NOVEL COMPUTATIONAL METHODS FOR WIND FARM INTEGRATED POWER SYSTEMS USING COMPUTATIONAL INTELLIGENCE AND ADVANCED COMPUTING TECHNIQUES

YINGYING CHEN

B.E. (Software Engineering)

M.E. (Computer Software and Theory)

A thesis submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

School of Electrical Engineering

and Computer Science

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

February, 2014

DECLARATION

The thesis contains no material which has been accepted for the award of any other degree or diploma in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. I give consent to this copy of my thesis, when deposited in the University Library, being made available for loan and photocopying subject to the provisions of the Copyright Act 1968.

I hereby certify that the work embodied in this thesis contains published chapters/scholarly work of which I am a joint author. I have included as part of the thesis a written statement, endorsed by my supervisor, attesting to my contribution to the joint publications/scholarly work.

Author: ____________________

Yingying CHEN

Feb., 2014

Approved by Principle Supervisor: ____________________

Prof. Zhao Yang Dong

Feb., 2014

To my beloved family,

in loving memory of my grandfather.

ACKNOWLEDGEMENTS

I was fortunate to have enjoyed the support of a tremendous number of people throughout the course of my Ph.D. research work – my supervisors, colleagues and my family, without whom this dissertation would have been impossible.

I would first like to grateful thank my principle supervisor, Prof. Zhao Yang Dong, for his support, guidance and constant encouragement throughout my research work. His personality, scholarship, hard work, insight, and leadership have set a perfect example I wish to match someday. I also would like to appreciate my co-supervisor, Prof. Kit Po Wong (The University of Western Australia), who taught me how to think and develop research ideas. His encouragements gave me confidence to set higher goals.

I would like to appreciate my colleague Dr. Ke Meng. During my research study, he always kindly helps me to overcome the research difficulties, and gives me many valuable suggestions and research directions.

This thesis would not be completed without the helps from my colleagues in our group: Dr. Junhua Zhao, Dr. Yan Xu, Jing Qiu and Yu Zheng, who keep support me in many ways both in my work and life. I am pretty appreciated for their supports and friendships.

Most of all, I would like to thank my family, especially my parents, who raised me and financially supported me until I finished my M.Sc., thus enabling me to devote myself to my PhD study. I would also like to appreciate my husband for his love, encouragement and continuous support. And I also wish to dedicate this thesis to the memory of my dear grandfather, who passed away recently.

The scholarships from the Hong Kong Polytechnic University, Hong Kong, and the University of Newcastle, Australia, are also acknowledged.

ABSTRACT

Over the last few decades, due to the environmental concerns and the increase of energy demand, wind power has strongly penetrated the electricity power generation industry. Wind power is variable, uncertain, and intermittent. The system planning, operation and control associated with wind farms are therefore considerably different from conventional power systems. Thus, the integration of wind farms has become the biggest challenges for system operators.

The research described in this thesis focuses on the novel computational methods used to address the challenges in wind farm integrated power systems. The computational intelligence (CI) methods and advanced computing techniques are researched and utilized to address the system planning, operation and control issues, which mainly involves optimal power flow (OPF), wind farm collector system layout optimization, and power dispatch problems for wind farm integrated power systems. Based on CI and advanced computing techniques, this research proposed a series of new methods, which can effectively overcome the shortcomings of the conventional approaches.

Chapter 1 specifies the research objectives and contributions behind this PhD research paper and also outlines the organization of this thesis.

Chapter 2 serves as a literature review of the techniques pertinent to the remainder of this thesis. It includes a brief description and discussion on wind power energy in emerging power systems, the advanced computing techniques for modern power systems, and the application of computational intelligence methods in power systems.

Chapter 3 presents a computational grid platform for distributed heterogeneous power systems and a cloud computing based information infrastructure for future power systems. Useful guidelines are also drawn for power engineers to construct the practical computing platforms for large-scale power systems.

Chapter 4 proposes a multi-constrained OPF (MCOPF) model with advanced differential evolution algorithms. This problem considers discrete control variables, as well as several practical operation constraints, including transient stability constraints, valve-point effects, POZ of generators, and branch flow thermal constraints. Moreover, cloud computing techniques are utilized for the parallelized optimization of this MCOPF problem, followed with simulation cases to demonstrate the practicability of the proposed approaches.

Chapter 5 presents a new and efficient collector system layout optimization (CSLO) model for large-scale offshore wind farms, which considers multiple substations and cable types, and focuses on cable topology optimization among wind turbines and substations with the objective to minimize the overall investment and maintenance cost, as well as the levelized power losses cost, while considering the network reliability and operational constraints simultaneously.

Chapter 6 proposes an optimal short-term wind farm dispatch model and an efficient method with battery energy system (BESS) for better integration of wind energy into power systems. The effectiveness of the proposed method is tested with wind farm case studies to demonstrate that the optimal plan of battery charging and discharging processes, and wind energy shedding can help reduce the fast intermittency and high fluctuation of wind power to meet the grid requirement.

Chapter 7 presents the conclusions and future direction of this research.

TABLE OF CONTENTS

Chapter 1 Introduction 1
1.1 Overview 2
1.2 Research Objectives and Contributions 2
1.3 Thesis Outline 5
1.4 List of Publications 6
Chapter 2 Background 11
2.1 Wind Energy in Emerging Power Systems 12
2.1.1 Present Status and Challenges in Wind Farm Integrated Power Systems 12
2.1.2 Collector Layout of Offshore Wind Farms 12
2.1.3 Wind Power Forecasting 13
2.1.4 Integrating Battery Energy Storage System into Wind Farms 14
2.2 Advanced Computing Techniques for Modern Power Systems 16
2.2.1 Advanced Computing Methods for Smart Grid 16
2.2.2 Grid Computing for Integrating Distributed Heterogeneous Resources 17
2.2.3 Cloud Computing based Data Center 18
2.3 Application of Computational Intelligence in Power Systems 19
2.3.1 Various Techniques of Computational Intelligence 19
2.3.2 Potential Research Area in Power Systems Using Computational Intelligence 20
2.4 Summary 20
Chapter 3 Advanced Computing Platforms for Modern Power Systems 23
3.1 Introduction 24
3.2 Grid Computing Technique 24
3.2.1 Grid Computing Introduction 24
3.2.2 Grid Computing System Architecture 25
3.3 Opportunity and Challenges of Grid Computing in Power Systems 26
3.3.1 Distributed Monitoring and Control 26
3.3.2 Grid Computing vs Other Computing Techniques 27
3.3.3 Fundamental Computation 28
3.3.4 Power System Information Grid 29
3.3.5 Reliability & Security Analysis 29
3.3.6 Power Market Grid 29
3.3.7 System Planning & Scheduling 30

3.3.8 Grid Computing & Smart Grid 30
3.4 A Computational Grid Platform for Distributed Heterogeneous Power Systems 31
3.4.1 System Model 31
3.4.2 Types of Grid Systems and Problem Description 32
3.4.3 Study Methodology 33
3.4.4 Grid System Simulation 39
3.4.5 Impact of Dynamic Factors in Small Scale Grid System 41
3.4.6 Evaluating the Improved Economic-based Scheduling Framework in Large-Scale Grid Systems 46
3.5 Cloud Computing Platform for Future's Power Systems 51
3.5.1 Cloud Data Centre for Smart Grid 51
3.5.2 Cloud Computing-based Information Infrastructure 54
3.5.3 Characteristics of the Infrastructure Model 55
3.5.4 IaaS-based Application in Power Systems 58
3.5.5 PaaS & SaaS-based Application in Power Systems 60
3.6 Summary 64
Chapter 4 Multi-Constrained Optimal Power Flow 65
4.1 Introduction 66
4.2 Traditional Optimal Power Flow 66
4.2.1 Formulation of OPF 66
4.2.2 Optimization Techniques 67
4.3 Practical Operation Constraints 68
4.3.1 Generator Valve Point Effects 68
4.3.2 Generator Prohibited Operating Zones 68
4.3.3 Transformer Discrete Control Region 69
4.3.4 Thermal Constraints 69
4.3.5 Transient Stability Constraints 69
4.4 Proposed Multi-Constrained OPF Model 70
4.4.1 Formulation of MCOPF 70
4.4.2 Handling of Discrete Variables 75
4.4.3 Fitness Evaluation Function 75
4.4.4 Transient Stability Assessment 76
4.5 Advanced Differential Evolution Algorithms for MCOPF 77
4.5.1 Conventional Differential Evolution Algorithm 77

4.5.2 Self-Adaptive Differential Evolution Algorithm 79
4.5.3 Opposition-based Differential Evolution Algorithm 80
4.5.4 SaDE-based MCOPF procedure 82
4.5.5 ODE-based MCOPF procedure 86
4.6 Applying Cloud Computing Techniques for MCOPF Problem 88
4.6.1 Cloud Computing Platform 88
4.6.2 Parallelized Optimization of MCOPF 89
4.7 Comparison Case Studies 89
4.7.1 Case A: New England 10-generator, 39-bus system 90
4.7.2 Case B: Cloud Computing of SaDE-based MCOPF on IEEE 17-generator, 162-bus system 97
4.8 Summary 100
Chapter 5 Collector System Layout Optimization for Offshore Wind Farms 101
5.1 Introduction 102
5.2 Problem Description and Formulation 102
5.2.1 Collector System Layout of Offshore Wind Farms 102
5.3 Self-Adaptive Allocation Method 104
5.3.1 Wind Turbine Clustering 104
5.3.2 Wind Turbine Allocation 106
5.4 CSLO using Minimum Spanning Tree Algorithm 107
5.4.1 Introduction of Minimum Spanning Tree Algorithm 107
5.4.2 Optimization Procedure 108
5.5 Benders Decomposition Method for the Collector System Layout of Large-Scale Offshore Wind Farms 109
5.5.1 Introduction of Benders Decomposition 109
5.5.2 Benders Reformulation of the Proposed Optimizer 109
5.6 Comparison Case Studies 113
5.6.1 Case Settings 113
5.6.2 Case I: 30-WT Offshore Wind Farm 116
5.6.3 Case II: 100-WT Offshore Wind Farm 120
5.7 Summary 121
Chapter 6 Optimal Power Dispatch for Wind Farm Integrated Power Systems 123
6.1 Introduction 124
6.2 Wind Farm System with BESS 124
6.2.1 System Modeling 124
6.2.2 Battery Charging Issues 125

6.2.3 Wind Speed Forecasting 126
6.3 Proposed Power Dispatch Scheduling Method based on Model Predictive Control 127
6.3.1 Model Predictive Control based Dispatch Scheduling 128
6.3.2 Objective and Constraints 129
6.4 Distributed Imperialist Competitive Algorithm 130
6.4.1 Introduction of ICA method 130
6.4.2 Binary Structure of ICA 133
6.4.3 Distributed Process Architecture of ICA 134
6.5 Solution to the Dispatch Model based on Distributed Imperialist Competitive Algorithm. 137
6.5.1 ICA Coding 137
6.5.2 Optimization Procedure 138
6.6 Case Studies 139
6.6.1 Parameter Settings 139
6.6.2 Case I: without Forecasting Error 139
6.6.3 Case II: with Forecasting Error 142
6.7 Summary 145
Chapter 7 Conclusions and Future Research 147
7.1 Conclusions 148
7.2 Future Research 149
7.2.1 Computing Platform Improvement 149
7.2.2 CSLO Model Extension 150
7.2.3 Reliability Consideration 150
7.2.4 Wind Speed Data Refinement 150
Bibliography 153

LIST OF FIGURES

Fig. 3-1. A Typical GC Layered Architecture 25
Fig. 3-2. Layered model of computational grid platform.... 31
Fig. 3-3. Computational grid platform in Internet environment 32
Fig. 3-4. Working mechanism of the grid scheduler 36
Fig. 3-5. Procedure of the scheduling algorithm.... 38
Fig. 3-6. Simulation result of Group 1 when all resources keep stable.... 43
Fig. 3-7. Simulation result of Group 1 considering the availability variations of resources.... 43
Fig. 3-8. Simulation result of Group 2 when all resources keep stable.... 44
Fig. 3-9. Simulation result of Group 2 considering the availability variations of resources.... 44
Fig. 3-10. Simulation result of Group 3 when all resources keep available 45
Fig. 3-11. Simulation Result of Group 3 considering the availability variations of resources ... 45
Fig. 3-12. Number of feasible normal jobs of cost-optimization algorithm with Buyya's algorithm 48
Fig. 3-13. Number of feasible normal jobs of cost-optimization algorithm with improved algorithm 48
Fig. 3-14. Number of feasible emergency jobs of cost-optimization algorithm with Buyya's algorithm 49
Fig. 3-15. Number of feasible emergency jobs of cost-optimization algorithm with improved algorithm 49
Fig. 3-16. Comparison of application execution time of cost-optimization algorithm and time-optimization algorithm.... 50
Fig. 3-17. Comparison of application cost of cost-optimization algorithm and time-optimization algorithm.... 51
Fig. 3-18. Snapshot of Google's data center 52
Fig. 3-19.Cloud data center conceptual model for smart grid.... 52
Fig. 3-20. Service model of cloud computing.... 53
Fig. 3-21. Cloud computing infrastructure for smart grid.... 55
Fig. 3-22. Configuration profile of an 8-core cluster compute instance 59
Fig. 3-23. Architecture of Google App Engine 60
Fig. 3-24. Parallel framework of pattern discovery process 62
Fig. 3-25. The computation time (left) and network overhead (right) of PD on GAE.... 63
Fig. 3-26. The computation time comparison between the single PC-based PD and the cloud-based PD 63
Fig. 3-27. PD demonstration on 3-dimension data space.... 64
Fig. 4-1. Variation of *KE, PE,* and *TEF* along a stable trajectory 74
Fig. 4-2. Variation of *KE, PE,* and *TEF* along an unstable trajectory 74
Fig. 4-3. The one-dimensional presentation of solution *x* and the opposite solution $\breve{x}$, in [*a*, *b*], where *c* is the center of the searching space, i.e., $c = (x + \breve{x}) / 2$ 81
Fig. 4-4. Flowchart of the proposed SaDE-based MCOPF procedure.... 83

