\tilde{x}\|^2 - \frac{\lambda_{\min}(Q_2)}{\delta_0}\|\tilde{\lambda}\|^2 + \frac{d_3}{\delta_0}\|\tilde{x}\|^2 + \left(\frac{d_1}{\delta_1} + \frac{d_3}{\delta_0}\right)\|\tilde{\psi}\|\|\tilde{x}\| \\ &\quad + \frac{d_4}{\delta_0}\|\tilde{\psi}\|\|\tilde{\lambda}\| + \frac{d_5}{\delta_0}\|\tilde{x}\|\|\tilde{\lambda}\| + d_2\|\tilde{\psi}\| + d_6\|\tilde{\lambda}\| \\ &\leq -\left(\frac{\lambda_{\min}(Q_1)}{\delta_2} - \frac{1}{4\epsilon_1}\right)\|\tilde{\psi}\|^2 - \left(\frac{m_0}{\delta_1} - \frac{d_3}{\delta_0}\right)\|\tilde{x}\|^2 - \left(\frac{\lambda_{\min}(Q_2)}{\delta_0} - \frac{1}{4\epsilon_2}\right)\|\tilde{\lambda}\|^2 \\ &\quad + \left(\frac{d_1}{\delta_1} + \frac{d_3}{\delta_0}\right)\|\tilde{\psi}\|\|\tilde{x}\| + \frac{d_4}{\delta_0}\|\tilde{\psi}\|\|\tilde{\lambda}\| + \frac{d_5}{\delta_0}\|\tilde{x}\|\|\tilde{\lambda}\| + \epsilon_1d_2^2 + \epsilon_2d_6^2 \\ &= -[\|\tilde{\psi}\| \ \|\tilde{x}\|]H_1 \begin{bmatrix} \|\tilde{\psi}\| \\ \|\tilde{x}\| \end{bmatrix} - [\|\tilde{\psi}\| \ \|\tilde{\lambda}\|]H_2 \begin{bmatrix} \|\tilde{\psi}\| \\ \|\tilde{\lambda}\| \end{bmatrix} - [\|\tilde{x}\| \ \|\tilde{\lambda}\|]H_3 \begin{bmatrix} \|\tilde{x}\| \\ \|\tilde{\lambda}\| \end{bmatrix} \\ &\quad + \epsilon_1d_2^2 + \epsilon_2d_6^2,\end{aligned}$$

where

$$\begin{aligned} H_1 &= \frac{1}{2} \begin{bmatrix} \frac{\lambda_{\min}(Q_1)}{\delta_2} - \frac{1}{4\epsilon_1} & -\left(\frac{d_1}{\delta_1} + \frac{d_3}{\delta_0}\right) \\ -\left(\frac{d_1}{\delta_1} + \frac{d_3}{\delta_0}\right) & \frac{m_0}{\delta_1} - \frac{d_3}{\delta_0} \end{bmatrix}, \\ H_2 &= \frac{1}{2} \begin{bmatrix} \frac{\lambda_{\min}(Q_1)}{\delta_2} - \frac{1}{4\epsilon_1} & -\frac{d_4}{\delta_0} \\ -\frac{d_4}{\delta_0} & \frac{\lambda_{\min}(Q_2)}{\delta_0} - \frac{1}{4\epsilon_2} \end{bmatrix}, \\ H_3 &= \frac{1}{2} \begin{bmatrix} \frac{m_0}{\delta_1} - \frac{d_3}{\delta_0} & -\frac{d_5}{\delta_0} \\ -\frac{d_5}{\delta_0} & \frac{\lambda_{\min}(Q_2)}{\delta_0} - \frac{1}{4\epsilon_2} \end{bmatrix}. \end{aligned}$$

By setting δ_0 , δ_1 , and δ_2 to satisfy the following conditions

$$\begin{aligned} \delta_0 &< \frac{\lambda_{\min}(Q_2)}{4\epsilon_2} \triangleq \delta_0^*, \\ \delta_1 &< \frac{m_0\delta_0^*(4\epsilon_2\lambda_{\min}(Q_2) - \delta_0^*)}{4\epsilon_2d_5^2 + d_3(4\epsilon_2\lambda_{\min}(Q_2) - \delta_0^*)} \triangleq \delta_1^*, \\ \delta_2 &< \frac{4\epsilon_1\lambda_{\min}(Q_1)}{\zeta + 1} = \delta_2^*, \end{aligned}$$

where $\zeta = \max\left\{\frac{16\epsilon_1\epsilon_2d_4^2}{\delta_0^*(4\epsilon_2\lambda_{\min}(Q_2) - \delta_0^*)}, \frac{4\epsilon_1\delta_0^*\delta_1^*(d_1/\delta_1^* + d_3/\delta_0^*)^2}{\delta_0^*m_0 - \delta_1^*d_3}\right\}$, H_1 , H_2 , and H_3 are positive definite.

Following the proof of Theorem 5.2, the conclusion is obtained. $\square$

5.3.3 Numerical Examples

A numerical example is given to show the efficacy of the proposed algorithm. Consider a system of five agents share two common resources. The communication graph among the five agents is depicted in Figure. 5.1.

The weights associated with the edges of the communication graph are $a_{ij} = 1$ if $(v_i, v_j) \in \mathcal{E}$, and $a_{ij} = 0$ otherwise. The agents intend to minimize the team objective function $\sum_{i=1}^n f_i(x_i)$. The local objective function $f_i(x_i) : \mathbb{R}^2 \rightarrow \mathbb{R}$ for

Figure 5.1: Communication graph for the five agents.

agent i , $i = 1, \dots, 5$, is

$$f_i(x_i) = x_i^T A_i x_i + B_i^T x_i,$$

where

$$A_1 = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}, A_2 = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 2 \end{bmatrix}, A_3 = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}, A_4 = \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}, A_5 = \begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix},$$

$$B_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, B_2 = \begin{bmatrix} 2 \\ 3 \end{bmatrix}, B_3 = \begin{bmatrix} 3 \\ 4 \end{bmatrix}, B_4 = \begin{bmatrix} 4 \\ 5 \end{bmatrix}, B_5 = \begin{bmatrix} 5 \\ 6 \end{bmatrix}.$$

The time-varying local resource $d_i(t) \in \mathbb{R}^2$ for $i = 1, \dots, 5$ is set as

$$d_i(t) = [i + ie^{-0.1it} \sin(it) \quad i + ie^{-0.1it} \cos(it)]^T.$$

Without local allocation feasibility constraint, algorithm (5.8)-(5.9) is used. The control parameters are set as $k_a = 100$ and $k_b = 10$. Figure. 5.2 plots the evolution of the agents' decision variables. The upper figure shows the tracking behavior of the first common resource and the lower figure shows the tracking behavior of the second common resource. It can be seen that the decision variables track their optimal solution trajectories, and the total allocation tracks the total resource. Figure. 5.3 plots the evolution of λ_i s, which converge to two values corresponding to the two common resource constraints.

Consider the local allocation feasibility constraint, algorithm (5.13) is used to find

Figure 5.2: Evolution of x without local allocation feasibility constraint produced by (5.8)-(5.9). The upper figure shows that the decision variables track the optimal allocation of resource 1, and the lower figure shows that the decision variables track the optimal allocation of resource 2.

Figure 5.3: Evolution of λ without local allocation feasibility constraint produced by (5.8)-(5.9).

Figure 5.4: Evolution of x with local allocation feasibility constraint produced by (5.13). The upper figure shows that the decision variables track the optimal allocation of resource 1, and the lower figure shows that the decision variables track the optimal allocation of resource 2.

the optimal solution. In this simulation, a lower bound of 0 and an upper bound of 6, which satisfy Assumption 5.3, are imposed on each agent. The control parameters are set as $\kappa = 530$, $\epsilon = 0.01$, $k = 1000$, $\delta_1 = \delta_2 = 0.01$, and agent 1 is selected to calculate λ_0 . Figure 5.4 shows that the decision variables track their optimal solution trajectories which are within their allocation feasibility constraints and the total allocation tracks the total resource. Figure 5.5 shows that λ_i s converge to two values.

Figure 5.5: Evolution of λ with local allocation feasibility constraint produced by (5.13).

5.4 Distributed Economic Dispatch with Time-varying Loads

In this section, the algorithms developed in Section 5.3 are applied to the ED problem with continuously time-varying loads.

5.4.1 System Model and Problem Formulation

In smart grids, distributed generating units work collaboratively to serve the loads over a common distribution network. Based on the load demand, ED produces optimal generation references for the distributed generating units such that the total generation cost is minimized under the power balance and generation capacity constraints. In practice, the load demand usually varies from time to time according to industrial production, residential activity, temperature, etc. As the tertiary control, ED produces operating point for the secondary and primary control. By taking the time-varying nature of the load into consideration, more economical operation of the smart grid is expected.

Suppose there are n control areas in a smart grid, area i has local generating units to supply power $P_{Gi} \in \mathbb{R}$ and local loads to consume power $P_{Di}(t) \in \mathbb{R}$. The ED problem is formulated as

$$\begin{aligned} & \text{minimize } f(P_G) = \sum_{i=1}^n f_i(P_{Gi}), \\ & \text{subject to } \sum_{i=1}^n P_{Gi} = \sum_{i=1}^n P_{Di}(t), \\ & P_{Gi}^{\min} \leq P_{Gi} \leq P_{Gi}^{\max}, \quad i = 1, 2, \dots, n, \end{aligned} \tag{5.18}$$

where $P_G = [P_{Gi}]_{\text{vec}}$, $f_i(P_{Gi})$ is the generation cost function of area i , $P_{Gi}^{\min}$ and $P_{Gi}^{\max}$ are the lower and upper bounds of P_{Gi} with $P_{Gi}^{\min} < P_{Gi}^{\max}$. The generation cost of

area i is modeled as a quadratic function with respect to its local generation P_{Gi} as

$$f_i(x_i) = a_i P_{Gi}^2 + b_i P_{Gi} + c_i,$$

where $a_i > 0$, $b_i \geq 0$ and $c_i \geq 0$ are the cost parameters. The local loads $P_{Di}(t)$ are supposed to be continuously time-varying and satisfy $\sum_{i=1}^n P_{Gi}^{\min} < \sum_{i=1}^n P_{Di}(t) < \sum_{i=1}^n P_{Gi}^{\max}$ for all t .