Fig. 4-5. Flowchart of the proposed ODE-based MCOPF procedure 87
Fig. 4-6. Cloud Computing deployment model of SaDE-based MCOPF 89
Fig. 4-7. The single line diagram of New England 10-generator, 39-bus test system. 91
Fig. 4-8. Average convergence curves of six heuristic methods for Case A-1. 94
Fig. 4-9. Average convergence curves of five methods for Case A-2... 95
Fig. 4-10. Average convergence curves of IEEE 17-Generator, 162-Bus system 99
Fig. 5-1. The proposed method for the optimal design of electrical layout............................... 108
Fig. 5-2. The proposed method for the CSLO problem.. 111
Fig. 5-3. The turbine layouts of 30-WT and 100-WT offshore wind farms 114
Fig. 5-4. The power output curve of Vestas V112/3.0MW offshore wind turbine 115
Fig. 5-5. Wind speed distribution and Weibull fitting 115
Fig. 5-6. The optimal layouts of 30-WT offshore wind farm 119
Fig. 5-7. The optimal layout of 100-WT OWF for Case 2 120
Fig. 5-8. The optimal layout of 100-WT OWF for Case 3 121
Fig. 6-1. Schematic of the grid-connected wind farm with BESS... 125
Fig. 6-2. Wind speed forecasting curve 127
Fig. 6-3. Schematic of MPC-based dispatch scheduling step.. 128
Fig. 6-4. Movement of the colony 132
Fig. 6-5. Distributed processing architecture of the ICA.. 135
Fig. 6-6. Re-assignment of the computing nodes 136
Fig. 6-7. Coding rules of the ICA solution method 137
Fig. 6-8.Flow chart of solution program... 138
Fig. 6-9. Wind power and penalty cost of wind farm for one day 140
Fig. 6-10. Wind farm output comparison through a day... 141
Fig. 6-11. Dispatch plan of step 1 and step 2 for time horizon.. 142
Fig. 6-12. Charging power and state of BESS... 143
Fig. 6-13. Wind farm output comparison through a week... 144
Fig. 6-14. Cost of the wind farm dispatch 144

LIST OF TABLES

Table 3-1 Resource Definition of Small Scale Grid System 34
Table 3-2. Scheduling Strategies of Proposed Framework 36
Table 3-3. Grid System Implementation on 30-Node Linux PCs 40
Table 3-4. Comparison of Application Cases between Sequential Computing and Grid Computing 41
Table 3-5. Configuration of Concurrent Monte-Carlo Probabilistic Power Flow Analysis 47
Table 4-1. Resource Definition of Small Scale Grid System 86
Table 4-2. Unit Ratings and Fuel Cost Coefficients with Valve-point Effects 90
Table 4-3. Parameters of EGA, EP, PSO, DE, SaDE & ODE 91
Table 4-4. Comparison Results among Seven Methods in Case A-1 93
Table 4-5. Best System Solution of Case A-1 Using SaDE 93
Table 4-6. Unit POZs Data for New England System 94
Table 4-7. Comparison Results among Five Methods in Case A-2 95
Table 4-8. Best System Solution of Case A-2 Using SaDE 96
Table 4-9. Unit POZs Data for IEEE 17-Generator, 162-Bus System 97
Table 4-10. Unit Ratings & Fuel Cost Coefficients with Valve-point Effects 98
Table 4-11. Simulation Results of IEEE 17-Generator, 162-Bus System 98
Table 4-12. Best Solution of IEEE 162-Bus System Using SaDE 99
Table 5-1. Electric Parameters of Cable 116
Table 5-2. Simulation Results for the 30-WTs Offshore Wind Farm 117
Table 5-3. Simulation Results for the 100-WTs Offshore Wind Farm 120
Table 6-1. Power change limit of wind farm 125
Table 6-2. Procedure of the Distributed ICA 136
Table 6-3. General Parameters of test system 139
Table 6-4. Optimal results at step for time horizon 140
Table 6-5. Operation cost comparison for Case I 141
Table 6-6. Operation cost comparison for Case II 144

Chapter 1

INTRODUCTION

1.1 Overview

Wind power, one of the most appealing sources of alternative energy, has gained widespread attentions during the last several years. The technology for wind power has become more advanced and the penetration is increasing rapidly around the world. Along with the proposal of the “smart grid” and the rapid development of wind farms, the complexity of power systems has increased. This chapter specifies the research objectives and contributions behind this PhD research and also outlines the organization of this thesis.

1.2 Research Objectives and Contributions

When incorporating large amount of wind power into electric power systems, a number of technical issues will be encountered that need solutions, especially for large-scale systems. Although integration of high levels of wind power into an existing transmission system does not require a major redesign, it necessitates additional control and compensating equipment to enable recovery from severe system disturbances. This project focuses mainly on the research of how to develop and apply advanced computational methods for modern power systems with the integration of wind farms.

This dissertation addresses four key research areas:

(i) Advanced computing platforms for power systems;

(ii) Multi-constrained optimal power flow (MCOPF);

(iii) Wind farm collector system layout optimization;

(iv) Power dispatch scheduling.

Throughout, new algorithms and approaches based on computational intelligence (CI) are proposed and utilized throughout this study. The unique contributions from this work are summarized as follows.

- **Contributions to computing platforms for power systems**

1) It proposes a computational grid platform for compute-intensive applications in modern power systems. Simulations on small-scale and large-scale grid systems are performed to study the impact of dynamic grid computing environments. Useful guidelines are drawn for power engineers to construct the practical grid system

platform. An economic-based grid scheduling framework, which shows its efficiency for scheduling concurrent compute-intensive applications of power systems with various requirements, is also proposed.

2) It proposes a cloud computing platform for future power systems. First, a cloud data centre model for smart grid with high penetration of renewable energy is constructed. Then, a cloud computing based information infrastructure, which can deliver multi-layer services to the participants of smart grid, ranging from IaaS to PaaS and SaaS, is modeled and presented. An analysis and discussion of how power system applications could be built based on the layered service model of cloud computing techniques is also presented.

- **Contributions to optimal power flow problem**

1) The work models a multi-constrained optimal power flow (MCOPF) problem, which considers discrete control variables and practical operation constraints, including transient stability constraints, valve-point effects, prohibited operating zones of generators, and branch flow thermal limits.
2) It proposes a robust and efficient framework to solve the MCOPF model. Two advanced differential evolution (DE) algorithms, self-adaptive DE (SaDE) and opposition-based DE (ODE), are presented for the framework. The simulation studies demonstrate that SaDE and ODE can address the limitations of conventional approaches and the classic differential evolution algorithm in solving this non-linear, non-convex, discrete and non-differentiable optimization problem.
3) It proposes a cloud computing based platform for the MCOPF problem. The platform structure is presented and the simulation shows its practicability for large-scale power systems.

- **Contributions to collector system layout problem of wind farm**

1) This paper proposes a new and efficient collector system layout optimization (CSLO) model for large-scale offshore wind farms, which considers multiple substations and cable types, and minimizes the annualized investment cost, maintenance cost, and the levelized power losses cost, while satisfying network reliability and operational

constraints.

2) It proposes a self-adaptive application (SAA) method for wind turbines, which can automatically obtain the locations of substations in a wind farm and allocate wind turbines to their nearest substations exclusively based on fuzzy c-means (FCM) clustering and binary integer programming (BIP) methods.
3) It proposes a Benders decomposition based method to solve the CSLO model, which is a mixed-integer nonlinear programming (MINLP) problem. The proposed method is simulated on two practical wind farms, and a minimum spanning tree (MST) method is also performed and compared with the proposed methods based on Benders decomposition. The simulation results show that the proposed CSLO model can be utilized as a reliable collector system layout tool for the large-scale offshore wind farms.

- **Contributions to system power dispatch problem**

1) It proposes an optimal power dispatch scheduling model for a wind farm integrated power system with battery energy storage system (BESS). The development of BESS enables renewable generation with flexible operation to meet the requirement of electric grid.
2) It proposes a wind speed forecasting method based on a forecasting software *OptiWind*, which highly customizes numerical weather prediction models and the latest statistical methods. Several important factors are considered in the forecasting model, which include temperature, seasonal weather, public holidays, and historical load data.
3) It proposes a distributed imperialist competitive (ICA) algorithm to solve the proposed optimization model. A model predictive control (MPC) based method is also applied for operation decisions in order to minimize the energy loss from wind farm and battery usage. The simulation studies demonstrate that with BESS and the proposed policies and methods, the wind farm operator can make beneficial decisions through the proposed optimal model. The ramping rate violation is also decreased significantly to satisfy the requirements of grid.

1.3 Thesis Outline

The presentation in this thesis is divided into seven chapters, and proceeds as follows:

Chapter 2 gives a comprehensive literature review of each research topic studied in this project. Specifically, these topics are:

1) The present status and challenges in wind farm integrated power systems;
2) Advanced computing for smart grid, especially grid computing and cloud computing techniques;
3) Various techniques of computational intelligence and the potential research area in power systems using computational intelligence.

Chapter 3 analyzes and discusses the prospective and challenges of advanced computing platforms for modern power systems, focuses on the research of grid computing and cloud computing technologies, and proposes a computational grid platform and a cloud computing based information infrastructure for power systems respectively.

Chapter 4 proposes a multi-constrained OPF (MCOPF) problem which is solved with advanced differential evolution algorithms. The MCOPF problem considers discrete control variables, as well as several practical constraints, including transient stability constraints, valve-point effects, prohibited operating zones of generators, and branch flow thermal constraints. Moreover, cloud computing techniques are utilized for the parallelized optimization of this MCOPF problem, followed by simulation cases to demonstrate the practicability of the proposed approaches. Moreover, a cloud computing platform is presented here and the simulation on a large-scale power system is also illustrated.

Chapter 5 proposes a new and efficient collector system layout optimization (CSLO) model for large-scale offshore wind farms to minimize the annualized investment cost, maintenance cost, and the levelized power losses cost, while satisfying network reliability and operational constraints, such as AC power flow constraints, capacity limits of substations and wind turbines, network radiality constraints, etc. Benders decomposition algorithm is applied here to solve this mixed-integer nonlinear programming problem. A self-adaptive allocation (SAA) method for wind turbines is proposed based on fuzzy c-means (FCM) clustering algorithm to exclusively allocate wind turbines to their nearest substations and obtain the topology structure of cables that are utilized to connect turbines and substations. A minimum spanning tree (MST) algorithm is utilized to compare with the proposed method based on Benders decomposition. Simulations on two practical wind farms are also included with the discussion in this study.

Chapter 6 proposes an optimal power dispatch scheduling for a wind farm integrated power system with battery energy storage system. A model predictive control-based method is applied to operation decisions to minimize the energy loss from the wind farm and battery usage while meeting the grid constraints. A distributed imperialist competitive algorithm is presented here to solve the proposed optimization problem. The effects of the proposed model, which are simulated on a benchmark wind farm system with and without wind speed forecasting error, are also illustrated

Chapter 7 presents the conclusions and future direction of this research.

1.4 List of Publications

The following publications are the major PhD research results of this candidate, and are included as part of this thesis. This candidate's contributions to these joint publications are also stated.

[P1] Y.Y. Chen, Z.Y. Dong, K. Meng, F.J. Luo, Z. Xu, K.P. Wong, "A Novel Technique for the Optimal Design of Offshore Wind Farm Electrical Layout", *Journal of Modern Power Systems and Clean Energy*, vol. 1, no. 3, pp. 258-263, 2013.

Contributions:

1) It proposed a novel electrical layout design optimization method for offshore wind farms, which is to minimize the investment costs of cable connection and the transmission power losses. The simulation results clearly demonstrated the feasibility and reliability of the proposed model for electrical layout design in offshore wind farms.

[P2] Feng Ji Luo, Zhao Yang Dong, Ying Ying Chen, Ke Meng, Guo Chen, Hui Qiao Tian, and Kit Po Wong, "A novel short-term dispatch scheme for wind farm with battery energy storage system," *Proc. of IEEE PES General. Meeting*, Canada, 2013.

Contributions:

1) It proposed a novel wind power forecasting approach, which retrieves the wind power data under three different scenarios, considering the uncertainty of wind power.
2) It analysed the performances of various batteries for improving the utilization of battery energy storage system (BESS) in wind farms, and simulated their charging and discharging behaviours in the benchmark tests.

[P3] Y.Y. Chen, C.Y. Chung, "Multi-Constrained Optimal Power Flow by an Opposition-Based Differential Evolution," *Proc. of IEEE PES General Meeting*, San Diego, California, USA, 2012.

Contributions:

1) It modelled a novel optimal power flow (OPF) model, which considers multiple practical operation constraints, such as valve point effects, prohibited operating zones of generators, and branch flow thermal limits.
2) It proposed a robust framework to handle with the proposed OPF model. An improved differential evolution (DE) algorithm, which is based on the opposition-based learning (OBL) method, was also presented. The simulation demonstrated the efficiency of the framework.

[P4] Feng Ji Luo, Zhao Yang Dong, Ying Ying Chen, Yan Xu, Ke Meng, and Kit Po Wong, "Hybrid cloud computing platform: the next generation IT backbone for smart grid," *Proc. of IEEE PES General Meeting*, San Diego, California, USA, 2012.

Contributions:

1) It analysed and discussed the prospective and challenges of cloud computing techniques for the smart grid.
2) It proposed a power community cloud data centre, which can be employed to manage the distributed and virtualized resources in power systems.

[P5] F.J. Luo, Z.Y. Dong, Y.Y. Chen, Eric Pozorski, Jing Qiu, Yu Zheng, Yan Xu and Ke Meng, “Constructing the power cloud data center to deliver multi-layer services for smart grid,” *Proc. Advances in Power System Control, Operation and Management International Conference* (APSCOM2012), Hong Kong, 2012.

Contributions:

1) It analysed and discussed the possibility of applying cloud computing technology into the development of the smart grid.
2) It modelled a cloud computing based conceptual framework as the “power cloud data centre” for the smart grid.

[P6] F.J. Luo, Z.Y. Dong, Can Wan, Y.Y. Chen, K. Meng, and K. P. Wong. “Applying computational grid technology to power system,” *Applied Mechanics and Materials, Information Technology for Manufacturing Systems*, vol. 58, pp.1442-1447, Jun. 2011.

Contributions:

1) It modelled a computational grid platform for solving the large-scale power system applications.
2) It simulated the framework in Monte-Carlo based probabilistic load flow problems. The test results showed the efficiency and robustness of the proposed platform.

[P7] Can Wan, Zhaoyang Dong, Fengji Luo, Rui Wang, Yingying Chen, Ke Meng, and Kitpo Wong, “A novel XML-based power resource modelling framework for power system heterogeneous data integrating,” *Applied Mechanics and Materials, Information Technology for Manufacturing Systems*, vol. 58, pp.1476-1481, 2011.