5.4.2 Distributed Algorithm

When the capacity limits of the generators are not considered, algorithm (5.8)-(5.9) in Section 5.3 is applied to derive the distributed algorithm for the ED problem as follows

$$\dot{P}_{Gi} = -k_a(2a_i P_{Gi} + b_i - \lambda_i) - k_b(2a_i \text{sgn}_{1i} + \text{sgn}_{2i}), \quad (5.19a)$$

$$\dot{\lambda}_i = k_a(P_{Di}(t) - P_{Gi} - \sum_{j=1}^n a_{ij}(\mu_i - \mu_j)) + k_b(\text{sgn}_{1i} + \sum_{j=1}^n a_{ij}(\text{sgn}_{3i} - \text{sgn}_{3j})), \quad (5.19b)$$

$$\dot{\mu}_i = k_a \sum_{j=1}^n a_{ij}(\lambda_i - \lambda_j) - k_b \sum_{j=1}^n a_{ij}(\text{sgn}_{2i} - \text{sgn}_{2j}), \quad (5.19c)$$

with

$$\begin{aligned} \text{sgn}_{1i} &= \text{sgn}(2a_i P_{Gi} + b_i - \lambda_i), \\ \text{sgn}_{2i} &= \text{sgn}(P_{Gi} + \sum_{j=1}^n a_{ij}(\mu_i - \mu_j) - P_{Di}(t)), \\ \text{sgn}_{3i} &= \text{sgn}(-\sum_{j=1}^n a_{ij}(\lambda_i - \lambda_j)). \end{aligned}$$

When the capacity limits are taken into consideration, algorithm (5.17) where there

is no virtual leader is applied to derive the distributed algorithm as

$$\delta_0 \dot{\lambda}_i = z_i - \sum_{j=1}^n a_{ij}(\lambda_i - \lambda_j), \quad (5.20a)$$

$$\delta_1 \dot{P}_{Gi} = -\nabla f_{pi}(P_{Gi}) + \lambda_i, \quad (5.20b)$$

$$\delta_2 \dot{z}_i = -z_i - \sum_{j=1}^n a_{ij}(z_i - z_j) - \sum_{j=1}^n a_{ij}(\theta_i - \theta_j) + P_{Di}(t) - P_{Gi}, \quad (5.20c)$$

$$\delta_2 \dot{\theta}_i = \sum_{j=1}^n a_{ij}(z_i - z_j), \quad (5.20d)$$

where $f_{pi}(P_{Gi}) = f_i(P_{Gi}) + p_i(P_{Gi} - P_{Gi}^{\max}) + p_i(P_{Gi}^{\min} - P_{Gi})$ and the gradient of $f_{pi}(P_{Gi})$ is

$$\nabla f_{pi}(P_{Gi}) = \begin{cases} 2a_i P_{Gi} + b_i - \kappa & : P_{Gi} \leq P_{Gi}^{\min} - \epsilon \\ 2a_i P_{Gi} + b_i - \frac{\kappa}{\epsilon}(P_{Gi}^{\min} - P_{Gi}) & : P_{Gi}^{\min} - \epsilon < P_{Gi} \leq P_{Gi}^{\min} \\ 2a_i P_{Gi} + b_i & : P_{Gi}^{\min} < P_{Gi} \leq P_{Gi}^{\max} \\ 2a_i P_{Gi} + b_i + \frac{\kappa}{\epsilon}(P_{Gi} - P_{Gi}^{\max}) & : P_{Gi}^{\max} < P_{Gi} \leq P_{Gi}^{\max} + \epsilon \\ 2a_i P_{Gi} + b_i + \kappa & : P_{Gi}^{\max} + \epsilon < P_{Gi}. \end{cases}$$

5.4.3 Simulation on the 24-bus Test System

Consider the 24-bus system [137] shown in Figure. 5.6. There are 12 generating units and 17 loads. The generating unit G10 located in bus 22 is a hydro generator which supplies a constant power of 300 MW, whereas the rest of them are all thermal generators. The hydro generator can be treated as a negative load. The system is divided into 9 control areas as shown by the red dashed line in Figure. 5.6. Control area A_i has local generation P_{Gi} and local load P_{Di} , which are the aggregated generations and loads in this area. Table 5.1 indicates the location of generating units, loads, and the capacity limits of each control area based on the data given in [137]. The cost parameters are random selected from intervals $a_i \in [0.0015, 0.0050](\$/\text{MW}/\text{MWh})$, $b_i \in [7, 8](\$/\text{MWh})$, and $c_i \in [200, 450](\$/\text{h})$.

Figure 5.6: 24-bus system and control areas. G_i represents generating unit, L_i represents load, and A_i represents control area.

Table 5.1: Composition and capacity limit of each control area

Control Area	Generating Unit	Load	$P_{Gi}^{\min}$ (MW)	$P_{Gi}^{\max}$ (MW)
A1	G1	L1 L5	30.4	152
A2	G2	L2 L4 L9	30.4	152
A3	G3	L3 L6 L7 L8 L10	75	300
A4	G5 G6	L13	66.25	215
A5	G7	L12 L14 L16	54.25	155
A6	G11 G12	L17	248.5	660
A7	G4	L11	206.85	591
A8	G8	L15	100	400
A9	G9 G10		100	400

The total load profile for a 24-hour horizon is listed in Table 5.2 and each area's contribution to the system load is shown in Table 5.3 based on the data in [137].

Using the 24-hour hourly load profile, we make a spline interpolation to produce a continuously time-varying total load $\sum_{i=1}^n P_{Di}(t)$ as shown in Figure 5.7. The

Table 5.2: 24-hour hourly total system load profile from [137]

Hour	Total Load (MW)	Hour	Total Load (MW)
0	1775.835	12	2517.975
1	1669.815	13	2517.975
2	1590.300	14	2464.965
3	1563.795	15	2464.965
4	1563.795	16	2623.995
5	1590.300	17	2650.500
6	1961.370	18	2650.500
7	2279.430	19	2544.480
8	2517.975	20	2411.955
9	2544.480	21	2199.915
10	2544.480	22	1934.865
11	2517.975	23	1669.815

Table 5.3: Control area distribution of the total system load from [137]

Area #	1	2	3	4	5	6	7	8	9
%	6.3	12.1	28.3	11.1	16.7	4.5	9.3	11.7	0

continuously time-varying local loads $P_{Di}(t)$ used in the algorithm are produced according to their contributions to the system load. In this simulation, we assume that there is no generating units start up or shut down.

The communication topology among the control areas is shown in Figure. 5.8, which is undirected and connected.

First, the capacity limits of the control areas are not considered, the algorithm (5.19) is applied. The parameters are set as $k_a = 100$ and $k_b = 10$. Figure. 5.9 plots the evolution of the generation decision variables and the evolution of λ_i s. It can be seen that the generation decision variables converge to and track the optimal generations, the total generation tracks the total load, and λ_i s converge to a common value.

Then, the capacity limits of the control areas are considered and the algorithm (5.20) is applied in this situation. The penalty parameter κ is set as $\kappa = 4.728$ and the smoothing parameter ϵ is set as $\epsilon = 0.01$. The control parameters are set as $\delta_0 = 0.01$, $\delta_1 = 0.001$, and $\delta_2 = 0.00005$. The upper figure in Figure. 5.10 shows that

Figure 5.7: Continuously time-varying system load by spline interpolation.

Figure 5.8: Communication topology among the nine control areas.

the total generation tracks the total load, and each generation is within its capacity limit. The lower figure in Figure. 5.10 plots the evolution of λ_i s, which converge to a common value.

In addition, the cost function of A1 is assumed to be non-quadratic but strongly

Figure 5.9: Power generation and evolution of λ without capacity limits produced by (5.19).

convex as

$$f_1(x_1) = a_1 P_{G1}^2 + b_1 P_{G1} + c_1 + 100e^{\frac{P_{G1}}{100}},$$

The simulation results in Fig. 5.11 show that the algorithm is able to solve the ED problem when the cost function is strongly convex instead of quadratic. Since the cost function of A1 has relatively large incremental cost, the power generation of A1 is low as shown in the upper figure in Figure. 5.11.

Figure 5.10: Power generation and evolution of λ with capacity limits produced by (5.20).

5.5 Conclusion

In this chapter, the RA problem with TVR is investigated. Distributed algorithms based on consensus algorithms and gradient descent are developed. For the case without local allocation feasibility constraint under quadratic objective functions, the proposed algorithm enables the agents to track the time-varying optimal solution trajectory. For the case with local allocation feasibility constraint under strongly convex objective functions, the tracking error is proven to be uniformly ultimately bounded. The algorithms are applied to the ED problem with continuously time-varying loads. The numerical simulation results and the simulation results on the 24-bus system show the efficacy of the proposed algorithms.

Figure 5.11: Power generation and evolution of λ with capacity limits under non-quadratic cost function produced by (5.20).

Chapter 6

Distributed Practical Non-Convex Economic Dispatch

6.1 Introduction

Generally, the cost functions for the thermal generating units in the ED problem have been approximately represented by quadratic functions as shown in Chapter 4. This approximation ignores some practical characteristics of thermal units and thus will introduce inaccuracy into the resulting dispatch. In practice, the thermal generating units have non-smooth and non-convex cost curves due to the valve-point effect [4]. Moreover, many generating units are supplied with multi-fuel options, such as coal, oil, and natural gas [4], which leads to the problem of determining the most economic fuel to use. These characteristics make the practical ED problem a non-convex and non-smooth optimization problem which is hard to be solved by mathematical optimization methods.

To handle the complication in the practical ED problem, meta-heuristics methods have been proposed, such as the particle swarm optimization technique [58, 138, 139, 140, 141], genetic algorithm [43, 142, 143], differential evolution [49, 144], artificial

bee colony [63, 145], and hybrid methods that combine two or more approaches [64, 65], etc. The meta-heuristics methods do not require strict assumptions on the cost function and have the ability to cope with non-convex and non-smooth cost functions. However, these methods usually have high computational time and are implemented in a centralized scheme, which suffer from the drawbacks of centralized algorithms such as low flexibility and high operation cost.

Inspired by the distributed structure of the algorithms proposed before, this chapter proposes a distributed algorithm based on the consensus algorithms and randomized gradient to solve the practical ED problem. Consensus algorithms are utilized to estimate the global information in a distributed manner. With the estimated information, the updating law using randomized gradient is performed within each generating unit. By utilizing the randomized gradient, the non-smooth characteristic of the cost functions can be handled, and the random parameter imposed in the randomized gradient can help to escape from local minima. Although the global optimal solution is not guaranteed to be found, a promising distributed algorithm is proposed to solve the practical ED problem. Simulation on a three generating units system is first conducted to illustrate the proposed algorithm. Then, application on a ten generating units system is provided to show the applicability.

6.2 Problem Formulation

Similar to the problem formulation in Chapter 4, consider the following practical ED problem which minimizes the total generation cost under the balance constraint and capacity limit constraint,

$$\min f(P_G) = \sum_{i=1}^n f_i(P_{Gi}) \quad (6.1a)$$

$$\text{s.t. } \sum_{i=1}^n P_{Gi} = \sum_{i=1}^n P_{Di} \quad (6.1b)$$

$$P_{Gi}^{\min} \leq P_{Gi} \leq P_{Gi}^{\max}, \forall i = 1, 2, \dots, n, \quad (6.1c)$$

where n is the number of buses, $P_{Gi} \in \mathbb{R}$ is the active power generated by the generating unit in bus i , $P_G = [P_{Gi}]_{\text{vec}} \in \mathbb{R}^n$, $P_{Di} \in \mathbb{R}$ is the active power needed by the load in bus i , $P_{Gi}^{\min} \in \mathbb{R}$ and $P_{Gi}^{\max} \in \mathbb{R}$ are the lower and upper bounds of P_{Gi} , and $f_i(P_{Gi}) \in \mathbb{R}$ is the cost function with valve-point effect and multi-fuel options of generating unit i . In steam turbine generators, steam admission valves are opened in sequence to increase the generation output. When a valve is first opened, a rippling effect is produced on the unit cost curve, which is modeled by a sinusoidal function. The cost function addressing the valve-point effect of generating unit i is formulated as [54]

$$f_i(P_{Gi}) = a_i^f P_{Gi}^2 + b_i^f P_{Gi} + c_i^f + |e_i^f \sin(h_i^f (P_{Gi}^{\min} - P_{Gi}))|,$$

where $a_i^f, b_i^f, c_i^f, e_i^f$, and h_i^f are the cost coefficients of generating unit i . A generating unit supplied with multiple fuel options needs to operate on the lower contour of the intersecting curves. Suppose there are k types of fuels for generating unit i , then, the cost function considers both valve-point effect and multi-fuel options is formulated as [43]

$$f_i(P_{Gi}) = \begin{cases} a_{i1}^f P_{Gi}^2 + b_{i1}^f P_{Gi} + c_{i1}^f + |e_{i1}^f \sin(h_{i1}^f (P_{Gi}^{\min} - P_{Gi}))|, & \text{for fuel 1, } P_{Gi}^{\min} \leq P_{Gi} < P_{Gi1}, \\ a_{i2}^f P_{Gi}^2 + b_{i2}^f P_{Gi} + c_{i2}^f + |e_{i2}^f \sin(h_{i2}^f (P_{Gi}^{\min} - P_{Gi}))|, & \text{for fuel 2, } P_{Gi1} \leq P_{Gi} < P_{Gi2}, \\ \vdots & \\ a_{ik}^f P_{Gi}^2 + b_{ik}^f P_{Gi} + c_{ik}^f + |e_{ik}^f \sin(h_{ik}^f (P_{Gi}^{\min} - P_{Gi}))|, & \text{for fuel } k, P_{Gi(k-1)} \leq P_{Gi} \leq P_{Gi}^{\max}, \end{cases} \quad (6.2)$$

where $a_{ik}^f > 0$, b_{ik}^f , c_{ik}^f , e_{ik}^f , and h_{ik}^f are the fuel cost coefficients of unit i using fuel type k . The cost function considers valve-point effects and multiple fuels is a piecewise superposition of sinusoidal functions and quadratic functions. It has non-differentiable points and is non-convex. In this chapter, we assume that the cost function $f_i(P_{Gi})$ is continuous.

6.3 Distributed Algorithm for the Non-convex Economic Dispatch Problem

6.3.1 Gaussian Smoothing and Randomized Gradient

Since the objective function is non-smooth, we using a Gaussian approximation to obtain its smoothed form. The Gaussian approximation of $f_i(P_{Gi})$ is [96]

$$f_{i,\mu_i}(P_{Gi}) = \frac{1}{\kappa} \int_{\mathbb{R}} f_i(P_{Gi} + \mu_i \xi_i) e^{-\frac{1}{2} \xi_i^2} d\xi_i,$$

where $\kappa = \int_{\mathbb{R}} e^{-\frac{1}{2} \xi_i^2} d\xi_i = \sqrt{2\pi}$ and $\mu_i \geq 0$ is the smoothing parameter. For $\mu_i > 0$, f_{i,μ_i} is differentiable. To calculate the gradient of f_{i,μ_i} , let $y = P_{Gi} + \mu_i \xi_i$, then $f_{i,\mu_i}(P_{Gi}) = \frac{1}{\kappa \mu_i} \int_{\mathbb{R}} f_i(y) e^{-\frac{1}{2\mu_i^2} (y - P_{Gi})^2} dy$. Based on the differential rule, the gradient of $f_{i,\mu_i}(P_{Gi})$ can be calculated as

$$\begin{aligned} \nabla f_{i,\mu_i} &= \frac{1}{\kappa \mu_i^3} \int_{\mathbb{R}} f_i(y) e^{-\frac{1}{2\mu_i^2} (y - P_{Gi})^2} (y - P_{Gi}) dy \\ &= \frac{1}{\kappa \mu_i} \int_{\mathbb{R}} f_i(P_{Gi} + \mu_i \xi_i) e^{-\frac{1}{2} \xi_i^2} \xi_i d\xi_i \\ &= \frac{1}{\kappa} \int_{\mathbb{R}} \frac{f_i(P_{Gi} + \mu_i \xi_i) - f_i(P_{Gi})}{\mu_i} e^{-\frac{1}{2} \xi_i^2} \xi_i d\xi_i + \frac{1}{\kappa} \int_{\mathbb{R}} \frac{f_i(P_{Gi})}{\mu_i} e^{-\frac{1}{2} \xi_i^2} \xi_i d\xi_i \\ &= \frac{1}{\kappa} \int_{\mathbb{R}} \frac{f_i(P_{Gi} + \mu_i \xi_i) - f_i(P_{Gi})}{\mu_i} e^{-\frac{1}{2} \xi_i^2} \xi_i d\xi_i. \end{aligned} \quad (6.3)$$

The last equation comes from the fact that $\frac{1}{\kappa} \int_{\mathbb{R}} \frac{f_i(P_{Gi})}{\mu_i} e^{-\frac{1}{2} \xi_i^2} \xi_i d\xi_i = 0$.

Denote $\Omega_i = \{P_{Gi} | P_{Gi}^{\min} \leq P_{Gi} \leq P_{Gi}^{\max}\}$. Since $f_i(P_{Gi})$ is continuous and Ω_i is compact, $f_i(P_{Gi})$ is Lipschitz over Ω_i . Denote the Lipschitz constant as $L_0(f_i)$. Then, we have

$$|f_i(x_i) - f_i(y_i)| \leq L_0(f_i)|x_i - y_i|, \quad \forall x_i, y_i \in \Omega_i.$$

Since $f_i(P_{Gi})$ is Lipschitz continuous over Ω_i , we can get the following properties of f_{i,μ_i} [96]. $\forall x_i, y_i \in \Omega_i$

1. $|f_{i,\mu_i}(x_i) - f_{i,\mu_i}(y_i)| \leq L_0(f_i)|x_i - y_i|;$
2. $|\nabla f_{i,\mu_i}(x_i) - \nabla f_{i,\mu_i}(y_i)| \leq \frac{1}{\mu_i} L_0(f_i)|x_i - y_i|.$

The randomized gradient is defined as [96]

$$g_{\mu_i}(P_{Gi}, \xi_{i,j}) = \frac{f_i(P_{Gi} + \mu_i \xi_{i,j}) - f_i(P_{Gi})}{\mu_i} \xi_{i,j}, \quad (6.4)$$

where $\xi_{i,j}$ is a standard normally distributed random variable. According to (6.3), we have

$$\mathbb{E}[g_{\mu_i}(P_{Gi}, \xi_{i,j})] = \nabla f_{i,\mu_i}(P_{Gi}). \quad (6.5)$$

Thus, $g_{\mu_i}(P_{Gi}, \xi_{i,j})$ can be viewed as an unbiased gradient of the smoothed f_{i,μ_i} . In addition, we assume that $g_{\mu_i}(P_{Gi}, \xi_{i,j})$ satisfies the following condition,

$$\mathbb{E}[(g_{\mu_i}(P_{Gi}, \xi_{i,j}) - \nabla f_{i,\mu_i}(P_{Gi}))^2] \leq \sigma^2,$$

where $\sigma > 0$ is a constant.

6.3.2 Distributed Algorithm

Based on the randomized gradient (6.4), we design the following distributed algorithm to update the variables.

$$z_i^{t+1} = \rho z_i^t - \epsilon \sum_{j=1}^n a_{ij}((z_i^t - z_j^t) + (\theta_i^t - \theta_j^t)) + \epsilon(1 - \rho)(P_{Di} - P_{Gi}^t), \quad (6.6a)$$

$$\theta_i^{t+1} = \theta_i^t + \epsilon \sum_{j=1}^n a_{ij}(z_i^t - z_j^t), \quad (6.6b)$$

$$\lambda_i^{t+1} = \lambda_i^t - \epsilon \sum_{j=1}^n a_{ij}(\lambda_i^t - \lambda_j^t) + \bar{\rho}_t z_i^t, \quad (6.6c)$$

$$P_{Gi}^{t+1} = P_{\Omega_i}[P_{Gi}^t - \gamma_t g_{i,t} + \lambda_i^t], \quad (6.6d)$$

where $t \in \mathbb{Z}^+$ is the iteration index, $\rho, \epsilon, \bar{\rho}_t$ are positive parameters, P_{Ω_i} is the projection operation onto the set Ω_i , γ_t is the step size, and $g_{i,t} = \frac{1}{m_t} \sum_{j=1}^{m_t} g_{\mu_i}(P_{Gi}^t, \xi_{i,j}^t)$ with m_t being a finite positive integer and $\xi_{i,j}^t$ for $j = 1, \dots, m_t$ being independent and identically distributed (i.i.d.) random variables.

Remark 6.1. Since f_i is non-convex, we can not provide theoretical performance guarantee for the optimality of the solution. The rationale behind the updates in (6.6) is to use the dynamic average consensus (6.6a) and (6.6b) to estimate the mismatch between total load and total generation. Then, with this mismatch, λ_i , which can be seen as the Lagrange multiplier corresponding to the equality constraint, is updated using (6.6c). Finally, P_{Gi} is updated using (6.6d) based on the randomized gradient, λ_i , and the projection operation.

If treating P_{Gi}^t in (6.6a) as a constant, then z_i^t will converge to $\frac{1}{n} \sum_{i=1}^n (P_{Di} - P_{Gi}^t)$ [146], which is the averaged mismatch between total load and total generation. In (6.6c), with properly set parameter $\bar{\rho}_t$, λ_i will converge to a common value. Let $\tilde{h}_i^t = h_{\Omega_i}(P_{Gi}^t, \lambda_i^t, \gamma_t, g_{i,t}) = \frac{1}{\gamma_t} (P_{Gi}^t - P_{\Omega_i}[P_{Gi}^t - \gamma_t g_{i,t} + \lambda_i^t])$ and $\delta_i^t = g_{i,t} - \nabla f_{i,\mu_i}(P_{Gi}^t)$. The upper bound of the expectation of $\tilde{h}_i^t$ can be seen as follows. Recall that $|\nabla f_{i,\mu_i}(x_i) -$

$|\nabla f_{i,\mu_i}(y_i)| \leq \frac{1}{\mu_i} L_0(f_i) |x_i - y_i|$, we have

$$\begin{aligned}
f_{i,\mu_i}(P_{Gi}^{t+1}) &\leq f_{i,\mu_i}(P_{Gi}^t) + \nabla f_{i,\mu_i}(P_{Gi}^t)(P_{Gi}^{t+1} - P_{Gi}^t) + \frac{1}{2\mu_i} L_0(f_i)(P_{Gi}^{t+1} - P_{Gi}^t)^2 \\
&= f_{i,\mu_i}(P_{Gi}^t) + \nabla f_{i,\mu_i}(P_{Gi}^t)(P_{\Omega_i}[P_{Gi}^t - \gamma_t g_{i,t} + \lambda_i^t] - P_{Gi}^t) \\
&\quad + \frac{1}{2\mu_i} L_0(f_i)(P_{\Omega_i}[P_{Gi}^t - \gamma_t g_{i,t} + \lambda_i^t] - P_{Gi}^t)^2 \\
&= f_{i,\mu_i}(P_{Gi}^t) - \gamma_t \nabla f_{i,\mu_i}(P_{Gi}^t) \tilde{h}_i^t + \frac{\gamma_t^2}{2\mu_i} L_0(f_i) \tilde{h}_i^{t2} \\
&= f_{i,\mu_i}(P_{Gi}^t) - \gamma_t g_{i,t} \tilde{h}_i^t + \gamma_t \delta_i^t \tilde{h}_i^t + \frac{\gamma_t^2}{2\mu_i} L_0(f_i) \tilde{h}_i^{t2}.
\end{aligned}$$

Inspired by [147], we have the following property of the projection operation,

$$g_{i,t} \tilde{h}_i^t \geq \tilde{h}_i^{t2} + \frac{1}{\gamma_t} \lambda_i^t \tilde{h}_i^t.$$