Contributions:

1) It analysed and discussed the heterogeneous data in power systems, and proposed a XML-based date integrating tool, which can transfer and store the heterogeneous data from different power utilities into the data centre with uniform format.

[P8] Y.Y. Chen, C.Y. Chung, F.J. Luo, Z.Y. Dong, and K.P. Wong, "An overview of grid computing application for distributed heterogeneous power systems: history, technologies, and prospect," *Proc. The 8th International Conference on Power System Control, Operations and Management (APSCOM)*, pp. 1-6, Hong Kong, 2009.

Contributions:

1) It reviewed the research activities of recent years in the field of grid computing application to power system problems, along with an introduction to the technologies utilized in grid computing research and application.
2) It discussed the opportunities and challenges of grid computing for future smart power systems.

Chapter 2

BACKGROUND

2.1 Wind Energy in Emerging Power Systems

2.1.1 Present Status and Challenges in Wind Farm Integrated Power Systems

Integration of renewable energy is currently one of the top concerns for utilities as they look to increase the percentage of clean energy into power grids and reduce the construction of fossil-based power plants. Wind power, one of the most appealing sources of alternative energy, has gained widespread attentions during the last years. With the rapid technological advances and growing manufacturing scale, wind power is now nearly at grid parity [1]. For example, the percentage of wind power penetration in Denmark is planned to reach 50% in 2025 [2]. Moreover, with the introduction of various emission reduction schemes, increasing the number of wind plants has been planned or installed around the world [3]. However, due to its stochastic nature, integration of wind energy into electric grid has become the biggest challenges for system operators. Variations of wind speed directly influences the power generated by wind farms, which means the output of fossil fuel-plants needs to be adjusted more frequently to cope with fluctuations in wind power output; this also causes difficulties in estimating suitable system reserve margins to ensure secure and reliable operations. As a result, high penetration of wind power will cause high potential risks and difficulties in system operations [4].

2.1.2 Collector Layout of Offshore Wind Farms

Wind energy, one of the fastest growing renewable energy sources for power generation, now meets a significant percentage of electrical demand worldwide [1,5]. Generally, a wind farm could be located onshore or offshore. In recent years, the offshore wind power has become the focus of the world's renewable energy development [6]. By 2013, the total installed capacity of offshore wind farms all over the world has reached up to 5589.57 MW, with around 62 offshore wind farms commissioned [7]. The offshore wind farms appear to be more advantageous because better wind sources are available offshore compared to onshore [3,8,9]. However, due to the large capacity of offshore wind farms, the long distance to shore substation, and the variable wind conditions, the capital investment costs for offshore wind power are much higher than those for onshore installations and operations. Unlike traditional power generating units, which are built around a few high rating generators within a centralized location, offshore wind farms aggregate the power generated by a great variety of small wind powered generators spread out over a large area. Traditionally, the energy generated by each wind turbine is collected and transferred to a substation via a network of cables, called a collector system. The increased size of wind turbines in large-scale wind farms has led to the problem of finding a suitable cable connection scheme that meets economic and operational needs simultaneously [10]. Therefore, the optimal layout of collector system plays a very important role in planning offshore wind farms.

There are many factors in the design and layout of offshore wind farms, including terrain, wind turbine microsites, reliability, power losses, economics, landowner requirements, and climate [11]. In a collector system, the electrical power losses primarily impact the economic evaluation of wind farm operations. However, shortening cable lengths may not always be available due to a group of constraints. Besides, in a real-world offshore wind farm, it is usually with single collector substation to collect and transfer the power outputs generated by wind turbines. This design may not guarantee the security and reliability of system operations because the wind power generation may be unavailable to the onshore power grid if the sole substation transformer breaks down. It is important that the collector system be planned for reliability and operability, especially for large-scale offshore wind farms.

2.1.3 Wind Power Forecasting

Over the last decades, researchers have made significant efforts to study the impacts of wind generation on the operation of power systems. A majority of researches focuses on developing forecasting tools to predict wind power production. Generally, current forecast methods can be classified in two ways: physical modelling approaches and statistical modelling techniques. In the physical modelling approaches, numerical weather prediction (NWP) is often used; in the statistical modelling approaches, the forecast is based on the time series of the measured wind power data and some techniques, such as time series model, artificial neural network (ANN), support vector machine (SVM) are employed to do the wind power forecast. Today, these two approaches are sometimes combined to reduce the forecast errors [12].

1) Numerical Weather Prediction

The numerical weather prediction (MWP) model is a powerful tool adopted by many researchers to study wind forecast [12, 13]. It is a kind of weather simulation method which uses the mathematical models of atmosphere and parameters of other processes to make the weather forecast for specific domain. The NWP model determines the future state of the atmosphere by numerically solving a set of equations representing the evolution process of the meteorological variables. Such variables include temperature, pressure, wind speed, wind angle, air density, and so on. The input data of NWP model is the observation data collected by radiosondes and weather satellites, and the outputs of the NWP model can then be converted to wind power output. The forecast accuracy of the NWP model is often affected by the integrity of the observed physical data of the resolution of the grid.

2) Machine Learning Prediction

Although wind speed is difficult to forecast by single predictor, composite forecast models can statistically produce an optimal forecast by computing prediction results from a number of different methods. The fundamental concept is that if the errors in the forecasts produced by different methods are unbiased and have a low degree of correlation with each other, the random errors from the individual forecasts will tend to offset each other, with the result that a composite of the forecasts will have lower errors than any individual forecast. Furthermore, in this composite model, results from different prediction models should be processed by an adaptive processing module which adjusts the weights of each model based on its prediction performance on a continuous basis. Meanwhile, human expert inputs are also need to select the preferred prediction results and incorporate them in such a way that best enhances the performance of the forecast tool.

2.1.4 Integrating Battery Energy Storage System into Wind Farms

Proponents of wind energy have spent decades obscuring the level of risk of imposing this resource on existing power systems. Comprehensive efforts have been made to study and develop advanced techniques to address the uncertainties brought about by wind power. Owing to the effective role in maintaining power system stability, energy storage systems (ESS) have attracted widespread support because they add significant flexibility to the operations of power systems with wind farms. The development of new ESS systems will be critical to effective levelling of the cyclic nature of these energy resources. Compared with other energy storage technologies, battery energy storage systems (BESS) are the most cost effective approach designed for short-term wind farm dispatch purposes. BESS was firstly proposed in the early 1980s to reinforce transmission and substation [14].Then, due to its attractive features, the potentials of applying BESS to power system problems were studied extensively and a number of concerns related to economics, reliability, operation, and impacts were presented [15].Ten years later, because the cost of BESS reduced dramatically, the modelling and economic analyses of BESS were again studied intensively [16,17]. Gradually, BESS became one of the most promising solutions for maintaining power system stability. In order to further reduce the cost of BESS projects, an optimal size decision technology was developed, which showed that BESS technology could be operated in a suitable and economical manner [18]. Some other investigations concentrated on the applications of BESS to frequency control [19-21] and power flow control [22]. Recent advances in energy storage technology provide an opportunity to using energy storage devices to address the intermittency challenges of sustainable energy. Combining BESS with wind farms can improve system availability because it increases the amount of wind power that can be penetrated into power grids without risking system stability. Various techniques have been proposed to address the coordination of BESS and wind power generation, with the view to increase the penetration level of wind energy in grids [23-30].

In general, how to use BESS techniques to effectively and economically address the wind power dispatch problem is one of the key issues in need of further study. Many efforts have been done in the area of integrating BESS into wind farm dispatch. Some literature [23] and [24] used simple schemes to change/discharge BESS with wind farms. Yoshimoto et al. [25] adopted a washout filter-based scheme to smooth out the short-term power fluctuations of a wind farm with vanadium redox-flow batteries. Teleke et al. [26] improved the above control scheme to address other constraints so that the BESS could be used to smooth the net power supplied to the system based on hourly dispatch levels. Alternatively, they also proposed an optimal control model to control the charge/discharge current of the lead-acid battery over a given time period [27], aimed at minimizing the deviations between the battery power output and the battery power reference. The above works mainly focused on smoothing the wind farm output, and did not address the issues of battery capacity determination and short-term dispatch implementation. In addition, frequently switching between charge/discharge modes will shorten the lifetime of the BESS because this use was not intended for their designs.

In order to stabilise the power output of wind farms, Yao et al. [28-30] proposed a dual WTG-BESS to do the short-term dispatch. This system includes an in-service battery and a stand-by battery, where the in-service battery provides constant power output to the grid and the stand-by battery is charged by wind power. The two batteries exchange roles when the in-service one is unable to sustain the delivery for constant power. By employing the dual- BESS scheme, the in-service battery can aggregate the dispensable energy to make the wind farm act as a conventional generator. The major limitation is the high cost. Because the power output to the grid of the wind farm is completely from the in-service battery, and all the wind power produced by the wind turbines is stored in the stand-by battery, it requires a large capacity for each battery, which leads to the high investment cost. In [31], the authors used decision tree to develop a rule-based real-time dispatch strategy to control the power output of the BESS.

2.2 Advanced Computing Techniques for Modern Power Systems

2.2.1 Advanced Computing Methods for Smart Grid

With the ever-increasing demand for electrical energy and the continual reform of power markets, modern power systems have become progressively large-scale, distributed, and complicated. Moreover, because of global environment concerns, especially global warming and energy shortage problems, the questions of how to reduce carbon footprint and conserve energy have been the most important issues in modern power systems. Many research studies on green and renewable energy sources have been carried out in this area [32,33], which had leads to the emergence of solar-turbines and wind-turbines. The next generation of power systems must be combined with larger numbers of small-scale, highly dispersed generation units that utilize diverse renewable energy sources. These systems will be characterised by huge scale networks that physically intermingle, along with virtual interconnections and interactions (i.e., a complex multi-organization business model). Recently, a new term has come out in the discussion of future power systems—“smart grid.” This concept includes all the parties involved in power systems, from energy supplier, energy transmission operator, and energy distribution operator, to energy customers. The development of the smart grid is pushing even further the requirements for intelligent computing in power system applications [34].Within the increasingly computer-dependent power system, high performance computing (HPC), parallel computing, and distributed computing have all been studied and deployed in power industries for providing supercomputing abilities and distributed operations.

2.2.2 Grid Computing for Integrating Distributed Heterogeneous Resources

In modern times, the continuous expansion of power systems and the increasing demand for energy have made power systems increasingly complex, resulting in many large-scale problems that require greater computational capacity to handle simulations and data sharing. PC clusters or supercomputers are being widely used to perform parallel computing as this has been proven to be much more efficient than a single machine for handling simulations [35-37]. However, there are still some problems. First, for large-scale networks, PC clusters or supercomputers show inadequate computational ability. Second, when engineers or scientists do not have such equipment (i.e., a PC cluster or supercomputer), they find it hard to perform such calculations.

In recent years, grid computing (GC), which provides an ideal solution for the above issues, springs up as a novel intelligent computing technology for the integration and management of heterogeneous computer resources from multiple administrative domains applied to a common task, especially for the problems which are compute-intensive or/and data-intensive [38, 39]. Grid computing provides powerful coordinated integration of a large amount of geographically distributed, heterogeneous resources through the Internet to provide a virtual platform for scientists and engineers that cannot be achieved by parallel computing. Relying on the powerful ability of grid computing to integrate various kinds of distributed resources, engineers can access resources seamlessly through the grid portal and develop many potential parallel or distributed power system applications.

Although grid computing has been successfully applied in many areas, there is only one paper that has discussed its application in power systems [40], and this paper gave an introduction to grid computing technology and discussed various possible power system applications based on it. However, while the research pointed out the wide application prospects of grid computing technology in power systems, it did not explain how to construct the grid system for power systems in a conventional way, and it also did not study issues related to implementation techniques for combining grid computing with power system applications. Such issues need to be studied in advance when constructing the grid system for power systems because they may affect the performance of power system applications.

2.2.3 Cloud Computing based Data Center

At the same time of the proposal of smart grid, a new technological trend called “cloud computing” emerged and rapidly developed in the IT industry. Cloud computing is not a completely new technology but an integration of multiple technologies, including automatic computing, grid/distributed computing, hardware virtualization, web service, and utility computing. Unlike other traditional computing models, cloud computing aims to deliver various kinds of IT resources (computing powers, storage devices, development platforms, and software) as services through the Internet to end users, while end users can acquire the resources as they need on a pay-per-use basis. According to the definition given by Foster [41], cloud computing is a large-scale distributed computing model which can form a virtualized, dynamical scalable resource pool to deliver services to users.

The core of cloud computing is the data centre, which is a facility to house the computer systems, telecommunication systems, and storage systems that provide Internet-scale, elastic on-demand services to users. Currently, the major cloud service providers (such as Amazon and Google) have established data centres backed up with thousands to millions of computational equipment. Integrating a cloud data centre into a smart grid can provide a platform to collect, storage, and analyze mass data.

2.3 Application of Computational Intelligence in Power Systems

2.3.1 Various Techniques of Computational Intelligence

Increased interconnection and load of the power system, along with deregulation and environmental concerns, has brought new challenges for modern power system operation, planning, and control. As a study of adaptive mechanisms that enable or facilitate intelligent behaviour in a complex and changing environment, computational intelligence (CI) [42] is a novel and modern tool for solving complex problems that are difficult to solve by conventional techniques. Due to time requirements of direct and modal analysis methods, researchers are inclined to apply computational intelligence methods, which are promising for real time applications. Developing solutions with CI techniques could offer the following advantages:

a) Researchers can learn from past experience;
b) CI techniques are fault tolerant in the sense that they are able to handle noisy and incomplete data/information, i.e., uncertainties, to make the systems more robust;
c) CI techniques are able to deal with nonlinear problems;
d) CI techniques can perform prediction and generalization at high speed with training, and the development time is much shorter than when utilizing the traditional approaches.

Common among them, as reported in most studies [42-44] are artificial neural networks (ANN), evolutionary computation (EC), fuzzy system (FS), and swarm intelligence (SI). Each of these CI techniques has its origins in biological systems. Neural networks model the biological network; EC models the natural evolution (including genetic and behavioural evolutions); FS originates from studies of how organisms interact with their environment; and SI models the social behaviour of organisms living in swarms or colonies (such as particle swarm optimization algorithm [45]).

2.3.2 Potential Research Area in Power Systems Using Computational Intelligence

There are several problems in modern power systems that cannot be solved with conventional approaches as these methods are based on several requirements which may not be true all the time. In those situations, computational intelligence techniques are the only choice; however, these techniques are not limited to these applications. The following areas of power systems could utilize the application of evolutionary computation.