Thus,

$$\begin{aligned}
f_{i,\mu_i}(P_{Gi}^{t+1}) &\leq f_{i,\mu_i}(P_{Gi}^t) - \gamma_t \tilde{h}_i^{t2} - \lambda_i^t \tilde{h}_i^t + \gamma_t \delta_i^t \tilde{h}_i^t + \frac{\gamma_t^2}{2\mu_i} L_0(f_i) \tilde{h}_i^{t2} \\
&\leq f_{i,\mu_i}(P_{Gi}^t) - (\gamma_t - \frac{\gamma_t^2}{2\mu_i} L_0(f_i) - \frac{1}{\alpha}) \tilde{h}_i^{t2} + \frac{\alpha}{4} \lambda_i^{t2} + \gamma_t \delta_i^t \tilde{h}_i^t \\
&= f_{i,\mu_i}(P_{Gi}^t) - (\gamma_t - \frac{\gamma_t^2}{2\mu_i} L_0(f_i) - \frac{1}{\alpha}) \tilde{h}_i^{t2} + \frac{\alpha}{4} \lambda_i^{t2} + \gamma_t \delta_i^t h_i^t + \gamma_t \delta_i^t (\tilde{h}_i^t - h_i^t),
\end{aligned}$$

where α is a positive constant satisfies $\alpha > \frac{2}{\gamma_t}$ and $h_i^t = \frac{1}{\gamma_t}(P_{Gi}^t - P_{\Omega_i}[P_{Gi}^t - \gamma_t \nabla f_{i,\mu_i}(P_{Gi}^t) + \lambda_i^t])$. From the non-expansive property of projection, we have

$$\begin{aligned}
|\tilde{h}_i^t - h_i^t| &= \frac{1}{\gamma_t} |P_{\Omega_i}[P_{Gi}^t - \gamma_t g_{i,t} + \lambda_i^t] - P_{\Omega_i}[P_{Gi}^t - \gamma_t \nabla f_{i,\mu_i}(P_{Gi}^t) + \lambda_i^t]| \\
&\leq |g_{i,t} - \nabla f_{i,\mu_i}(P_{Gi}^t)| \\
&= |\delta_i^t|.
\end{aligned}$$

Thus,

$$f_{i,\mu_i}(P_{Gi}^{t+1}) \leq f_{i,\mu_i}(P_{Gi}^t) - (\gamma_t - \frac{\gamma_t^2}{2\mu_i} L_0(f_i) - \frac{1}{\alpha}) \tilde{h}_i^{t2} + \frac{\alpha}{4} \lambda_i^{t2} + \gamma_t \delta_i^t h_i^t + \gamma_t \delta_i^{t2}.$$

By setting $\gamma_t \leq \frac{\mu_i}{L_0(f_i)}$, we have $\gamma_t - \frac{\gamma_t^2}{2\mu_i} L_0(f_i) - \frac{1}{\alpha} > 0$. Summing up the above inequalities for $t = 1, \dots, N$ gives

$$\begin{aligned} \sum_{t=1}^N \left(\gamma_t - \frac{\gamma_t^2}{2\mu_i} L_0(f_i) - \frac{1}{\alpha} \right) \tilde{h}_i^{t2} &\leq f_{i,\mu_i}(P_{Gi}^1) - f_{i,\mu_i}(P_{Gi}^{N+1}) + \sum_{t=1}^N \left(\frac{\alpha}{4} \lambda_i^{t2} + \gamma_t \delta_i^t h_i^t + \gamma_t \delta_i^{t2} \right) \\ &\leq f_{i,\mu_i}(P_{Gi}^1) - f_{i,\mu_i}^* + \sum_{t=1}^N \left(\frac{\alpha}{4} \lambda_i^{t2} + \gamma_t \delta_i^t h_i^t + \gamma_t \delta_i^{t2} \right). \end{aligned} \quad (6.7)$$

Since P_{Gi}^t is a function of history $(\xi^1, \dots, \xi^{t-1})$, h_i^t is a function of history $(\xi^1, \dots, \xi^{t-1})$. Thus, h_i^t is independent of ξ^t . Since $\mathbb{E}[g_{\mu_i}(P_{Gi}^t, \xi_{i,j}^t)] = \nabla f_{i,\mu_i}(P_{Gi}^t)$ in Eq. (6.5), $\mathbb{E}[g_{i,t}] = \mathbb{E}[\frac{1}{m_t} \sum_{j=1}^{m_t} g_{\mu_i}(P_{Gi}^t, \xi_{i,j}^t)] = \frac{1}{m_t} \sum_{j=1}^{m_t} \mathbb{E}[g_{\mu_i}(P_{Gi}^t, \xi_{i,j}^t)] = \nabla f_{i,\mu_i}(P_{Gi}^t)$. Thus, $\mathbb{E}[\delta_i^t h_i^t] = \mathbb{E}[\mathbb{E}[\delta_i^t h_i^t | (\xi^1, \dots, \xi^{t-1})]] = \mathbb{E}[h_i^t \mathbb{E}[\delta_i^t | (\xi^1, \dots, \xi^t)]] = 0$ by noticing that $\delta_i^t = g_{i,t} - \nabla f_{i,\mu_i}(P_{Gi}^t)$. Denote $\delta_{i,j}^t = g_{\mu_i}(P_{Gi}^t, \xi_{i,j}^t) - \nabla f_{i,\mu_i}(P_{Gi}^t)$, $S_{m_t} = \sum_{j=1}^{m_t} \delta_{i,j}^t$, and $S_0 = 0$, we have

$$\mathbb{E}[S_{m_t}^2] = \mathbb{E}[S_{m_{t-1}}^2 + 2S_{m_{t-1}}\delta_{i,m_t}^t + \delta_{i,m_t}^{t2}] = \mathbb{E}[S_{m_{t-1}}^2] + \mathbb{E}[\delta_{i,m_t}^{t2}] = \dots = \sum_{j=1}^{m_t} \mathbb{E}[\delta_{i,j}^{t2}].$$

Thus,

$$\mathbb{E}[\delta_{i,j}^{t2}] = \mathbb{E}\left[\left(\frac{1}{m_t} \sum_{j=1}^{m_t} \delta_{i,j}^t\right)^2\right] = \frac{1}{m_t^2} \mathbb{E}[S_{m_t}^2] = \frac{1}{m_t^2} \sum_{j=1}^{m_t} \mathbb{E}[\delta_{i,j}^{t2}] \leq \frac{1}{m_t} \sigma^2.$$

Taking expectation on both sides of (6.7) gives

$$\sum_{t=1}^N \left(\gamma_t - \frac{\gamma_t^2}{2\mu_i} L_0(f_i) - \frac{1}{\alpha} \right) \mathbb{E}[\tilde{h}_i^{t2}] \leq f_{i,\mu_i}(P_{Gi}^1) - f_{i,\mu_i}^* + \sum_{t=1}^N \left(\frac{\alpha}{4} \mathbb{E}[\lambda_i^{t2}] + \gamma_t \sigma^2 / m_t \right).$$

Thus,

$$\min_t \mathbb{E}[\tilde{h}_i^{t2}] \leq \frac{f_{i,\mu_i}(P_{Gi}^1) - f_{i,\mu_i}^* + \sum_{t=1}^N \left(\frac{\alpha}{4} \mathbb{E}[\lambda_i^{t2}] + \gamma_t \sigma^2 / m_t \right)}{\sum_{t=1}^N \left(\gamma_t - \frac{\gamma_t^2}{2\mu_i} L_0(f_i) - \frac{1}{\alpha} \right)}.$$

The expectation of $\tilde{h}_i^{t2}$ is upper bounded by the initial value P_{Gi}^1 , the expectation of λ_i^{t2} , the step size γ_t , and m_t . If z_i^t is at its steady-state $\frac{1}{n} \sum_{i=1}^n (P_{Di} - P_{Gi}^t)$, then $\mathbb{E}[z_i^t]$

is upper bounded due to the projection operation on P_{Gi}^t . Thus, by properly selecting the parameter $\bar{\rho}_t$ in (6.6c), $\mathbb{E}[\lambda_i^{t2}]$ is bounded. As a result, $\min_t \mathbb{E}[\tilde{h}_i^{t2}]$ is bounded.

The stability of the whole system is expected by treating the estimation part (6.6a)-(6.6b) and the updating part (6.6c) as two-time scales subsystems and using the singular perturbation analysis.

Based on the updates in (6.6), Algorithm 2 is proposed to solve the non-convex ED problem. In Algorithm 2, we first generate a set of candidate solutions P_G^s , and then the averaged values of the S runs are given as the final output.

The implementation procedure of Algorithm 2 is shown in the flowchart in Figure. 6.1.

6.4 Simulation

6.4.1 Numerical Example

In this section, we conduct a numerical simulation on three generating units. The cost function parameters are listed in Table 6.1. The cost functions are continuous, non-convex, and non-smooth shown in Fig 6.2. The communication topology among the three generating units is shown in Figure. 6.3.

Three cases where the total load $\sum_{i=1}^3 P_{Di}$ equals 0.9, 3, 5.5 are tested. With a total load of 0.9, all generating units can only select fuel type 1; with a total load of 5.5, all generating units can only select fuel type 2. The control parameters are set as $a_{ij} = 10$ if $(v_i, v_j) \in \mathcal{E}$, $a_{ij} = 0$ otherwise, $\mu_i = 0.0001$, $\epsilon = 0.001$, $\rho = 0.1$, $\bar{\rho}_t = \bar{\rho} = 0.003$, $\gamma_t = 0.1/\sqrt{t+1}$, $m_t = 1$, $N = 10000$, and $S = 10$. The initial values of the variables are randomly selected.

The updates of the average values of P_{Gi} , z_i , and λ_i are shown in Figure. 6.4. As can

Algorithm 2: Distributed algorithm for non-convex ED problem

Input : Coefficients of cost functions $a_{ik}^f, b_{ik}^f, c_{ik}^f, e_{ik}^f, h_{ik}^f$, local load demand P_{Di} , generation limits $P_{Gi}^{\min}$ and $P_{Gi}^{\max}$.

Output : Generation reference P_{Gi}^* .

Initialize: Set smoothing parameter μ_i , positive parameters $\rho, \epsilon, \bar{\rho}_t, \gamma_t, m_t$, iteration limit N , number of runs S , and weights of the communication graph a_{ij} .

```

1 for  $s = 1, \dots, S$  do
2   for  $t = 1, \dots, N$  do
3     Set  $\bar{\rho}_t, \gamma_t, m_t$ .
4     for  $i = 1, \dots, n$  do
5       Set  $P_{Gi}^1 \in \Omega_i, z_i^1, \theta_i^1, \lambda_i^1$ .
6       Generate random variable  $\xi_{i,j}^t$  for  $j = 1, \dots, m_t$  in an i.i.d manner
          according to the standard Gaussian distribution.
7       Compute the randomized gradient
           $g_{\mu_i}(P_{Gi}^t, \xi_{i,j}^t) = \frac{f_i(P_{Gi}^t + \mu_i \xi_{i,j}^t) - f_i(P_{Gi}^t)}{\mu_i} \xi_{i,j}^t$  and
           $g_{i,t} = \frac{1}{m_t} \sum_{j=1}^{m_t} g_{\mu_i}(P_{Gi}^t, \xi_{i,j}^t)$ .
8       Compute
          
$$z_i^{t+1} = \rho z_i^t - \epsilon \sum_{j=1}^n a_{ij} ((z_i^t - z_j^t) + (\theta_i^t - \theta_j^t)) + \epsilon(1 - \rho)(P_{Di} - P_{Gi}^t),$$

          
$$\theta_i^{t+1} = \theta_i^t + \epsilon \sum_{j=1}^n a_{ij} (z_i^t - z_j^t),$$

          
$$\lambda_i^{t+1} = \lambda_i^t - \epsilon \sum_{j=1}^n a_{ij} (\lambda_i^t - \lambda_j^t) + \bar{\rho}_t z_i^t,$$

          
$$P_{Gi}^{t+1} = P_{\Omega_i} [P_{Gi}^t - \gamma_t g_{i,t} + \lambda_i^t].$$

9     end
10  end
11  Output  $P_G^s = [P_{Gi}^{N+1}]_{\text{vec}}$ .
12 end
13  $P_G^* = \frac{1}{S} \sum_{s=1}^S P_G^s$ .
```

Table 6.1: Cost function parameters of the numerical example

Unit	Fuel Type	Parameters					Operation Range
		a_i^f	b_i^f	c_i^f	e_i^f	h_i^f	
1	1	1	0	1	0.02	10	0-1
	2	1	-2	3	0.02	10	1-2
2	1	0.8	0	1	0.02	10	0-1
	2	1	-1	1.8	0.02	10	1-2
3	1	1.2	0	1	0.02	10	0-1
	2	1	-1	2.2	0.02	10	1-2

be seen from Figure. 6.4, the outputs converge to steady values under different total loads. The total generation converges to the total load, z_i which is the estimation of the mismatch between total load and total generation converges to zero, and λ_i converges to common values. The fluctuation comes from the randomized gradient. The output of Algorithm 2 which is the average value of the 10 runs of P_{Gi} , the total generation (TG), and total cost (TC) under different total loads are listed in Table 6.2, where results obtained using a genetic algorithm (GA) are also listed. The results of the proposed algorithm exhibit similar solutions with the genetic algorithm. Furthermore, the proposed algorithm is implemented in a distributed manner and the parameters in the algorithm are robust to different total loads.

Table 6.2: Comparison of the numerical example results of Algorithm 2 and Genetic Algorithm

Total Load	0.9		3		5.5	
Unit	GA	Algorithm 1	GA	Algorithm 1	GA	Algorithm 1
1	0.3098	0.3110	1.5711	1.5834	2.0000	1.9922
2	0.3141	0.3142	0.8324	0.8439	1.7454	1.7790
3	0.2751	0.2757	0.5955	0.5742	1.7537	1.7336
TG	0.8990	0.9009	2.9990	3.0014	5.4990	5.5048
TC	3.2742	3.2754	5.3303	5.3359	9.6798	9.6980

6.4.2 Simulation on a Ten Generating Units System

In this section, a system composed of 10 generating units is employed to test Algorithm 2. The system data is adopted from [43], which is listed in Table 6.3. A total load demand of 2700 MW is to be met. To represent the distributed nature of load demand, we uniformly distribute the total load demand into 10 local loads.

Table 6.3: Parameters of the cost function [43]

Unit	Fuel Type	Cost Function Parameters					Operation Range
		a_i^f	b_i^f	c_i^f	e_i^f	h_i^f	
1	1	.2176e-2	-.3975e0	.2697e2	.2697e-1	-.3975e1	100-196
	2	.1861e-2	-.3059e0	.2113e2	.2113e-1	-.3059e1	196-250
2	1	.4194e-2	-.1269e1	.1184e3	.1184e0	-.1269e2	157-230
	2	.1138e-2	-.3988e-1	.1865e1	.1865e-2	-.3988e0	50-114
	3	.1620e-2	-.1980e0	.1365e2	.1365e-1	-.1980e1	114-157
3	1	.1457e-2	-.3116e0	.3979e2	.3979e-1	-.3116e1	200-332
	2	.1176e-4	.4864e0	-.5914e2	-.5914e-1	.4864e1	388-500
	3	.8035e-3	.3389e-1	-.2875e1	-.2876e-2	.3389e0	332-388
4	1	.1049e-2	-.3114e-1	.1983e1	.1983e-2	-.3114e0	99-138
	2	.2758e-2	-.6348e0	.5285e2	.5285e-1	-.6348e1	138-200
	3	.5935e-2	-.2338e1	.2668e3	.2668e0	-.2338e2	200-265
5	1	.1066e-2	-.8733e-1	.1392e2	.1392e-1	-.8733e0	190-338
	2	.1597e-2	-.5206e0	.9976e2	.9976e-1	-.5206e1	338-407
	3	.1498e-3	.4462e0	-.5399e2	-.5399e-1	.4462e1	407-490
6	1	.2758e-2	-.6348e0	.5285e2	.5285e-1	-.6348e1	138-200
	2	.1049e-2	-.3114e-1	.1983e1	.1983e-2	-.3114e0	85-138
	3	.5935e-2	-.2338e1	.2668e3	.2668e0	-.2338e2	200-265
7	1	.1107e-2	-.1325e0	.1893e2	.1893e-1	-.1325e1	200-331
	2	.1165e-2	-.2267e0	.4377e2	.4377e-1	-.2267e1	331-391
	3	.2454e-3	.3559e0	-.4335e2	-.4335e-1	.3559e1	391-500
8	1	.1049e-2	-.3114e-1	.1983e1	.1983e-2	-.3114e0	99-138
	2	.2758e-2	-.6348e0	.5285e2	.5285e-1	-.6348e1	138-200
	3	.5935e-2	-.2338e-1	.2668e3	.2668e0	-.2338e2	200-265
9	1	.1554e-2	-.5675e0	.8853e2	.8853e-1	-.5675e1	213-370
	2	.7033e-2	-.4514e-1	.1530e2	.1423e-1	-.1817e0	440-
	3	.6121e-3	-.1817e-1	.1423e2	.1423e-1	-.1817e0	130-213 370-440
10	1	.1102e-2	-.9938e-1	.1397e2	.1397e-1	-.9938e0	200-362
	2	.4164e-4	.5084e0	-.6113e2	-.6113e-1	.5084e1	407-490
	3	.1137e-2	-.2024e0	.4671e2	.4671e-1	-.2024e1	362-407

The communication topology among the 10 generating units is shown in Figure 6.5. The weights associated with the edges of the communication graph are defined as $a_{ij} = 1$ if $(v_i, v_j) \in \mathcal{E}$, otherwise, $a_{ij} = 0$. The parameters are set as $\mu_i = 0.1, \epsilon = 0.01, \rho = 0.1, \bar{\rho}_t = 1/\sqrt{t+1}, \gamma_t = 0.1/\sqrt{t+1}, m_t = 1, N = 6000$, and $S = 10$. The updates of the variables are shown in Fig. 6.6. The upper figure in Figure 6.6 shows that the total generation converges to the total load. The lower figure in Figure 6.6 shows that z , which is the estimation of the mismatch between total generation and total load, converges to zero.

The comparison between the simulation result of Algorithm 2 and the result obtained in [43] using an improved genetic algorithm (GA) is given in Table 6.4. Although the result obtained by Algorithm 1 is not the best, it gives an acceptable solution. Furthermore, Algorithm 1 is performed in a distributed manner.

Table 6.4: Comparison of the simulation results on the 10 generating units system

Methods	GA [43]		Algorithm 2	
Unit	Fuel Type	Generation	Fuel Type	Generation
1	2	219.1261	2	249.9394
2	1	211.1645	1	229.9988
3	1	280.6572	1	313.7661
4	3	238.4770	3	250.9132
5	1	276.4179	1	280.5469
6	3	240.4672	3	258.1474
7	1	287.7399	1	274.4386
8	3	240.7614	3	264.9991
9	3	429.3370	1	303.4556
10	1	275.8518	1	273.6572
TG (MW)	2700.00		2699.86	
TC (\$)	624.5178		640.1383	

6.5 Conclusion

The practical ED problem with valve-point effect and multi-fuel options is investigated in this Chapter. The cost function of the practical ED problem is non-

convex and non-smooth. To handle the non-convex and non-smooth properties, the Gaussian smoothing is used to approximate the cost function, and the randomized gradient is utilized to update the decision variables. By using the dynamic average consensus, the mismatch between the total load and the total generation is estimated in a distributed manner. With this estimated mismatch as feedback, the Lagrange multiplier is updated based on a consensus algorithm. Then, the generation is updated based on the randomized gradient, the Lagrange multiplier, and the projection operation. Simulations on a numerical example composed of three generating units and an example adopted from previous work composed of ten generating units are conducted. The simulation results show the applicability of the proposed distributed algorithm.

Figure 6.1: Flowchart of Algorithm 2.

Figure 6.2: Cost function curve of the numerical example.

Figure 6.3: Communication topology of the three generating units.

Figure 6.4: Updates of average values of P_{Gi} , z_i , and λ_i under different total load of the numerical example.

Figure 6.5: Communication topology of the ten generating units.

Figure 6.6: Generation P_G and z produced by Algorithm 2.

Chapter 7

Conclusions and Future Works

7.1 Conclusions

In this thesis, we have investigated the basic RA problem with separable convex objective functions and equality resource constraint. Two cases where the resources are static and continuously time-varying are investigated, respectively. To demonstrate the practical relevance, distributed ED among a set of distributed generating units is studied. Furthermore, the practical ED problem with consideration of valve-point effect and multi-fuel options is solved in a distributed manner to make the dispatch more accurate. The obtained results are summarized as follows.

1. The basic RA problem with constant resource is solved in a distributed manner in Chapter 3. Distributed algorithms based on dynamic average consensus, leader-following consensus, and SPD are developed for the cases without and with local allocation feasibility constraint, respectively. Our algorithms have the advantages that do not have stringent initial requirement, can keep private information from communicating with other users, and do not need to discover the global information such as the total number of users. The stability of the proposed algorithms is analyzed by singular perturbation theory and

Lyapunov stability analysis. For the case without local allocation feasibility constraint, the proposed algorithm can ensure exponential convergence, and for the case with local allocation feasibility constraint, the proposed algorithm ensures practical convergence. The algorithm and result for the case with local allocation feasibility constraint is applicable to a more general RA problem with strictly convex objective function and local convex inequality constraint.

2. As a typical application of the RA, the ED problem in smart grids is investigated in Chapter 4. The basic ED problem is solved in a distributed manner by the algorithms proposed in Chapter 3. For the ED problem with line flow limits, a distributed algorithm is proposed by revising the algorithm proposed before. Numerous case studies are conducted on the IEEE 9-bus and the IEEE 118-bus systems to show the effectiveness of the proposed algorithms. Demonstration platform based on Matlab GUI is developed to demonstrate the algorithms in an interactive manner.
3. Considering the resources often exhibit time-varying nature, RA problem with continuously TVR is investigated in Chapter 5. The existence of the optimal primal and dual solution of the RA problem with TVR is proven, and the condition the optimal solution should satisfy is given. For the case under quadratic objective functions without local allocation feasibility constraint, a distributed algorithm is designed from the primal dual viewpoint, sign function is utilized to tracking the time-varying optimal solution. The tracking error is proven to converge to zero in finite time using the Lyapunov stability analysis. For the case under strongly convex objective function with local allocation feasibility constraint, a distributed algorithm is developed based on the singular perturbation theory, consensus algorithms, and the saddle point dynamics. The tracking error is proven to be uniformly ultimately bounded. The algorithms developed are applied to solve the ED problem with continuously TVR.
4. To achieve higher accuracy of the load dispatch, a more practical ED problem

considering valve-point effect and multi-fuel option is investigated in Chapter 6. The ED problem with valve-point effect and multi-fuel option is a non-convex and non-smooth optimization problem. A distributed algorithm based on consensus algorithms and the randomized gradient is proposed to solve this problem. The consensus algorithms are utilized to estimate the global information in a distributed manner. With the estimated information, the updating law using randomized gradient and projection operation is performed within each generating unit. By utilizing the randomized gradient, the non-smooth characteristic of the cost functions can be handled, and the random parameter imposed in the randomized gradient provides a possibility to escape from local minima.

7.2 Recommendations for Further Works

Besides the work achieved in this thesis, there are several interesting research topics in the area of distributed RA, which may be considered in the future.

1. Solving the RA problem with non-convex and non-smooth objective functions and stochastic resources in a distributed manner. Many resource allocation problems in reality have non-convex and non-smooth objective functions and the resources often have stochastic character. Such as the ED with renewable energy resources, which can be treated as negative loads often vary randomly [148]. In Chapter 6, we investigated the practical ED with non-convex and non-smooth cost functions utilizing a randomized gradient, which is also promising to handle the stochastic character of the resources. In the future, we will utilize the randomized gradient to solve this problem. By singular perturbation analysis, the whole system can be decoupled into different time scales subsystems, theoretical analysis on the convergence of the subsystems as well as the whole system will be conducted. Furthermore, it is important

to generalize the method to a broader scope of RA problems. One promising direction is to explore distributed intelligent searching method, such as distributed genetic algorithm, to solve this kind of problem.