- Power system operation, which includes economic dispatch, unit commitment, hydro-thermal coordination, scheduling, congestion management, power flow, and state estimation.
- Power system planning, which includes generation expansion planning, transmission expansion planning, and reactive power planning.
- Power system control, which includes voltage control, load frequency control, power flow control, stability control, and dynamic security assessment.
- Distribution system applications, which include demand side management, demand response, network reconfiguration, and smart grid operation and control.
- Forecasting applications, which include short-term load forecasting, long-term load forecasting, electricity market forecasting, wind power forecasting, and solar power forecasting.

2.4 Summary

This chapter serves as a literature review of those techniques pertinent to the remainder of this thesis. It includes a brief description and discussion of wind power energy in emerging power systems, the advanced computing techniques for modern power systems, and the application of computational intelligence methods in power systems.

Chapter 3

Advanced Computing Platforms for Modern Power Systems

3.1 Introduction

In modern times, the trend of high penetration of renewable energy sources, deregulation, the emergence of super scale interconnected systems, and the distribution generations make power systems increasingly complex, which poses some challenges for the computing platforms of modern power systems.

Following the review of advanced computing techniques discussed in Chapter 2, this chapter will present a computational grid platform for distributed heterogeneous power systems and a cloud computing based information infrastructure for future power systems. Grid computing and cloud computing techniques have both received much attention from power engineers and researchers over the past few years. They can provide high performance computing ability and communicate among distributed and heterogeneous resources in power systems.

3.2 Grid Computing Technique

3.2.1 Grid Computing Introduction

Grid Computing (GC) is a novel intelligent computing technology, for sharing and managing computerized distributed and heterogeneous resources (CPU, physical memory, storage, network, etc.) for some specific tasks [46]. These tasks can be simulation and forecasting of financial market, or some scientific researches such as earthquake prediction, or information integration for some special business demands. With the continuously improved GC standards, such as Web Services, OGSA, WSRF introduced in [47] , GC can be used for a wide range of tasks from pure sharing of computing power to providing powerful management and coordination capabilities among diverse distributed heterogeneous resources, thereby to deliver nontrivial qualities of service (i.e. application demands). According to the functions of GC applications, it can be divided roughly into three main domains: Computational Grid, Data Grid and Knowledge Grid.

One of the keywords in GC is “virtualization”. It means by seamless integration of geographically distributed and heterogeneous systems [47], to provide grid users with a transparent environment. From user's prospective, they just need to submit their service requests as long as entering the "virtual grid". Diverse organizations which comprise the integrated "virtual grid" are called "Virtual Organizations" (VOs) [48]. In the case of power systems, future power grids will consist of many participants including generator operators, generator maintenance providers, generation aggregators, transmission network operators, distribution network operators, electrical load managers, energy market makers, energy supply companies, metering companies, energy customers, regulators, and governments [49]. How to manage and coordinate these organizations is a key issue for future power systems.

3.2.2 Grid Computing System Architecture

During the design and realization of a GC-based system (i.e. grid system), many aspects need to be considered. GC architecture can be visualized as a layered architecture according to [48], including application layer, grid middleware layer and grid resource layer. The topmost layer consists of many grid applications and the APIs from users' perspectives. The middleware layer includes software and tools utilized for grid environment implementation, such as Globus Toolkit [50], gLite [51] and BONIC [52]. The third layer covers all the resources available to the grid, such as storage, processing capabilities (CPUs), and other application-specific machines. Ali et al. presented a suchlike architecture in [53, 54], named RSA-Grid for power system reliability and security assessment purpose. The architecture is shown in Fig. 3-1.

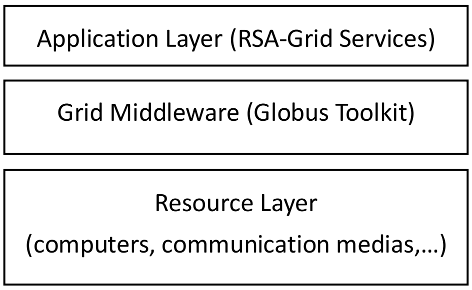


Fig. 3-1. A Typical GC Layered Architecture, [53]

GC has been well confirmed for its supercomputing capability in power systems.

However, the big difference of GC system to other distributed system is the VO characteristic and consequent management mechanism. With these mechanisms, the geographically distributed and structurally heterogeneous resources could coexist and collaboratively contribute to the development of the future “Smart Grid”, which may be a new path for power systems research.

3.3 Opportunity and Challenges of Grid Computing in Power Systems

The research and application of GC in power systems have been launched since 2004. However, because of the inchoate standards and technologies, there are still challenges as well as opportunities. This section gives the analysis and discussion on the opportunities and challenges of GC in various application fields of modern power systems.

3.3.1 Distributed Monitoring and Control

From the prospect paper of Irving et al in [55], the distributed monitoring and control had been put forward as one of the potential benefits of GC for future power systems. Along with the increasing small-scale generators come from discrete locations, even be characteristic of different structures (i.e. traditional generators mixed with new renewable-based ones), the first step of building a “Smart Grid” is to achieve real-time monitoring and control of the whole power grid. Irving et al and Huang et al described respectively a GC-based platform for distributed monitoring and control of power systems in 2006 [56, 57]. The former one, established based on the GRIDCC project [58], presented a PSSimulator used for initial Grid environment experiment, which virtually connected power generators instances in Italy, Switzerland and UK. With a resource broker middleware and the simplified Java Program, the real-time operation and communication costs were tested to demonstrate the feasibility of this platform. Whereas the latter one proposed by Huang et al. focused on the virtual database technology, for accessing heterogeneous shared data. There should be more profound research on how to complete complex management of these power grids, which will be a challenge for future power systems.

3.3.2 *Grid Computing vs Other Computing Techniques*

In this section, GC is compared and contrasted with other computing techniques, including High Performance Computing, Parallel Computing and Distributed Computing methods. And it also gives insights into the essential characteristics of all.

1) Grid Computing Vs. High Performance Computing.

High Performance Computing (HPC) technology uses supercomputers and/or high-performance computer clusters to solve advanced computation problems, and substantial research and development work have been conducted in power systems analysis [59]. However, for many experimental science fields, scientific progress and quality of research are strongly linked to computing throughput. Because for many research and engineering projects, the quality of research or projects is heavily dependent upon the quantity of computing cycles available. High Throughout Computing (HTC) focuses on this kind of situation. There are many differences between HPC and HTC. In short, HPC environments, whose performance is often measured by Floating Point Operations Per Second (FLOPS), deliver a tremendous amount of compute power over a short period of time; while HTC deals with how to deliver large amounts of processing capacity over long periods of time. GC has just emerged as an important field synonymous to HTC. It is the revolutionary technology built upon HTC concept, which can better complete lots of tasks coming from different locations and can provide some communication standards or protocols for the overall operation of the system.

2) Grid Computing Vs. Parallel Computing and Distributed Computing.

Parallel Computing, or Distributed Computing belongs to the field of computer science that studies compute-intensive problems and distributed systems. It has been utilized in many fields of power systems as well, [60-63]. A distributed system is static and has no concept of "virtualization". It focuses on information sharing often using the Client/Server (C/S) model. GC is the next generation of Distributed Computing. Distributed Computing technologies enable information sharing within a single organization, whereas GC technologies enable resource sharing among VOs. A grid system could support resource discovery and monitoring on a global scale. Therefore, it could work together with Parallel/Distributed Computing and supercomputing technologies, to provide an efficient solution of competition and cooperation among geographically distributed and structurally heterogeneous participants.

3.3.3 Fundamental Computation

Many problems in complicated power systems demand high performance computing. While historically HPC or Parallel Computing could provide a better solution compared to single computer processing unit, a grid system can tackle the problems as well, and even better. There are many fundamental computation problems for analysis and control of a power system. Being service-oriented, GC can be used to provide such fundamental computation services for more involved and synthetic applications. One example is Monte Carlo simulation method, which is perceived as computationally intensive but naturally parallel and has been widely used in many areas of scientific research.

In most of the optimization issues in power systems, a realistic formulation leads to highly nonlinear relationships, discrete and integer variables, and many other ill-behaved characteristics of the mathematical models. Some of these problems are combinatorial optimization problems with exponential increase computing power requirement, or involve thousands of constraints. Most of these problems are adequate for parallel processing pattern, which decomposes problems to raise the computational efficiency. Recently, many heuristic search optimization techniques have been proposed to solve some of these problems. Evolutionary Algorithms (EAs), for instance, are robust and flexible in handling nonlinearities and discrete variables, and have been successfully applied in different power system optimization problems such as load flow calculation [64], optimal reactive power flow (ORPF) [60,65-68], transient stability constrained OPF [69], damping control design [70], TNEP [71], unit commitment [72], unit maintenance scheduling [73], and power market analysis [74-77]. These algorithms can be implemented on GC platform and provide efficient tools for power system operation and planning.

3.3.4 Power System Information Grid

The trend of future power systems is large-scale with dispersed and diverse small generators. Distributed power system leads to distributed management. With the emergence of smart sensors which will offer real-time filtered data from electrical devices, GC can provide a relatively low-cost information integration service for the whole power system. All the distributed data could form a huge virtual database, with near drop-in storage spaces and convenient data backup methods. It will accelerate the realization of the Smart Grid.

3.3.5 Reliability & Security Analysis

Maintaining system security is the most important task for power system operators. There are large amounts of constant dynamic instability factors which require comprehensive analysis and control [78]. Many of these analyses must build diverse mathematical models which are often too complicated to be solved effectively. Because of the dynamic feature of power systems, these calculations still need to be completed within a given time period, sometimes even just few seconds. GC provides an effective approach in solving these computational-intensive and time-intensive problems. Several experiments had been unfolded with preferable efforts [54,79]. It is obviously that GC can provide better computational service superior to distributed/parallel computing. Thanks to GC's robustness and expandability, the turbulences or human errors could be reduced by self-healing and fault-tolerance technologies.

3.3.6 Power Market Grid

GC is created for distributed and heterogeneous resources. This feature is well suited for power market analysis. With the progress in the power market reform, the deregulation and autonomy of power industries derives a large number of heterogeneous resources. A competitive power market involves energy supplier, energy customer, market objects (energy, energy transmission rights and assistant services), market price and market regulations, etc. Data is the kernel of power market, as the market analysis, forecasting, decision-making support problems all demand for large amounts of data. Via GC technology, combined with other smart method such as data mining, a power market grid will be established meaningfully.

3.3.7 System Planning & Scheduling

Grid resource scheduling is a component of GC technology, which plays an important role in the overall performance of an application running on the grid. Similarly, for electrical power grids, prominent electrical planning and scheduling determines the efficiency and security of power systems. Considering the particularity of electrical energy (i.e. it cannot be stored), GC could provide real-time access to disperse resources and serve power systems with very fast forecast results, thus balances the difference between generation and load demand [54, 80]. In [81] and [82], two methods for electrical load balancing and hydro-thermal scheduling were shown, independently and GC based respectively. The authors of [81] claimed that GC technology was suitable for real-time data transmission and real-time load balancing process among power plants; while the authors of [82] utilized GC to build the scheduling model, which adopted Monte Carlo method to produce system stochastic parameters.

Power system economic dispatch is the method of determining the most efficient, low-cost and reliable operation of a power system by dispatching the available electricity generation resources to supply the load in the system. Its object is to minimize the total cost of generation subject to operational constraints of the available generation resources. Load balancing and scheduling problem could not be simply regarded as a computational-intensive operation, but involves many smaller tasks, such as data exchange among different domains which is related to transmission operations in the large power network. GC will continue to provide effective computational means for solving these dispatch problems.

3.3.8 Grid Computing & Smart Grid

Smart Grid is a modernized electrical power grid that utilizes new equipment, information and communication, control and automation technologies to respond to our 21st century demand of electricity. It must need smart technologies, especially on the computing layer. GC could be seen as a smart and intelligence computing method due to its high efficiency, low cost and easy accessibility.

Due to the complexity of power systems, the possible GC applications are wide in nature and hard to be merged. They, however, are mainly based on two aspects: virtual data storage and inexpensive parallel/distributed computing, with more standardized architecture and communication protocols.

3.4 A Computational Grid Platform for Distributed Heterogeneous Power Systems

3.4.1 System Model

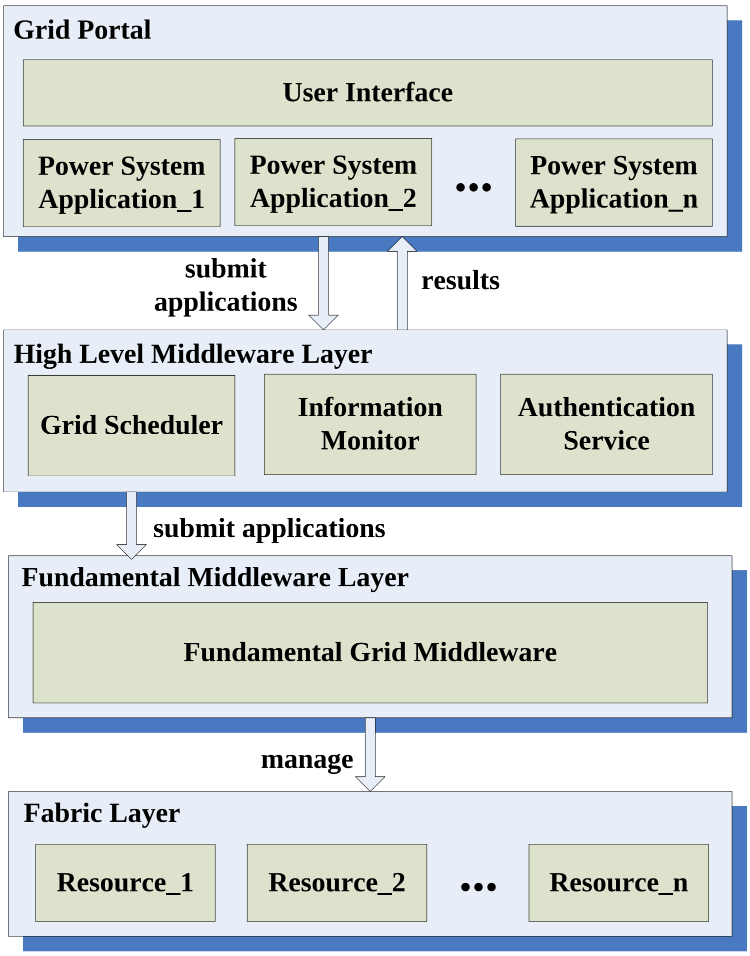


Fig. 3-2. Layered model of computational grid platform

The whole computational grid platform is designed as a layered model (Fig. 3-2). Based on this model, power scientists and engineers can access the distributed resources to perform computational tasks through a scientifically designed grid portal. Middleware components handle task execution, data accessing and scheduling. Functions of each layer are introduced in this section and each layer can be connected via Internet to form the grid platform, as shown in Fig. 3-3.