2. Solving the RA problem with system dynamics in a distributed manner. In this thesis, we consider no dynamic constraint on the decision variable x_i , i.e., $\dot{x}_i = u_i$, where u_i is the algorithm designed to make x_i converge to the optimal solution. The decision variable with physical meanings may subject to additional dynamic constraint. If the optimization procedure can be integrated with the control procedure, the problem is expected to be solved more efficiently. For example, if the decision variable x_i represents the position of an agent, the control input imposed on the decision variable should be the force, which leads to the relationship of $\ddot{x}_i = u_i$ [87]. Another practical example is the ED problem. The ED generates the scheduled generation for the generators based on the load demand. In the ED problem considered in this thesis, the dynamics of the generating units are not considered. The dynamics of the generating units are usually constrained by a number of factors [149, 150], thus it takes time to set the output of the generating units to the scheduled reference. With separated economic dispatch and generating units control, the economic efficiency will be decreased, especially under fast variations in power supply and demand. One promising direction to solve this problem is to design the optimization objective functions incorporating the dynamics of the decision variables [149]. Designing algorithms to minimize such optimization objective functions will naturally handle the dynamics of the system. With this idea, the key part will be how to design appropriate optimization objective functions, where the internal model principle can be considered.
3. Solving time-varying RA problem with time-varying non-quadratic objective function and time-varying constraints, and exploring general convex TVO problems with general time-varying constraints and coupled time-varying objectives. In Chapter 5, the RA problem under non-quadratic objective function

is solved to get an approximated solution. How to get an exact solution in a distributed architecture is still a question. Furthermore, the time-varying RA problem is a special kind of TVO problem, which has a relatively simple separable objective function and linear constraints. However, many practical optimization problems have more complicated objective functions and constraints. Since there are limited results on distributed TVO with constraints, this kind of work can make valuable contributions to the body of distributed TVO. One potential method is to decouple the coupled objective functions using consensus algorithms, utilize the estimated Hessian to track the time-varying optimal solution, and use the Lagrange multiplier or exact penalty method to handle the constraints.

Author's Publications

1. L. Bai, M. Ye, C. Sun, and G. Hu, "Distributed Economic Dispatch Control via Saddle Point Dynamics and Consensus Algorithms", *IEEE Transactions on Control Systems Technology*, vol. 27, no. 2, pp. 898-905, 2019.
2. L. Bai, M. Ye, C. Sun, and G. Hu, "Distributed Control for Optimal Economic Dispatch via Saddle Point Dynamics and Consensus Algorithms", *IEEE Conference on Decision and Control (CDC)*, Las Vegas, NV, USA, pp. 6934-6393, Dec 2016.
3. L. Bai, C. Sun, Z. Feng, and G. Hu, "Distributed Continuous-time Resource Allocation with Time-varying Resources under Quadratic Cost Functions", *IEEE Conference on Decision and Control (CDC)*, Miami Beach, FL, USA, Dec 2018.
4. L. Bai, C. Sun, Z. Feng, and G. Hu, "Distributed Continuous-time Resource Allocation with Time-varying Resources and Its Application to Economic Dispatch", *IEEE Transactions on Control Systems Technology*, conditionally accepted, 2019.

Bibliography

- [1] P. Kubat and H. S. Koch, “Managing test-procedures to achieve reliable software,” *IEEE Transactions on Reliability*, vol. 32, no. 3, pp. 299–303, 1983.
- [2] A. J. Conejo, M. Carrión, and J. M. Morales, *Decision Making under Uncertainty in Electricity Markets*. New York, USA: Springer, 2010.
- [3] C.-L. Chiang, “Improved genetic algorithm for power economic dispatch of units with valve-point effects and multiple fuels,” *IEEE Transactions on Power Systems*, vol. 20, no. 4, pp. 1690–1699, 2005.
- [4] H. Markowitz, “Portfolio selection,” *The Journal of Finance*, vol. 7, no. 1, pp. 77–91, 1952.
- [5] C. C. Aggarwal, J. L. Wolf, and P. S. Yu, “On optimal batching policies for video-on-demand storage servers,” in *IEEE International Conference on Multimedia Computing and Systems*. IEEE, 1996, pp. 253–258.
- [6] H. Ohtera and S. Yamada, “Optimal allocation and control problems for software-testing resources,” *IEEE Transactions on Reliability*, vol. 39, no. 2, pp. 171–176, 1990.
- [7] A. J. Wood and B. F. Wollenberg, *Power Generation, Operation, and Control*. New York, USA: John Wiley & Sons, 2012.

- [8] M. Patriksson, “A survey on the continuous nonlinear resource allocation problem,” *European Journal of Operational Research*, vol. 185, no. 1, pp. 1–46, 2008.
- [9] N. Katoh, A. Shioura, and T. Ibaraki, “Resource allocation problems,” in *Handbook of Combinatorial Optimization*. New York, USA: Springer, 2013, pp. 2897–2988.
- [10] M. B. Cain, R. P. O’neill, and A. Castillo, “History of optimal power flow and formulations,” *Federal Energy Regulatory Commission*, vol. 1, pp. 1–36, 2012.
- [11] A. M. Annaswamy and M. Amin, *IEEE Vision for Smart Grid Controls: 2030 and Beyond*. New York, USA: IEEE, 2013.
- [12] X. Fang, S. Misra, G. Xue, and D. Yang, “Smart grid—the new and improved power grid: A survey,” *IEEE Communications Surveys & Tutorials*, vol. 14, no. 4, pp. 944–980, 2011.
- [13] J. R. Marden and T. Roughgarden, “Generalized efficiency bounds in distributed resource allocation,” *IEEE Transactions on Automatic Control*, vol. 59, no. 3, pp. 571–584, 2014.
- [14] G. Binetti, A. Davoudi, F. L. Lewis, D. Naso, and B. Turchiano, “Distributed consensus-based economic dispatch with transmission losses,” *IEEE Transactions on Power Systems*, vol. 29, no. 4, pp. 1711–1720, 2014.
- [15] H. Liang, B. J. Choi, A. Abdrabou, W. Zhuang, and X. Shen, “Decentralized economic dispatch in microgrids via heterogeneous wireless networks,” *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 6, pp. 1061–1074, 2012.
- [16] C. G. Cassandras, L. Dai, and C. G. Panayiotou, “Ordinal optimization for a class of deterministic and stochastic discrete resource allocation problems,” *IEEE Transactions on Automatic Control*, vol. 43, no. 7, pp. 881–900, 1998.

- [17] A. Nedic and D. P. Bertsekas, “Incremental subgradient methods for nondifferentiable optimization,” *SIAM Journal on Optimization*, vol. 12, no. 1, pp. 109–138, 2001.
- [18] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, UK: Cambridge University Press, 2004.
- [19] T. Ibaraki and N. Katoh, *Resource Allocation Problems: Algorithmic Approaches*. Cambridge, UK: MIT Press, 1988.
- [20] L. Hurwicz, “The design of mechanisms for resource allocation,” *The American Economic Review*, vol. 63, no. 2, pp. 1–30, 1973.
- [21] K. Arrow and L. Hurwicz, “Decentralization and computation in resource allocation,” *Essays in Economics and Econometrics*, pp. 34–104, 1960.
- [22] G. M. Heal, “Planning without prices,” *Review of Economic Studies*, vol. 36, no. 107, pp. 347–362, 1969.
- [23] B. Yang and M. Johansson, “Distributed optimization and games: A tutorial overview,” in *Networked Control Systems*. London, UK: Springer, 2010, pp. 109–148.
- [24] Y. Ho, L. Servi, and R. Suri, “A class of center-free resource allocation algorithms,” *IFAC Proceedings Volumes*, vol. 13, no. 6, pp. 475–482, 1980.
- [25] L. Xiao and S. Boyd, “Optimal scaling of a gradient method for distributed resource allocation,” *Journal of Optimization Theory and Applications*, vol. 129, no. 3, pp. 469–488, 2006.
- [26] H. Lakshmanan and D. P. De Farias, “Decentralized resource allocation in dynamic networks of agents,” *SIAM Journal on Optimization*, vol. 19, no. 2, pp. 911–940, 2008.

- [27] I. Necoara, “A random coordinate descent method for large-scale resource allocation problems,” in *IEEE Conference on Decision and Control*. IEEE, 2012, pp. 4474–4479.
- [28] E. Ghadimi, I. Shames, and M. Johansson, “Multi-step gradient methods for networked optimization,” *IEEE Transactions on Signal Processing*, vol. 61, no. 21, pp. 5417–5429, 2013.
- [29] E. Wei, A. Ozdaglar, and A. Jadbabaie, “A distributed newton method for network utility maximization I: Algorithm,” *IEEE Transactions on Automatic Control*, vol. 58, no. 9, pp. 2162–2175, 2013.
- [30] A. Cherukuri and J. Cortés, “Initialization-free distributed coordination for economic dispatch under varying loads and generator commitment,” *Automatica*, vol. 74, pp. 183–193, 2016.
- [31] P. Yi, Y. Hong, and F. Liu, “Initialization-free distributed algorithms for optimal resource allocation with feasibility constraints and application to economic dispatch of power systems,” *Automatica*, no. 74, pp. 259–269, 2016.
- [32] X. Zeng, P. Yi, Y. Hong, and L. Xie, “Continuous-time distributed algorithms for extended monotropic optimization problems,” *arXiv preprint arXiv:1608.01167*, 2016.
- [33] Z. Deng, S. Liang, and Y. Hong, “Distributed continuous-time algorithms for resource allocation problems over weight-balanced digraphs,” *IEEE Transactions on Cybernetics*, vol. 48, no. 11, pp. 3116–3125, 2017.
- [34] P. Yi, J. Lei, and Y. Hong, “Distributed resource allocation over random networks based on stochastic approximation,” *Systems & Control Letters*, vol. 114, pp. 44–51, 2018.
- [35] S. Liang, X. Zeng, and Y. Hong, “Distributed sub-optimal resource allocation over weight-balanced graph via singular perturbation,” *Automatica*, vol. 95, pp. 222–228, 2018.

- [36] A. Cherukuri and J. Cortés, “Distributed generator coordination for initialization and anytime optimization in economic dispatch,” *IEEE Transactions on Control of Network Systems*, vol. 2, no. 3, pp. 226–237, 2015.
- [37] M. Zhang, W. Quan, N. Cheng, Q. Wu, J. Zhu, R. Zheng, and K. Li, “Distributed conditional gradient online learning for iot optimization,” *IEEE Internet of Things Journal*, 2019, doi: 10.1109/JIOT.2019.2919562.
- [38] J. M. Guerrero, J. C. Vasquez, J. Matas, L. G. De Vicuña, and M. Castilla, “Hierarchical control of droop-controlled AC and DC microgrids - A general approach toward standardization,” *IEEE Transactions on Industrial Electronics*, vol. 58, no. 1, pp. 158–172, 2011.
- [39] J. M. Guerrero, M. Chandorkar, T.-L. Lee, and P. C. Loh, “Advanced control architectures for intelligent microgrids - Part I: Decentralized and hierarchical control,” *IEEE Transactions on Industrial Electronics*, vol. 60, no. 4, pp. 1254–1262, 2013.
- [40] A. S. Dobakhshari, S. Azizi, and A. M. Ranjbar, “Control of microgrids: Aspects and prospects,” in *IEEE International Conference on Networking, Sensing and Control*. IEEE, 2011, pp. 38–43.
- [41] K. Padiyar, *Power System Dynamics*. Hyderabad, India: BS Publications, 2008.
- [42] A. Conejo, J. Arroyo, N. Alguacil, and A. Guijarro, “Transmission loss allocation: a comparison of different practical algorithms,” *IEEE Transactions on Power Systems*, vol. 17, no. 3, pp. 571–576, 2002.
- [43] I. G. Damousis, A. G. Bakirtzis, and P. S. Dokopoulos, “Network-constrained economic dispatch using real-coded genetic algorithm,” *IEEE Transactions on Power Systems*, vol. 18, no. 1, pp. 198–205, 2003.

- [44] P. Yi, Y. Hong, and F. Liu, “Distributed gradient algorithm for constrained optimization with application to load sharing in power systems,” *Systems & Control Letters*, vol. 83, pp. 45–52, 2015.
- [45] X. Li, L. Fang, Z. Lu, J. Zhang, and H. Zhao, “A line flow granular computing approach for economic dispatch with line constraints,” *IEEE Transactions on Power Systems*, vol. 32, no. 6, pp. 4832–4842, 2017.
- [46] M. I. Abouheaf, S. Haesaert, W. Lee, and F. L. Lewis, “Q-learning with eligibility traces to solve non-convex economic dispatch problems,” *International Journal of Electrical and Computer Engineering*, vol. 6, no. 7, pp. 1390–1396, 2012.
- [47] M. Kheshti, X. Kang, Z. Bie, Z. Jiao, and X. Wang, “An effective lightning flash algorithm solution to large scale non-convex economic dispatch with valve-point and multiple fuel options on generation units,” *Energy*, vol. 129, pp. 1–15, 2017.
- [48] C. Shilaja and K. Ravi, “Optimization of emission/economic dispatch using euclidean affine flower pollination algorithm (efpa) and binary fpa (bfpa) in solar photo voltaic generation,” *Renewable Energy*, vol. 107, pp. 550–566, 2017.
- [49] F. P. Mahdi, P. Vasant, V. Kallimani, J. Watada, P. Y. S. Fai, and M. Abdullah-Al-Wadud, “A holistic review on optimization strategies for combined economic emission dispatch problem,” *Renewable and Sustainable Energy Reviews*, vol. 81, pp. 3006–3020, 2018.
- [50] X. Xia and A. Elaiw, “Optimal dynamic economic dispatch of generation: a review,” *Electric Power Systems Research*, vol. 80, no. 8, pp. 975–986, 2010.
- [51] M. Zaman, S. M. Elsayed, T. Ray, and R. A. Sarker, “Evolutionary algorithms for dynamic economic dispatch problems,” *IEEE Transactions on Power Systems*, vol. 31, no. 2, pp. 1486–1495, 2016.

- [52] N. Rahbari-Asr, U. Ojha, Z. Zhang, and M.-Y. Chow, “Incremental welfare consensus algorithm for cooperative distributed generation/demand response in smart grid,” *IEEE Transactions on Smart Grid*, vol. 5, no. 6, pp. 2836–2845, 2014.
- [53] N. I. Nwulu and X. Xia, “Optimal dispatch for a microgrid incorporating renewables and demand response,” *Renewable Energy*, vol. 101, pp. 16–28, 2017.
- [54] C. E. Lin, S. T. Chen, and C. L. Huang, “A direct newton-raphson economic dispatch,” *IEEE Transactions on Power Systems*, vol. 7, no. 3, pp. 1149–1154, 1992.
- [55] J. Zhu, *Optimization of Power System Operation*. New York, USA: John Wiley & Sons, 2015, vol. 47.
- [56] D. C. Walters and G. B. Sheble, “Genetic algorithm solution of economic dispatch with valve point loading,” *IEEE Transactions on Power Systems*, vol. 8, no. 3, pp. 1325–1332, 1993.
- [57] L. Jebaraj, C. Venkatesan, I. Soubache, and C. C. A. Rajan, “Application of differential evolution algorithm in static and dynamic economic or emission dispatch problem: a review,” *Renewable and Sustainable Energy Reviews*, vol. 77, pp. 1206–1220, 2017.
- [58] M. H. R. Nascimento, M. V. A. Nunes, J. L. M. Rodríguez, and J. C. Leite, “A new solution to the economical load dispatch of power plants and optimization using differential evolution,” *Electrical Engineering*, vol. 99, no. 2, pp. 561–571, 2017.
- [59] S. Xia, Y. Xu, and X. Ge, “An improved multi-objective differential evolution algorithm for active power dispatch in power system with wind farms,” in *Advanced Computational Methods in Energy, Power, Electric Vehicles, and Their Integration*. Singapore: Springer, 2017, pp. 643–652.

- [60] Z.-L. Gaing, "Particle swarm optimization to solving the economic dispatch considering the generator constraints," *IEEE Transactions on Power Systems*, vol. 18, no. 3, pp. 1187–1195, 2003.
- [61] W. T. Elsayed, Y. G. Hegazy, M. S. El-bages, and F. M. Bendary, "Improved random drift particle swarm optimization with self-adaptive mechanism for solving the power economic dispatch problem," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 3, pp. 1017–1026, 2017.
- [62] K. Wong and C. Fung, "Simulated annealing based economic dispatch algorithm," in *IEE Proceedings C (Generation, Transmission and Distribution)*, vol. 140, no. 6. IET, 1993, pp. 509–515.
- [63] J. d. A. B. Júnior, M. V. A. Nunes, M. H. R. Nascimento, J. L. M. Rodríguez, and J. C. Leite, "Solution to economic emission load dispatch by simulated annealing: case study," *Electrical Engineering*, vol. 100, no. 2, pp. 749–761, 2018.
- [64] D. Aydin, G. Yavuz, S. Özyön, C. Yaşar, and T. Stützle, "Artificial bee colony framework to non-convex economic dispatch problem with valve point effects: a case study," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. ACM, 2017, pp. 1311–1318.
- [65] S. Sharifi, M. Sedaghat, P. Farhadi, N. Ghadimi, and B. Taheri, "Environmental economic dispatch using improved artificial bee colony algorithm," *Evolving Systems*, vol. 8, no. 3, pp. 233–242, 2017.
- [66] J. X. V. Neto, G. Reynoso-Meza, T. H. Ruppel, V. C. Mariani, and L. dos Santos Coelho, "Solving non-smooth economic dispatch by a new combination of continuous grasp algorithm and differential evolution," *International Journal of Electrical Power & Energy Systems*, vol. 84, pp. 13–24, 2017.
- [67] E. Naderi, A. Azizivahed, H. Narimani, M. Fathi, and M. R. Narimani, "A comprehensive study of practical economic dispatch problems by a new hy-

- brid evolutionary algorithm,” *Applied Soft Computing*, vol. 61, pp. 1186–1206, 2017.
- [68] H. Farhangi, “The path of the smart grid,” *Power and Energy Magazine*, vol. 8, no. 1, pp. 18–28, 2010.
- [69] Z. Zhang and M. Y. Chow, “Convergence analysis of the incremental cost consensus algorithm under different communication network topologies in a smart grid,” *IEEE Transactions on Power Systems*, vol. 27, no. 4, pp. 1761–1768, 2012.
- [70] S. Kar and G. Hug, “Distributed robust economic dispatch in power systems: A consensus+ innovations approach,” in *Power and Energy Society General Meeting*. IEEE, 2012, pp. 1–8.
- [71] S. Yang, S. Tan, and J. X. Xu, “Consensus based approach for economic dispatch problem in a smart grid,” *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 4416–4426, 2013.
- [72] H. Xing, Y. Mou, M. Fu, and Z. Lin, “Distributed bisection method for economic power dispatch in smart grid,” *IEEE Transactions on Power Systems*, vol. 30, no. 6, pp. 3024–3035, 2015.
- [73] Y. Zhang, N. Rahbari Asr, and M. Y. Chow, “A robust distributed system incremental cost estimation algorithm for smart grid economic dispatch with communications information losses,” *Journal of Network and Computer Applications*, vol. 59, pp. 315–324, 2016.
- [74] C. Li, X. Yu, W. Yu, T. Huang, and Z. W. Liu, “Distributed event-triggered scheme for economic dispatch in smart grids,” *IEEE Transactions on Industrial Informatics*, vol. 12, no. 5, pp. 1775–1785, 2015.
- [75] F. Guo, C. Wen, J. Mao, and Y.-D. Song, “Distributed economic dispatch for smart grids with random wind power,” *IEEE Transactions on Smart Grid*, vol. 7, no. 3, pp. 1572–1583, 2016.

- [76] F. Guo, C. Wen, J. Mao, G. Li, and Y.-D. Song, “A distributed hierarchical algorithm for multi-cluster constrained optimization,” *Automatica*, vol. 77, pp. 230–238, 2017.
- [77] H. Pourbabak, J. Luo, T. Chen, and W. Su, “A novel consensus-based distributed algorithm for economic dispatch based on local estimation of power mismatch,” *IEEE Transactions on Smart Grid*, vol. 9, no. 6, pp. 5930–5942, 2018.
- [78] Z. Tang, D. J. Hill, and T. Liu, “A novel consensus-based economic dispatch for microgrids,” *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 3920–3922, 2018.
- [79] R. Wang, Q. Li, B. Zhang, and L. Wang, “Distributed consensus based algorithm for economic dispatch in a microgrid,” *IEEE Transactions on Smart Grid*, vol. 10, no. 4, pp. 3630–3640, 2019.
- [80] A. Y. Popkov, “Gradient methods for nonstationary unconstrained optimization problems,” *Automation and Remote Control*, vol. 66, no. 6, pp. 883–891, 2005.
- [81] W. Su, “Traffic engineering and time-varying convex optimization,” Ph.D. dissertation, The Pennsylvania State University, Pennsylvania, USA, May 2009.
- [82] V. M. Zavala and M. Anitescu, “Real-time nonlinear optimization as a generalized equation,” *SIAM Journal on Control and Optimization*, vol. 48, no. 8, pp. 5444–5467, 2010.
- [83] A. Simonetto, A. Mokhtari, A. Koppel, G. Leus, and A. Ribeiro, “A class of prediction-correction methods for time-varying convex optimization,” *IEEE Transactions on Signal Processing*, vol. 64, no. 17, pp. 4576–4591, 2016.
- [84] M. Fazlyab, S. Paternain, V. M. Preciado, and A. Ribeiro, “Prediction-correction interior-point method for time-varying convex optimization,” *IEEE Transactions on Automatic Control*, vol. 63, no. 7, pp. 1973–1986, 2018.

- [85] M. J. Neely and H. Yu, “Online convex optimization with time-varying constraints,” *arXiv preprint arXiv:1702.04783*, 2017.
- [86] A. Simonetto, “Time-varying convex optimization via time-varying averaged operators,” *arXiv preprint arXiv:1704.07338*, 2017.
- [87] —, “Dual prediction-correction methods for linearly constrained time-varying convex programs,” *IEEE Transactions on Automatic Control*, vol. 64, no. 8, pp. 3355–3361, 2019.
- [88] M. Ye and G. Hu, “Distributed extremum seeking for constrained networked optimization and its application to energy consumption control in smart grid,” *IEEE Transactions on Control Systems Technology*, vol. 24, no. 6, pp. 2048–2058, 2016.
- [89] S. Rahili and W. Ren, “Distributed continuous-time convex optimization with time-varying cost functions,” *IEEE Transactions on Automatic Control*, vol. 62, no. 4, pp. 1590–1605, 2017.
- [90] M. Ye and G. Hu, “Distributed optimization for systems with time-varying quadratic objective functions,” in *IEEE Conference on Decision and Control*. IEEE, 2015, pp. 3285–3290.
- [91] C. Sun, M. Ye, and G. Hu, “Distributed time-varying quadratic optimization for multiple agents under undirected graphs,” *IEEE Transactions on Automatic Control*, vol. 62, no. 7, pp. 3687–3694, 2017.
- [92] A. Simonetto, A. Koppel, A. Mokhtari, G. Leus, and A. Ribeiro, “Decentralized prediction-correction methods for networked time-varying convex optimization,” *IEEE Transactions on Automatic Control*, vol. 62, no. 11, pp. 5724–5738, 2017.
- [93] W. Ren and R. W. Beard, *Distributed Consensus in Multi-Vehicle Cooperative Control*. London, UK: Springer, 2008.