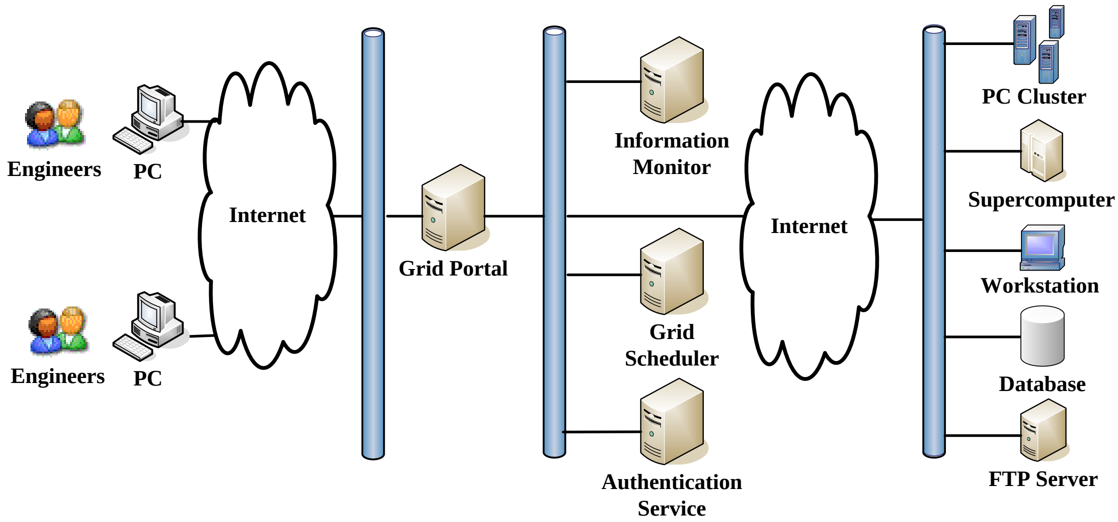


Fig. 3-3. Computational grid platform in Internet environment

3.4.2 *Types of Grid Systems and Problem Description*

Based on the general architecture of the grid described in Section 3.4.1, there are mainly two types of systems that can be constructed for power system applications.

1) Small Scale Scientific-Purpose Grid System

The first is the small scale scientific–purpose system that uses grid computing technology to connect multiple computational resources for simulations. This type of system is based on existing Internet infrastructures and, therefore, it is low cost and very easy to construct. If the resources have a public IP address, only grid components need to be installed to construct the system. However, bandwidth of the existing Internet infrastructure may affect the performance of the system. Besides, since the system scale is relatively small, variation of availability of a particular resource might also have significant effect on performance of the whole system. In this system, the number of users is often limited, and there are few concurrently running applications at a given time. Thus, individual applications submitted to the system often do not need to compete for resources with other applications. In this situation, the main problem is how the dynamic grid environment affects power system applications.

2) Large Scale Industrial or Commercial Grid System

Another type of system is the large scale industrial or commercial grid system. This type of system often needs significant funding for constructing private high bandwidth network and using grid technology to harness a large number of widely distributed resources based on a private network to facilitate reliable scientific, industrial and commercial applications. In this type of system, bandwidth of the whole system is often very high, and there often exist many users and a large number of concurrently running applications that compete for resources. Thus the main problem is how to schedule the concurrent power system applications effectively.

3.4.3 Study Methodology

For addressing the problems described in Section 3.4.2, this research designs a series of experiments. For the small scale scientific-purpose grid system, experiments are designed to simulate the dynamic grid environment and observe its impact on power system application. For large scale industrial systems, an economic-based framework is proposed to schedule concurrently running power system applications among heterogeneous resources.

1) Dynamic Factors Simulation for Small Scale Grid Systems

There are two major factors causing the dynamics of Grid system: Internet bandwidth and variation of resource availability. In this research, these two factors are simulated to observe their effect on the performance of compute-intensive power system applications. In the test bench, four resources in Table 3-1 are created to represent the small scale grid system, and different bandwidths and availability levels are assigned to each resource.

Table 3-1 Resource Definition of Small Scale Grid System

Group	Resource_1	Resource_2	Resource_3	Resource_4	Code
Group 1	H1, RA1	H2, RA1	H3,RA1	H4,RA2	G1_1
	H1, RA1	H2, RA1	H3,RA1	M2,RA3	G1_2
	H1, RA2	H3, RA3	H4, RA2	L1, RA3	G1_3
Group 2	M1, RA1	M2, RA1	M3, RA1	M4, RA2	G2_1
	H4, RA1	H5, RA1	M4, RA1	M5, RA3	G2_2
	M1, RA1	M2, RA1	L2, RA1	L5, RA2	G2_3
	H2, RA1	H3, RA1	L2, RA2	L3, RA3	G2_4
Group 3	L1, RA1	L2, RA1	L3, RA1	L4, RA2	G3_1
	L1, RA1	L2, RA1	L3, RA1	M3, RA3	G3_2
	L2, RA1	L3, RA1	L4, RA2	H1, RA3	G3_3

Firstly, bandwidth of the existing Internet infrastructure is simulated. The procedure is designed as follows. An Internet transference speed measurement tool [83] is used to sample the bandwidth data of multiple sites around the world. This tool can measure the transference speed between different servers around the world and the user's host. Sampled bandwidths are divided into three intervals, low, moderate and high, denoted as L, M and H, respectively. Then some typical sites from each speed interval are chosen and assigned to each resource. Changes in the speed of each site in one hour are measured. The target of this step is to estimate the floating scope of each site during a certain time period. In my research, from the sampled Internet data, bandwidth intervals are estimated as high bandwidth (more than 0.8 MB/s); moderate bandwidth (from 0.2 MB/s to 0.8 MB/s); and low bandwidth (less than 0.2 MB/s). Five typical sites are selected for each interval, i.e. L1-L5, M1-M5 and H1-H5.

Secondly, resource availability is simulated. In the simulation, majority of resources remain available while other resources might become unavailable at uncertain times. Three levels of availability of resources are defined.

a) The first level (named as RA1) is where resources are dedicated and are always available during the experiments;

b) In the second level (named as RA2), resources become unavailable at uncertain times during the experiment and then become available again after 10 minutes. This strategy can simulate temporary shutdowns or temporary network failures;

c) The third level (named as RA3) is where resources become unavailable at uncertain times during one experiment and then remain unavailable.

Finally, a certain bandwidth and availability level can be assigned to each resource. Different combinations can thus be defined, as shown in Table 3-1. The combinations can be divided into 10 cases of three groups, with each group reflecting a performance level of the whole system. For example, for Group 1, bandwidth of the whole system is high. Resources in G1_1 and G1_2 cases of this group are stable generally but resources in G1_3 are relatively unstable.

2) Improved Economic-based Scheduling Framework for Large Scale Grid Systems

Popular existing grid computing schedulers use various scheduling algorithms to optimize execution time of applications. However, R. Buyya [84] (a famous expert of grid computing whose main contributions are developing the well-known grid simulator 'GridSim' and grid scheduler Nimrod-G, and who first proposed the idea of economic-based grid scheduling mechanism) pointed out that these methods are inadequate when applied in practical industrial or commercial grid systems. This is because in large scale industrial or commercial grid systems, the cost of accessing the resources cannot be neglected, and such cost must be considered in scheduling algorithms [85]. Buyya thus proposed an economic-based scheduling model which considers both execution time requirements and the cost of resource accessing [86,87]. The model consists of different scheduling strategies to optimize execution time and cost separately, and the grid scheduler can select suitable strategies according to requirements of different applications.

However, Buyya's framework assumes all applications have same priorities. In power systems, a notable characteristic is different applications might have different urgency degrees. For example, some applications belong to periodically executed engineering tasks (such as real time power dispatching and stability analysis) that must be completed within a certain time (say 15 minutes); these tasks have higher priority. Other applications, such as long-term system planning problems, do not have such strong time limits and thus have lower priority. When the grid system is established for power systems, this characteristic must be considered in the scheduling algorithm.

Based on Buyya's framework, this research proposes an improved economic-based framework for scheduling power system applications, as shown in Table 3-2. There are a total of four strategies in the framework. The cost-optimization strategy aims to minimize the resource accessing cost of the application, while keeping the execution time within the deadline; the time-optimization strategy aims to minimize the execution time of the application, while keeping the resource accessing cost within the budget; cost-time-optimization minimizes the resource accessing cost, and for resources having the same cost, the strategy uses time-optimization among resources. These three scheduling strategies have already been implemented in Buyya's framework but in the framework of this research, they are improved to incorporate priorities of different applications. Execution of jobs with higher priority is ensured on a priority basis. At last, a new strategy called feasible job optimization is proposed to maximize the count of feasible jobs. Here, a job is feasible if it is assigned to a suitable resource after the scheduling procedure.

Table 3-2. Scheduling Strategies of Proposed Framework

Strategy	Execution Time	Execution Cost	Feasible Job Count
cost-optimization	Limited by deadline	Minimize within budget	---
cost-time- optimization	Minimize when possible	Minimize within budget	---
time-optimization	Minimize within deadline	Limited by budget	---
feasible job optimization	Limited by deadline	Limited by budget	Maximize

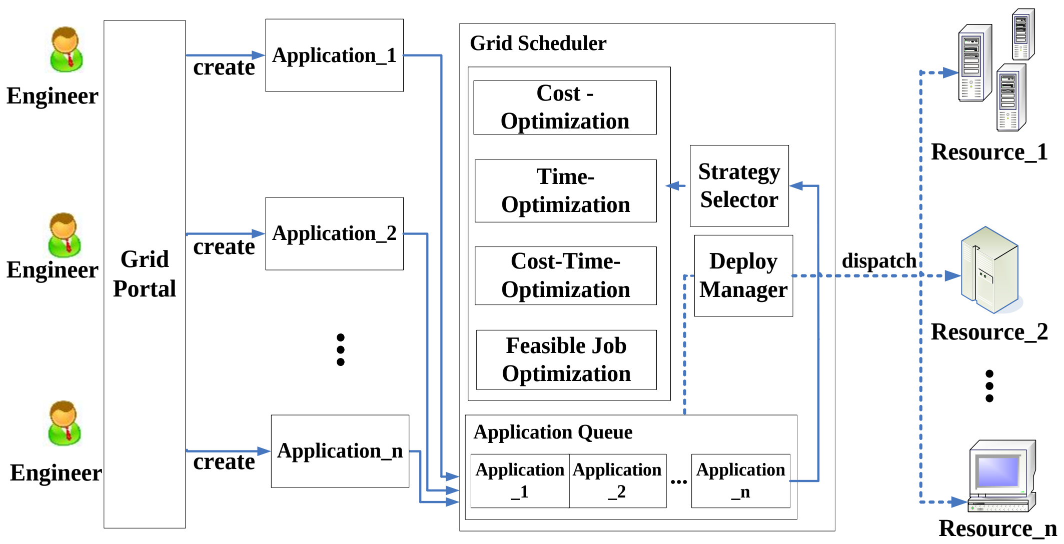


Fig. 3-4. Working mechanism of the grid scheduler

The working mechanism of the economic-based grid scheduler is shown in Fig. 3-4. Power system applications submitted by the users are stored in an application queue. Each application comprises multiple jobs. For each application in the queue, the scheduler chooses a suitable strategy, according to its requirements, and schedules the jobs of that application. The scheduling strategy finally selects a resource for that job, and the deploy manager dispatches the job to the resource.

The four scheduling algorithms follow similar procedures, as shown in Fig. 3-5, while there are some distinctions in resource sorting:

a) The cost-optimization strategy aims to minimize the cost of the application while keeping the execution time within the deadline. In the beginning, the cost-optimization algorithm forms a resource pool by identifying resources which can process the job within the deadline; and then it sorts resources by increasing order of the accessing cost and gets the resources sequentially to perform the remaining scheduling procedures.

b) The time-optimization strategy aims to minimize the application execution time while keeping the cost within the budget. In the beginning it forms a resource pool by identifying the resources adequate for the job; and then sorts resources from fast to slow.

c) The cost-time-optimization strategy is similar to the cost-optimization strategy, but if multiple resources having the same accessing cost, it performs time-optimization scheduling among the resource.

d) The basic idea of the feasible job optimization is that it always tries to reserve faster resources for later jobs. In this way, it does not aim to minimize the execution time or cost of the jobs, but tries to reserve better resources to the later jobs, so as to make more jobs feasible after the whole scheduling procedure.