- [94] R. A. Freeman, P. Yang, and K. M. Lynch, “Stability and convergence properties of dynamic average consensus estimators,” in *IEEE Conference on Decision and Control*, 2006, pp. 338–343.
- [95] H.-B. Dürr, E. Saka, and C. Ebenbauer, “A smooth vector field for quadratic programming,” in *IEEE Conference on Decision and Control*. IEEE, 2012, pp. 2515–2520.
- [96] A. Cherukuri, B. Gharesifard, and J. Cortes, “Saddle-point dynamics: conditions for asymptotic stability of saddle points,” *SIAM Journal on Control and Optimization*, vol. 55, no. 1, pp. 486–511, 2017.
- [97] A. Cherukuri and J. Cortés, “Asymptotic stability of saddle points under the saddle-point dynamics,” in *American Control Conference*, 2015, pp. 2020–2025.
- [98] Y. Nesterov and V. Spokoiny, “Random gradient-free minimization of convex functions,” *Foundations of Computational Mathematics*, vol. 17, no. 2, pp. 527–566, 2017.
- [99] A. P. Ruszczyński, *Nonlinear Optimization*. Princeton, USA: Princeton university press, 2006.
- [100] B. Gharesifard and J. Cortés, “Distributed continuous-time convex optimization on weight-balanced digraphs,” *IEEE Transactions on Automatic Control*, vol. 59, no. 3, pp. 781–786, 2014.
- [101] M. Ye and G. Hu, “Distributed extremum seeking for constrained networked optimization and its application to energy consumption control in smart grid,” *IEEE Transactions on Control Systems Technology*, vol. 24, no. 6, pp. 2048–2058, 2016.
- [102] B. Johansson, T. Keviczky, M. Johansson, and K. H. Johansson, “Subgradient methods and consensus algorithms for solving convex optimization problems,” in *IEEE Conference on Decision and Control*. IEEE, 2008, pp. 4185–4190.

- [103] A. Nedic, A. Ozdaglar, and P. A. Parrilo, “Constrained consensus and optimization in multi-agent networks,” *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 922–938, 2010.
- [104] S. S. Kia, J. Cortés, and S. Martínez, “Distributed convex optimization via continuous-time coordination algorithms with discrete-time communication,” *Automatica*, vol. 55, pp. 254–264, 2015.
- [105] Z. Qiu, S. Liu, and L. Xie, “Distributed constrained optimal consensus of multi-agent systems,” *Automatica*, vol. 68, pp. 209–215, 2016.
- [106] X. Zeng, P. Yi, and Y. Hong, “Distributed continuous-time algorithm for constrained convex optimizations via nonsmooth analysis approach,” *IEEE Transactions on Automatic Control*, vol. 62, no. 10, pp. 5227–5233, 2017.
- [107] P. Lin, Ren, and J. A. Farrell, “Distributed continuous-time optimization: nonuniform gradient gains, finite-time convergence, and convex constraint set,” *IEEE Transactions on Automatic Control*, vol. 62, no. 5, pp. 2239–2253, 2017.
- [108] T.-H. Chang, M. Hong, and X. Wang, “Multi-agent distributed optimization via inexact consensus ADMM,” *IEEE Transactions on Signal Processing*, vol. 63, no. 2, pp. 482–497, 2015.
- [109] A. Nedić, A. Olshevsky, and W. Shi, “Improved convergence rates for distributed resource allocation,” in *IEEE Conference on Decision and Control*. IEEE, 2018, pp. 172–177.
- [110] P. Frihauf, M. Krstic, and T. Basar, “Nash equilibrium seeking in noncooperative games,” *IEEE Transactions on Automatic Control*, vol. 57, no. 5, pp. 1192–1207, 2011.
- [111] L. J. Ratliff, S. A. Burden, and S. S. Sastry, “On the characterization of local Nash equilibria in continuous games,” *IEEE Transactions on Automatic Control*, vol. 61, no. 8, pp. 2301–2307, 2016.

- [112] M. Ye and G. Hu, “Game design and analysis for price-based demand response: An aggregate game approach,” *IEEE Transactions on Cybernetics*, vol. 47, no. 3, pp. 720–730, 2016.
- [113] —, “Distributed Nash equilibrium seeking by a consensus based approach,” *IEEE Transactions on Automatic Control*, vol. 62, no. 9, pp. 4811–4818, 2017.
- [114] C. Sun and G. Hu, “Distributed Nash equilibrium seeking for generalized convex games with shared constraints,” in *Journal of Physics: Conference Series*, vol. 1016, no. 1. IOP Publishing, 2018, p. 012012.
- [115] —, “Distributed Nash equilibrium seeking for a generalized convex game with nonsmooth objective functions and certain nonsmooth constraints,” in *IEEE International Conference on Information and Automation*. IEEE, 2018.
- [116] M. Zhu and S. Martínez, “On distributed convex optimization under inequality and equality constraints,” *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 151–164, 2012.
- [117] T.-H. Chang, A. Nedić, and A. Scaglione, “Distributed constrained optimization by consensus-based primal-dual perturbation method,” *IEEE Transactions on Automatic Control*, vol. 59, no. 6, pp. 1524–1538, 2014.
- [118] D. Yuan, D. W. Ho, and S. Xu, “Regularized primal-dual subgradient method for distributed constrained optimization,” *IEEE Transactions on Cybernetics*, vol. 46, no. 9, pp. 2109–2118, 2016.
- [119] G. Chen, J. Ren, and E. N. Feng, “Distributed finite-time economic dispatch of a network of energy resources,” *IEEE Transactions on Smart Grid*, vol. 8, no. 2, pp. 822–832, 2017.
- [120] C. Li, X. Yu, T. Huang, and X. He, “Distributed optimal consensus over resource allocation network and its application to dynamical economic dispatch,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 6, pp. 2407–2418, 2018.

- [121] G. Chen and Z. Li, "A fixed-time convergent algorithm for distributed convex optimization in multi-agent systems," *Automatica*, vol. 95, pp. 539–543, 2018.
- [122] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.
- [123] F. Chen and W. Ren, "Distributed consensus in networks," *Wiley Encyclopedia of Electrical and Electronics Engineering*, pp. 1–15, 2016, doi: 10.1002/047134608X.W8314.
- [124] T. Yang, X. Yi, J. Wu, Y. Yuan, D. Wu, Z. Meng, Y. Hong, H. Wang, Z. Lin, and K. H. Johansson, "A survey of distributed optimization," *Annual Reviews in Control*, vol. 47, pp. 278–305, 2019.
- [125] H. K. Khalil and J. Grizzle, *Nonlinear Systems*. Upper Saddle River, NJ: Prentice Hall, 2002.
- [126] A. Menon and J. S. Baras, "Collaborative extremum seeking for welfare optimization," in *IEEE Conference on Decision and Control*. IEEE, 2014, pp. 346–351.
- [127] X. Yu and Y. Xue, "Smart grids: A cyber-physical systems perspective," *Proceedings of the IEEE*, vol. 104, no. 5, pp. 1058–1070, 2016.
- [128] H. Shirzeh, "Plug and play in the smart grid," Ph.D. dissertation, University of Wollongong, Wollongong, Australia, May 2016.
- [129] "Power system test case archive [online]," *Available:* <http://www.ee.washington.edu/research/pstca/>.
- [130] "California iso - today's outlook [online]," *Available:* <http://www.caiso.com/TodaysOutlook/Pages/default.aspx>.

- [131] L. Bai, M. Ye, C. Sun, and G. Hu, “Distributed economic dispatch control via saddle point dynamics and consensus algorithms,” *IEEE Transactions on Control Systems Technology*, vol. 27, no. 2, pp. 898–905, 2019.
- [132] B. T. Polyak, *Introduction to Optimization*. New York, USA: Optimization Software, 1987.
- [133] J.-J. Slotine and S. S. Sastry, “Tracking control of non-linear systems using sliding surfaces, with application to robot manipulators,” *International Journal of Control*, vol. 38, no. 2, pp. 465–492, 1983.
- [134] Y. Hong, J. Hu, and L. Gao, “Tracking control for multi-agent consensus with an active leader and variable topology,” *Automatica*, vol. 42, no. 7, pp. 1177–1182, 2006.
- [135] S. P. Bhat and D. S. Bernstein, “Finite-time stability of continuous autonomous systems,” *SIAM Journal on Control and Optimization*, vol. 38, no. 3, pp. 751–766, 2000.
- [136] L. Bai, C. Sun, Z. Feng, and G. Hu, “Distributed continuous-time resource allocation with time-varying resources under quadratic cost functions,” in *IEEE Conference on Decision and Control*. IEEE, 2018, pp. 823–828.
- [137] M. Ç. Pinar and S. A. Zenios, “On smoothing exact penalty functions for convex constrained optimization,” *SIAM Journal on Optimization*, vol. 4, no. 3, pp. 486–511, 1994.
- [138] D. Jeyakumar, T. Jayabarathi, and T. Raghunathan, “Particle swarm optimization for various types of economic dispatch problems,” *International Journal of Electrical Power & Energy Systems*, vol. 28, no. 1, pp. 36–42, 2006.
- [139] A. I. Selvakumar and K. Thanushkodi, “A new particle swarm optimization solution to nonconvex economic dispatch problems,” *IEEE Transactions on Power Systems*, vol. 22, no. 1, pp. 42–51, 2007.

- [140] Q. Qin, S. Cheng, X. Chu, X. Lei, and Y. Shi, "Solving non-convex/non-smooth economic load dispatch problems via an enhanced particle swarm optimization," *Applied Soft Computing*, vol. 59, pp. 229–242, 2017.
- [141] M. Mohammadian, A. Lorestani, and M. Ardehali, "Optimization of single and multi-areas economic dispatch problems based on evolutionary particle swarm optimization algorithm," *Energy*, vol. 161, pp. 710–724, 2018.
- [142] P.-H. Chen and H.-C. Chang, "Large-scale economic dispatch by genetic algorithm," *IEEE Transactions on Power Systems*, vol. 10, no. 4, pp. 1919–1926, 1995.
- [143] M. Nemati, M. Braun, and S. Tenbohlen, "Optimization of unit commitment and economic dispatch in microgrids based on genetic algorithm and mixed integer linear programming," *Applied Energy*, vol. 210, pp. 944–963, 2018.
- [144] M. Pandit, L. Srivastava, and M. Sharma, "Environmental economic dispatch in multi-area power system employing improved differential evolution with fuzzy selection," *Applied Soft Computing*, vol. 28, pp. 498–510, 2015.
- [145] D. C. Secui, "A new modified artificial bee colony algorithm for the economic dispatch problem," *Energy Conversion and Management*, vol. 89, pp. 43–62, 2015.
- [146] S. S. Kia, B. Van Scoy, J. Cortes, R. A. Freeman, K. M. Lynch, and S. Martinez, "Tutorial on dynamic average consensus: the problem, its applications, and the algorithms," *arXiv preprint arXiv:1803.04628*, 2018.
- [147] S. Ghadimi, G. Lan, and H. Zhang, "Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization," *Mathematical Programming*, vol. 155, no. 1-2, pp. 267–305, 2016.
- [148] H. Shuai, J. Fang, X. Ai, Y. Tang, J. Wen, and H. He, "Stochastic optimization of economic dispatch for microgrid based on approximate dynamic programming," *IEEE Transactions on Smart Grid*, vol. 10, no. 3, pp. 2440–2452, 2019.

- [149] X. Zhang, N. Li, and A. Papachristodoulou, “Achieving real-time economic dispatch in power networks via a saddle point design approach,” in *Proceedings of IEEE Power and Energy Society General Meeting*, 2015, pp. 1–5.
- [150] S. Trip, M. Bürger, and C. De Persis, “An internal model approach to (optimal) frequency regulation in power grids with time-varying voltages,” *Automatica*, vol. 64, pp. 240–253, 2016.