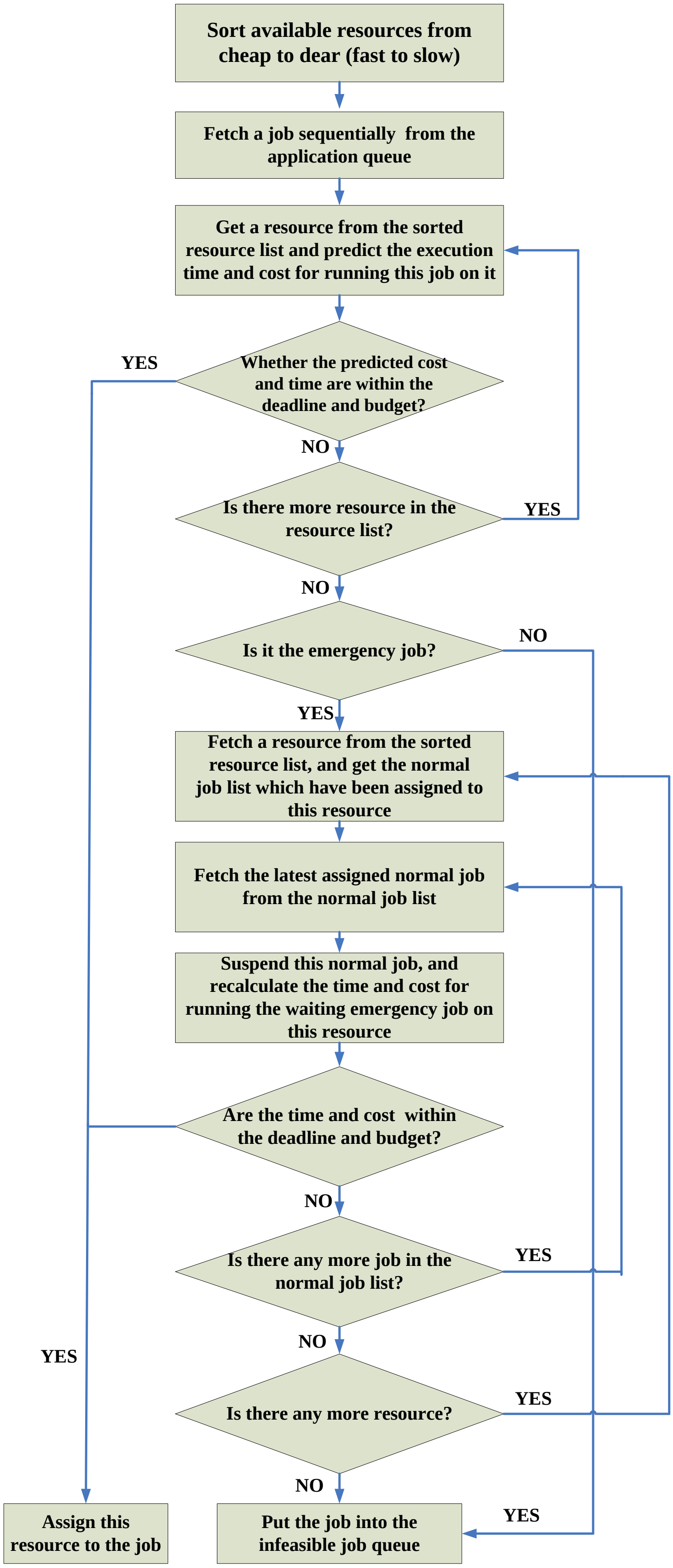


Fig. 3-5. Procedure of the scheduling algorithm

3.4.4 Grid System Simulation

In the simulation, this research first demonstrate how to make use of state-of-the-art open source grid computing products to implement the computational grid system described in Section 3.4.1, in a Linux PC cluster. This implementation approach can be directly applied for constructing a practical grid system. Secondly, the implemented system is directly used to study the impact of dynamic factors in power system applications in the small scale grid system. Thirdly, the GridSim 2.0 [88] is utilized on a single PC to simulate the large scale resource pool and model the concurrent power system applications, in order to evaluate the efficiency of the economic-based scheduling framework.

The Grid system described in Section 3.4.1 is implemented in a 30-node Linux PC cluster. Four computational resources are created, and the mature open source grid components are used. Globus Toolkit 4 [89] is selected as the fundamental middleware; GridWay [90] is used as the grid scheduler; MyProxy [91] is used to provide the authentication service; PBS [92] is used to manage each resource and Ganglia [93] is used to collect state information of the resources. All grid products are open sourced and can be downloaded from the corresponding websites. Thus the above implementation approach can be directly followed to construct a practical grid system.

Table 3-3. Grid System Implementation on 30-Node Linux PCs

Node ID	Role	Configuration
00	Client Node	--
01	Grid Portal Grid Scheduler	Globus Grid Portal Program GridWay
02	Information Monitor	Globus
03	Authentication Service	Globus MyProxy
04-10	Resource_1	Globus PBS Ganglia Grid simulator
11-16	Resource_2	Globus PBS Ganglia Grid simulator
17-23	Resource_3	Globus PBS Ganglia Grid simulator
24-30	Resource_4	Globus PBS Ganglia Grid simulator

In addition to this, in order to simulate the two dynamic factors introduced above, a grid simulator developed by this research is installed in each resource to control availability of the resource and simulate the assigned bandwidth. The simulator is a set of Linux script programs running in each resource. For simulating the bandwidth, it produces varying bandwidth and calculates the transference delay time needed by transferring the result files from the resource to the scheduler, and then holds the result files returned by PBS for such period of time; for simulating resource availability, it shuts down the PBS at a random time, and then restarts the PBS after a period of time, according to the stability level of the resource. The roles of each machine are shown in Table 3-3.

3.4.5 Impact of Dynamic Factors in Small Scale Grid System

The Monte-Carlo probabilistic power flow analysis in the IEEE 118 bus system [94] is used as the test problem. The total sampling times in the range of 1,000 to 100,000 is studied as shown in Table 3-4. Each probabilistic power flow calculation is divided into several jobs, and each job performs a certain number of sampling. These divided jobs can be executed in parallel in the grid environment. Four cases using different resources groups in Table 3-1 (Groups 1 to 3) are performed as follows.

Table 3-4. Comparison of Application Cases between Sequential Computing and Grid Computing

Total sampling times	**Sampling times per job**	**Total number of jobs**	**Sequential execution time (s)**	**Grid execution time without Internet delay (s)**	**Speed up**
1,000	100	10	19.7	112	0.18
2,000	100	20	45.2	138	0.33
5,000	200	25	167.4	147	1.14
8,000	200	40	373.6	213	1.75
10,000	200	50	537.0	203	2.65
15,000	300	50	1100.2	277	3.97
20,000	400	50	1938.1	287	6.75
30,000	400	75	4523.3	369	12.26
50,000	400	50	11359.2	589	19.29
80,000	500	80	28353.7	856	33.12
100,000	500	100	40913.5	890	45.97

Case 1 is the system which has an adequately large bandwidth and all resources of which are stable. This can be considered as the best situation. Case 2 is where the whole grid system has good network performance in general (Group 1). Case 3 has moderate network performance (Group 2). Case 4 is where the whole network has low transference speed in general (Group 3).

In Case 1, all resources remain reliable, and the Internet latency can be ignored because of the adequately large bandwidth. The efficiency of running the probabilistic power flow on the grid system and on a single PC sequentially are compared and shown in Table 3-4. It can be seen that as the problem scale increases, the grid computing platform can solve the problem much faster than the single PC. When the sampling count is 100,000, the single PC uses 40,913.5s to solve the problem while the grid computing platform needs only 890 seconds. The speed-up ratio is around 46.

In Case 2, the whole system has high network performance (G1_1, G1_2, and G1_3 in Table 3-1). It can be observed that when the whole network has high bandwidth, performance of the system is close to the best situation, even when speed of some resources is slow (curve G1_3 of Fig. 3-6). When considering variations in availability of resources, it suggests that when simulation time is small and proportion of unstable resources is also small (curves G1_1 and G1_2 in Fig. 3-7), the resource availability variation has little effect only.

In Case 3, the whole system performance of Group 2 is moderate (G2_1, G2_2, G2_3, and G2_4 in Table 3-1). The simulation result is shown in Fig. 3-8 and Fig. 3-9. According to curves G2_1 and G2_2, when the network performance is moderate and there are few low-speed resources, computation time can be considered as close to the best situation. Comparing Fig. 3-7 and Fig. 3-9, it can be seen that when network performance is moderate in general, variation in computation time caused by unstable resources is larger than in Case 2, especially when there are some low-speed resources.

In Case 4, simulation results of Group 3 are reported in Fig. 3-10 and Fig. 3-11. Fig. 3-10 shows the performance of the system when the system is in a bad condition (G3_1, G3_2, G3_3 in Table 3-1). All the four combinations of Group 3 spent much more time to solve the problem, compared with the ideal result, especially curves G3_1 and G3_2. In G3_1 and G3_2, all resources have very low transference speed. When considering the resource availability variation, the result reveals that computation time curves are far away from the best situation.

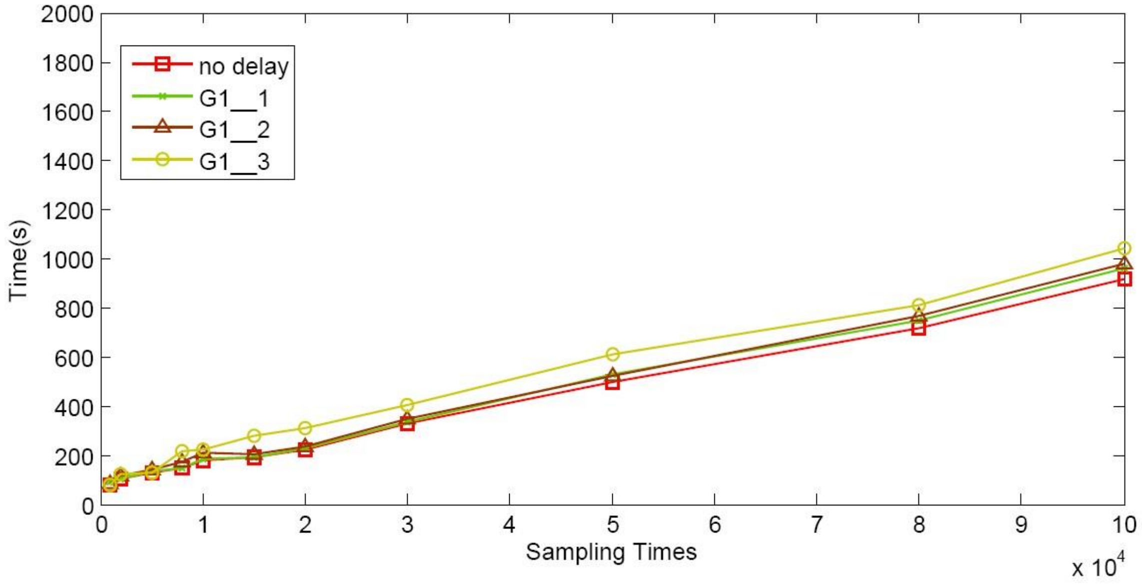


Fig. 3-6. Simulation result of Group 1 when all resources keep stable

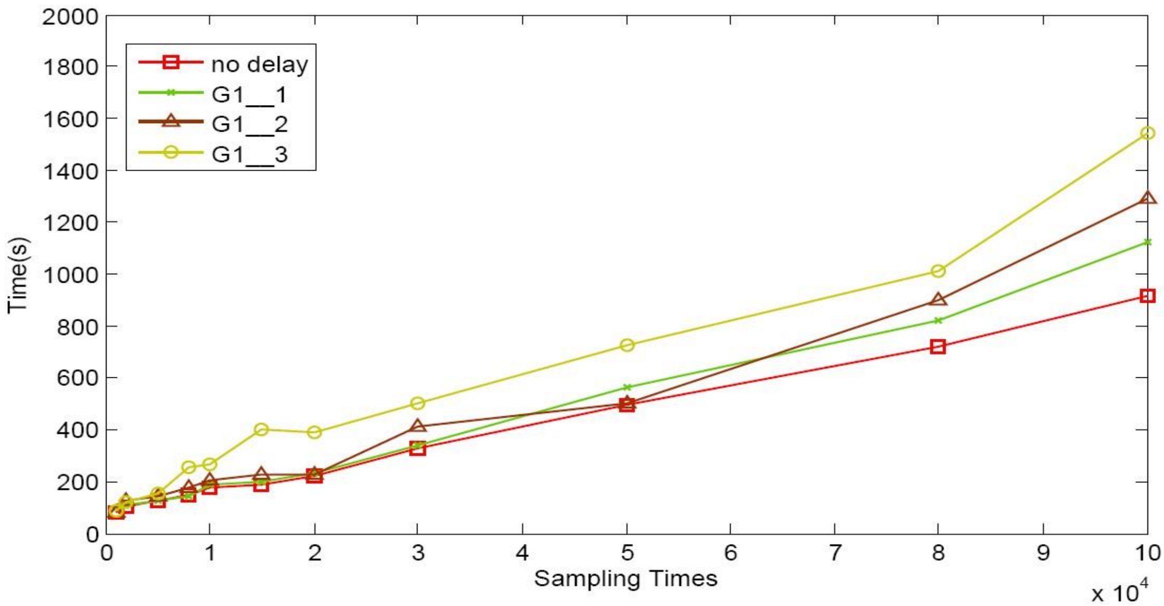


Fig. 3-7. Simulation result of Group 1 considering the availability variations of resources

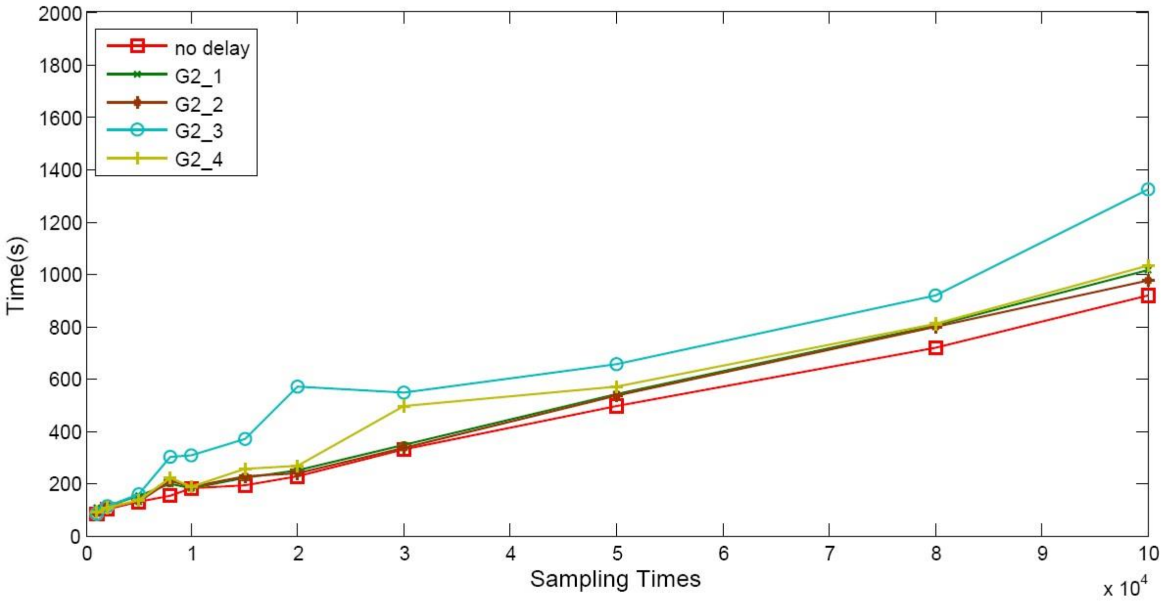


Fig. 3-8. Simulation result of Group 2 when all resources keep stable

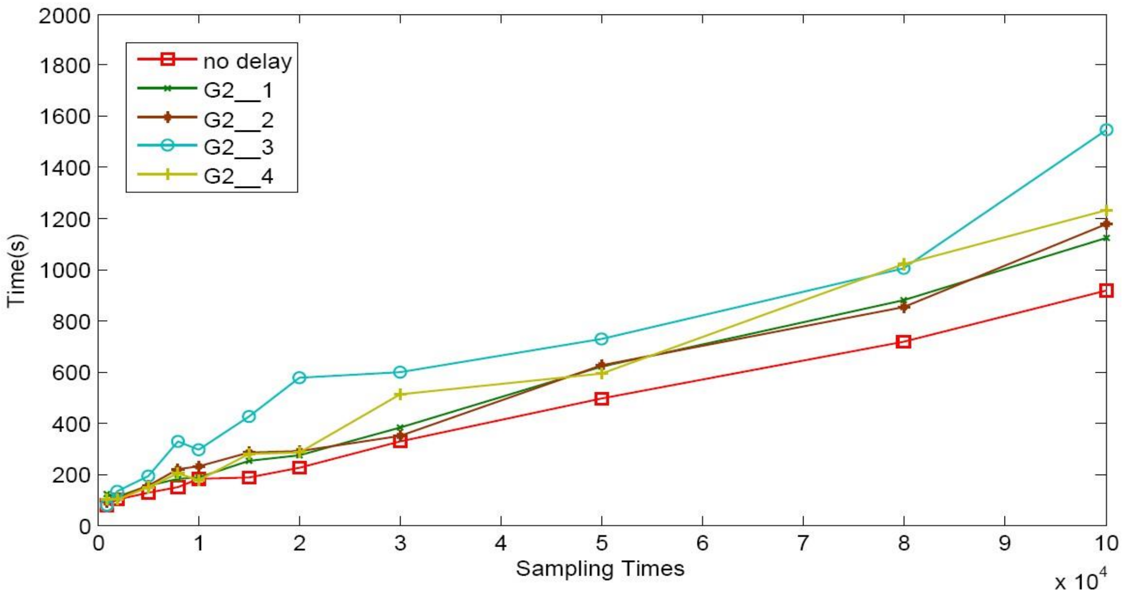


Fig. 3-9. Simulation result of Group 2 considering the availability variations of resources

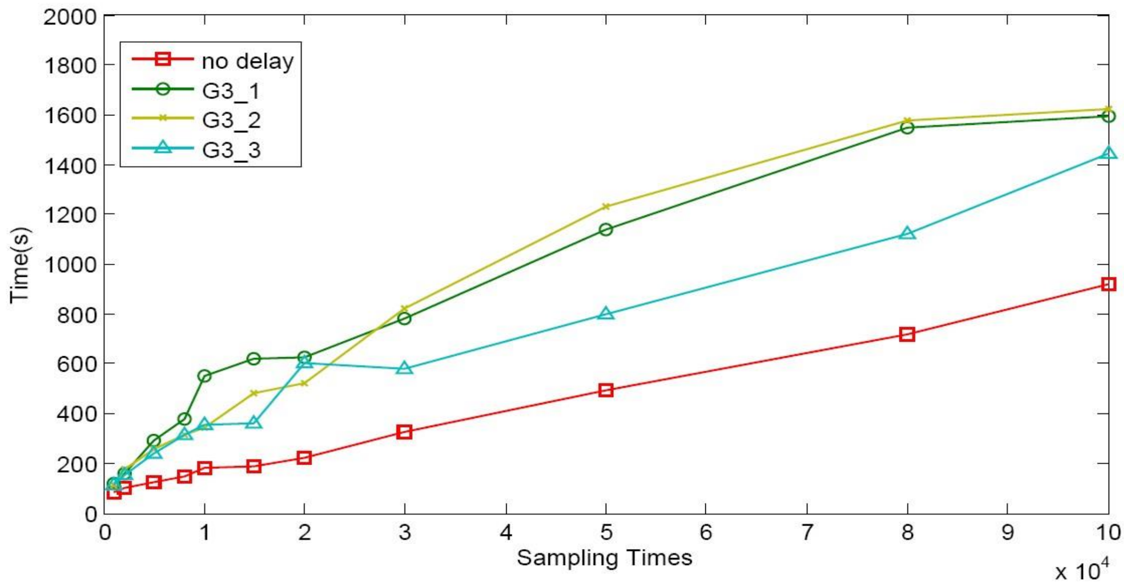


Fig. 3-10. Simulation result of Group 3 when all resources keep available

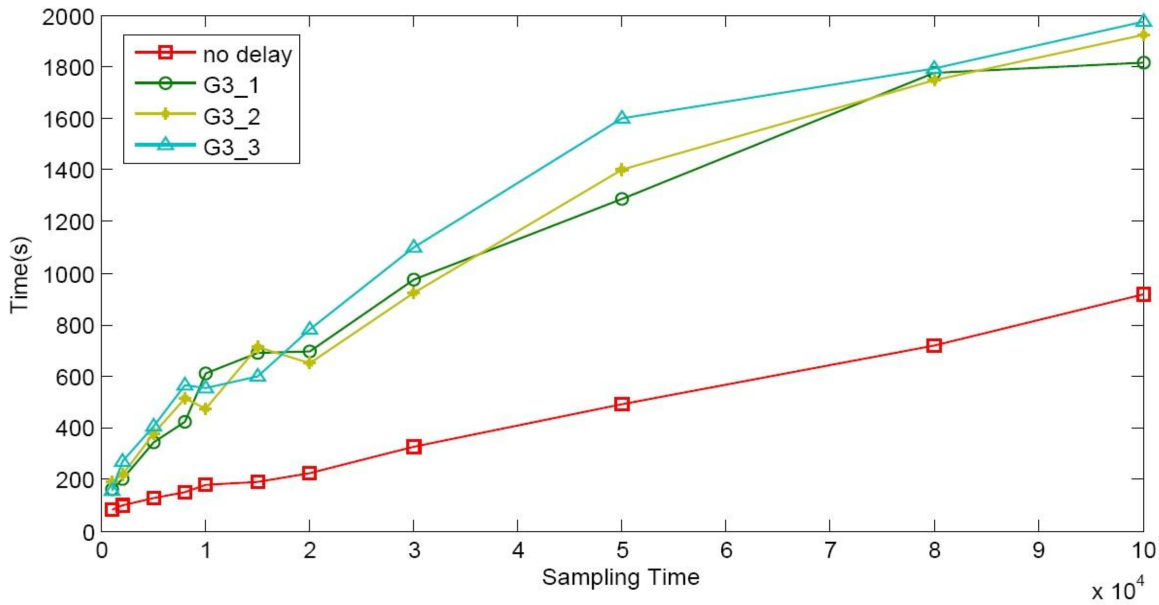


Fig. 3-11. Simulation Result of Group 3 considering the availability variations of resources

From the above simulation, some useful guidelines can be drawn to guide the construction of the scientific-purpose small scale grid computing platform for power system.

1) When the system has high bandwidth in general, even if there are a few unstable resources, the system performance is not affected much;

2) When the whole system has moderate bandwidth in general, if resources are stable, then the system performance is close to the ideal performance; otherwise, the system

performance is seriously affected;

3) When the system has low bandwidth, its efficiency is very low in solving compute-intensive problems; and

4) The impact of unstable resources on grid system increases with increase in the problem scale.

3.4.6 Evaluating the Improved Economic-based Scheduling Framework in Large-Scale Grid Systems

World Wide Grid (WWG) is a widely used test bed for grid computing and peer-to-peer computing research. WWG includes multiple heterogeneous computational resources around the world [95]. In this research, resources of WWG are created in GridSim 2.0.

For simulating concurrent power system applications, 15 Monte-Carlo probabilistic power flow calculations are assumed. Although the probabilistic power flow analysis is used here as an example, the scheduling framework can be used for various kinds of applications. The power flow calculations are divided into two classes: normal applications and emergency applications. Each application consists of several divided jobs and average running time as shown in Table 3-5. There are totally 200 jobs, 43 jobs of which are belong to 4 emergency applications (denoted as E) while 157 jobs are belong to 11 normal applications (denoted as N).

In the following experiments, different deadlines and budgets are assigned to the applications. The unit of the deadline is grid second(s) and the unit of the budget is grid dollar (G$) [86]. In the practical grid system, grid second and grid dollar can easily be mapped to time and money under real situations. In my study, the time required to process all jobs of an application in parallel, given the fastest resource is the highest priority, is defined as T_{fast}; the cost of processing all jobs of an application, given the cheapest resource is the highest priority, is defined as C_{cheap}. The economic-based grid scheduling framework is then evaluated in three aspects: feasible jobs, application execution time and application cost.

Table 3-5. Configuration of Concurrent Monte-Carlo Probabilistic Power Flow Analysis

Application Name	**Number of Jobs**	**Average Running Time of Each Job (s)**	**Type**
App1	15	25	*N*
App2	15	25	*E*
App3	10	30	*E*
App4	10	30	*N*
App5	20	35	*N*
App6	25	35	*N*
App7	30	45	*N*
App8	5	45	*N*
App9	10	10	*E*
App10	5	15	*N*
App11	20	30	*N*
App12	8	15	*N*
App13	8	15	*E*
App14	10	30	*N*
App15	9	20	*N*

1) Feasible Jobs

Feasible jobs of the 15 scheduled applications are first compared under Buyya's framework and the improved framework proposed in this research that considers job priority. The cost-optimization strategy is selected as an example. In the test case, deadline of each application varies from T_{fast} to $5T_{fast}$, in steps of $T_{fast}/5$, and budget of each application varies from C_{cheap} to $5C_{cheap}$, in steps of $C_{cheap}/5$. The results for normal jobs with two different frameworks are as shown in Fig. 3-12 and Fig. 3-13; while the results for emergency jobs are in Fig. 3-13Fig. 3-14 and Fig. 3-15.

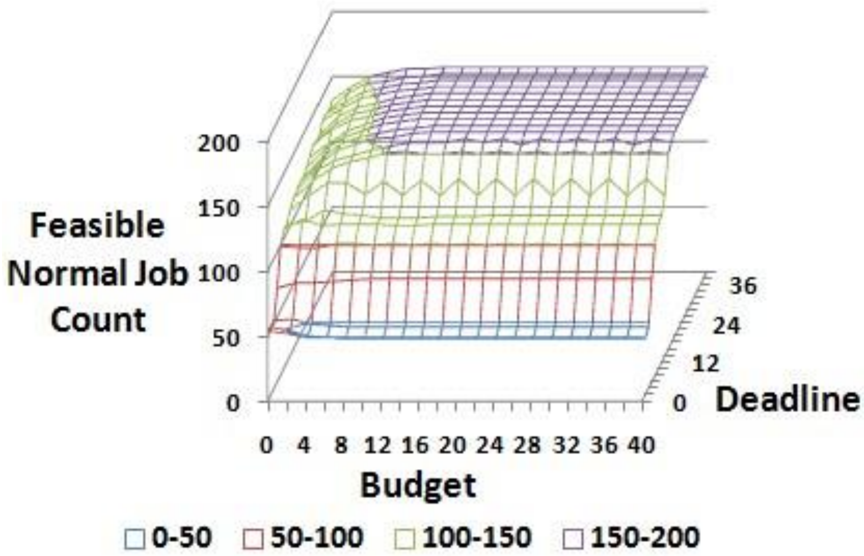


Fig. 3-12. Number of feasible normal jobs of cost-optimization algorithm with Buyya's algorithm

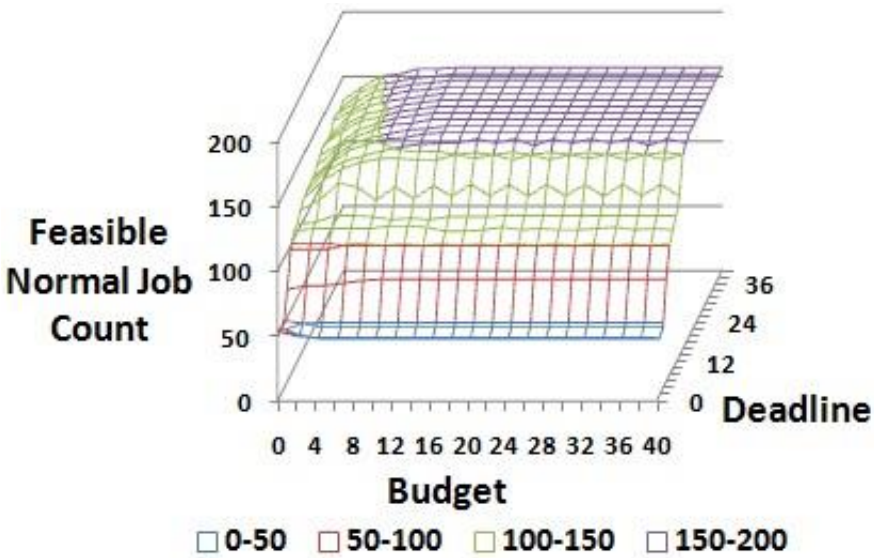


Fig. 3-13. Number of feasible normal jobs of cost-optimization algorithm with improved algorithm

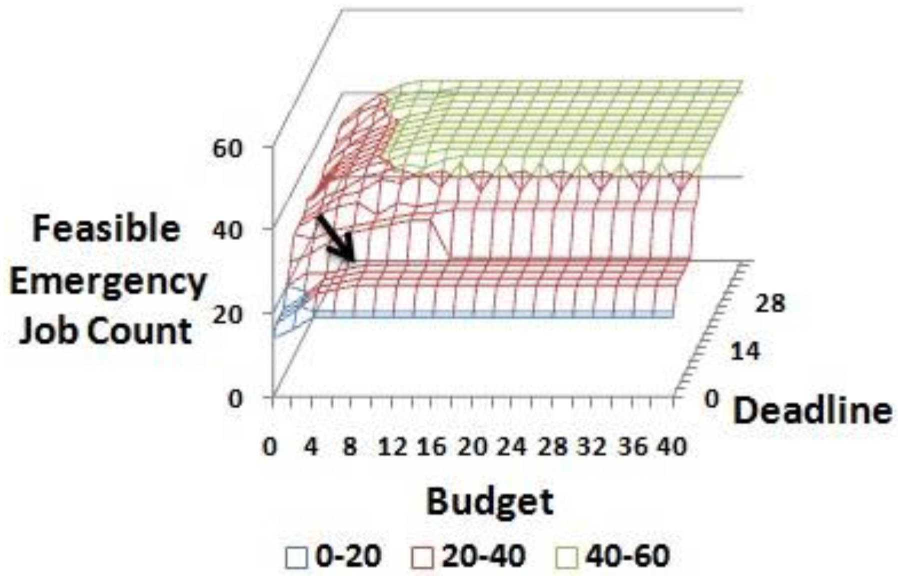


Fig. 3-14. Number of feasible emergency jobs of cost-optimization algorithm with Buyya's algorithm

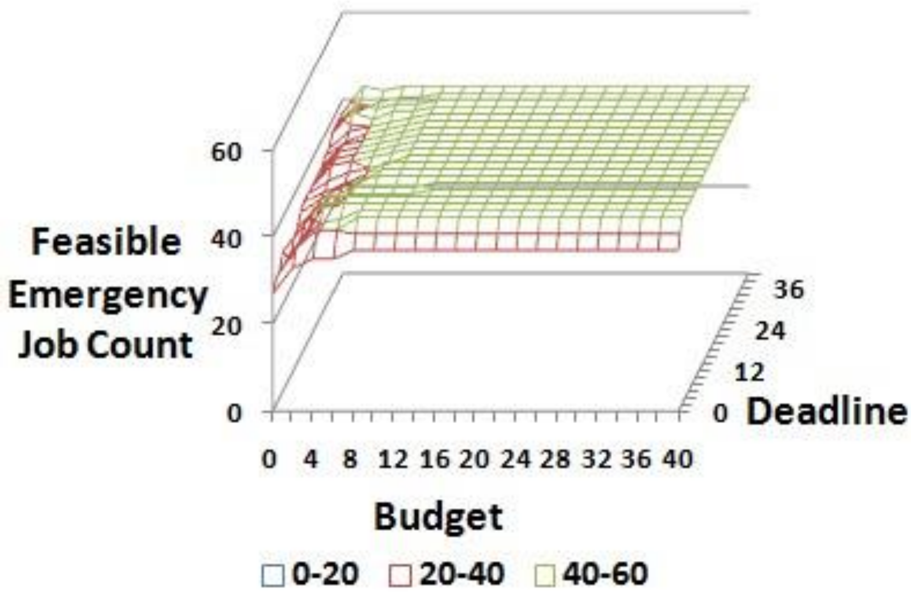


Fig. 3-15. Number of feasible emergency jobs of cost-optimization algorithm with improved algorithm

The results show that the number of feasible jobs increases gradually along with the deadline and budgets become larger and larger. But in my proposed algorithm, emergency applications can be satisfied even when the deadline and budget of the applications are tight. And even if the algorithm satisfies the requirements of emergency applications first, the change of the number of feasible jobs of normal applications is nearly the same as Buyya's algorithm. For the time-optimization algorithm also, the situation is similar.

The feasible job optimization shows the advantage in respect of feasible jobs. Fig. 3-13 shows comparison of feasible jobs execution with feasible job optimization algorithm and time-optimization algorithm. The slope of the right surface is larger than the left surface. This means that generally the feasible job optimization algorithm can process more jobs than the time-optimization algorithm, with a given deadline and budget.

2) Application Execution Time

Fig. 3-16 shows comparison of execution time of cost-optimization algorithm and time-optimization algorithm under the scenario of deadline and budget of each application being T_{fast}*2 and C_{cheap}*2, respectively. In Buyya's framework, the emergency applications cannot be completed under this configuration (see the arrow in Fig. 3-14). In the proposed framework, those applications are feasible. And it can be seen that execution time is significantly reduced by the time-optimization algorithm.

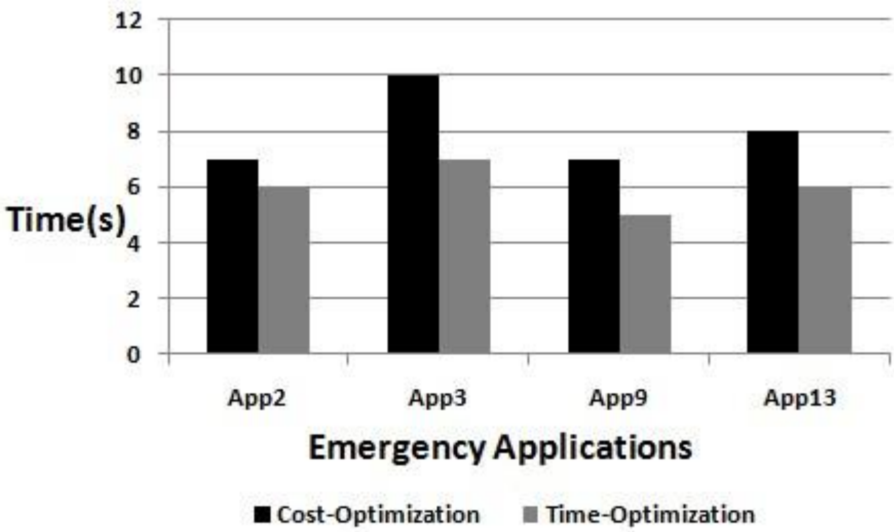


Fig. 3-16. Comparison of application execution time of cost-optimization algorithm and time-optimization algorithm

3) Application Cost

Fig. 3-17 shows comparison of execution cost of cost-optimization algorithm and time-optimization algorithm when deadline and budget of each emergency application are T_{fast}*2 and C_{cheap}*2, respectively. The results show the execution cost is significantly reduced by the cost-optimization algorithm.

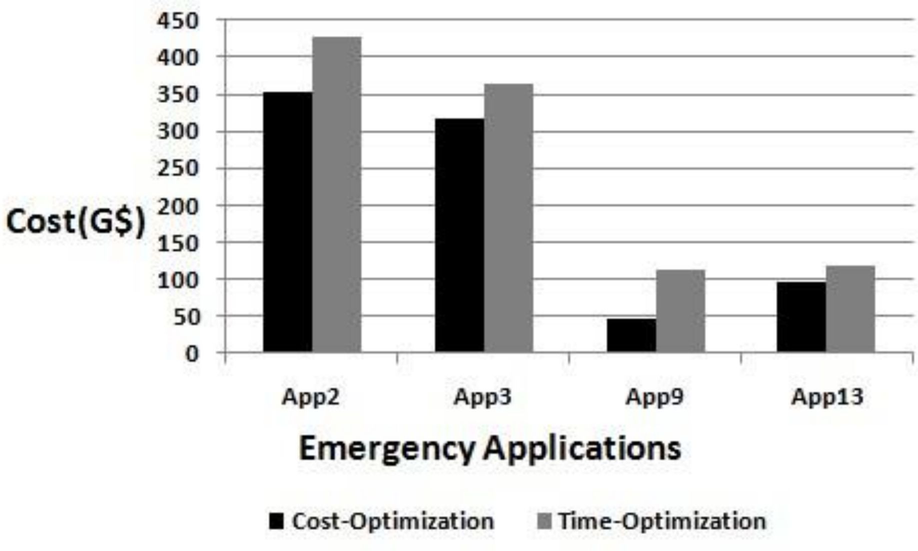


Fig. 3-17. Comparison of application cost of cost-optimization algorithm and time-optimization algorithm

3.5 Cloud Computing Platform for Future's Power Systems

3.5.1 Cloud Data Centre for Smart Grid

The development of smart grid requires integrating various kinds of new technologies in different domains. It is quite meaningful to construct a new IT platform for smart grid to deliver data sharing and high performance computing services. As a new technology trend, cloud computing provides an ideal solution. The cloud computing is proposed in the early of 21th century and draws lots of attention around the world rapidly. Cloud computing can provide various kinds of services (computing power, storage devices, developing platform, software, etc.) in a platform-level and Internet-scale pattern. Literatures [96-98] give the comprehensive introduction of cloud computing.

Physically, the core component of cloud computing is the data center, which houses the physical IT equipment to deliver services to end users. Currently, the large cloud service providers, such as Google and Amazon, are backed up with multiple data centers scattered around the world, while each data center is equipped with hundreds to thousands of servers. For example, Google now has more than ten data centers located in some cities of USA, Hong Kong, Tai Wan and Singapore [99]. Fig. 3-18 shows a snapshot of Google's data center.

Fig. 3-18. Snapshot of Google's data center

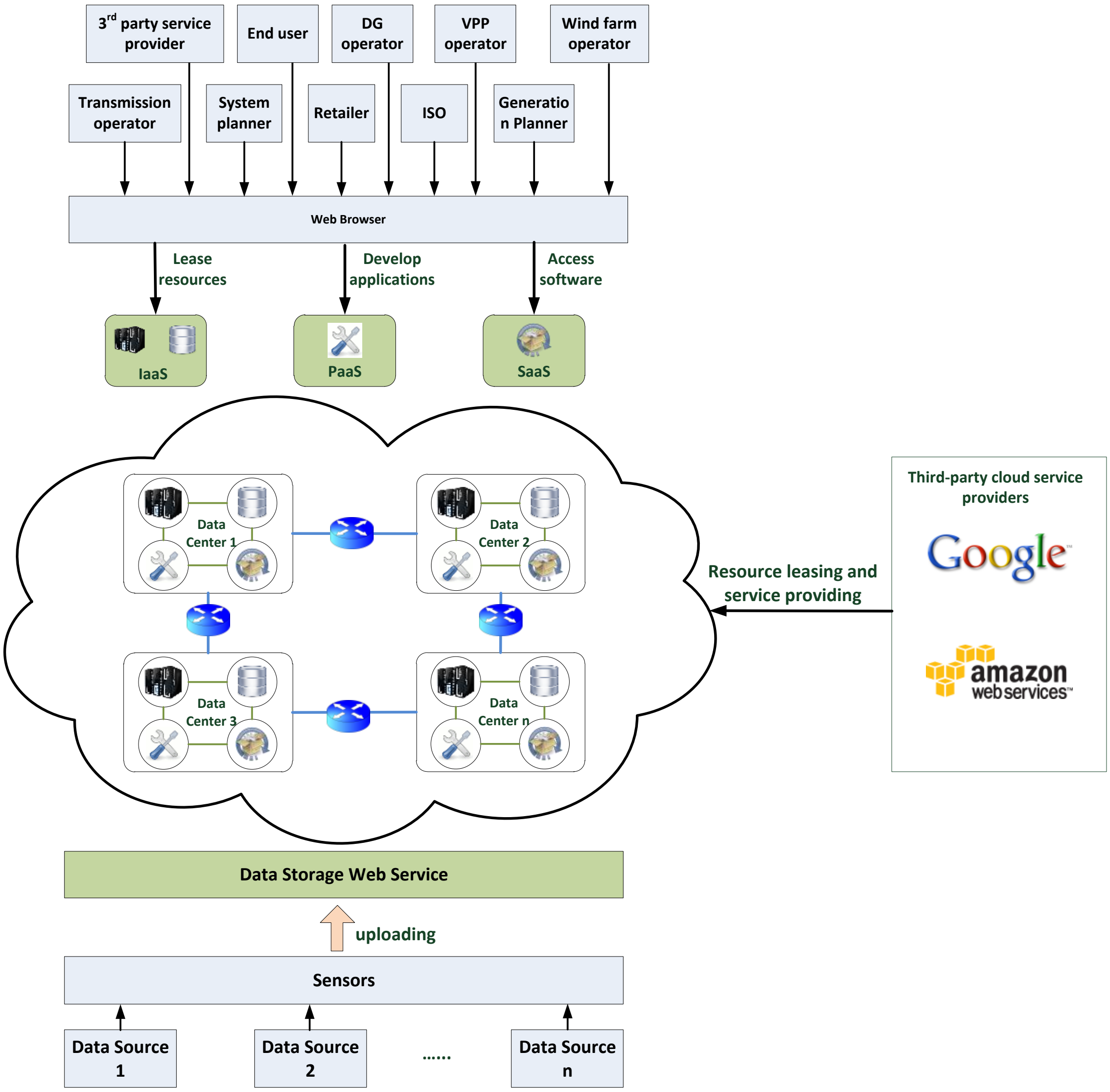


Fig. 3-19.Cloud data center conceptual model for smart grid

In this research, a conceptual cloud data center model for smart grid is proposed, as shown in Fig. 3-19. More than one data center can be constructed. The data centers are connected by Internet. While each data center is equipped with high performance servers and storage devices, it can also lease resources and other kinds of services from the third-party public cloud service providers as necessary. As a virtualized platform, the data centers can integrate the mass data produced by various sources of smart grid, and deliver different kinds of services to the different participants of smart grid.

In cloud computing domain, the services provided by data center are classified into 3 layers: infrastructure as service (*IaaS*), platform as service (PaaS), and software as service (SaaS). In this research, the conception of power cloud data center (PCDC) is proposed to aggregate the distributed information resources such as data storage facility, computational resource, software, etc., and deliver the web-based services to different power market participants, which is depicted in Fig. 3-20.

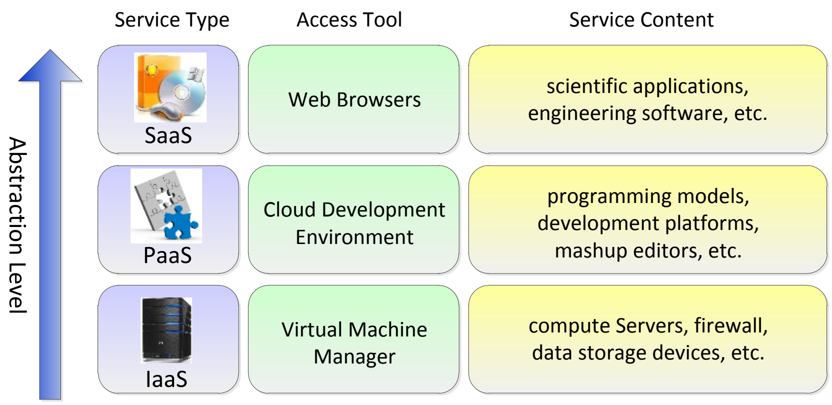


Fig. 3-20. Service model of cloud computing

In the proposed conceptual model, the power cloud data centers also provide these 3-layer services to users. The structure of the proposed conceptual model of PCDC is highly decentralized and distributed, where the underlying information resources are scattered in the grid. Note that this structure is different with the state-of-the-art public cloud computing platforms, such as Google Cloud Engine or Amazon Elastic Cloud (EC2). Those platforms are also decentralized and backed up by multiple geographically distributed data centers. In each data center, hundreds to thousands servers are hosted and connected by local area network (LAN) [100]. However, in PCDC, the large numbers of information resources are more decentralized and cannot be connected by LAN, but pure Internet. This is corresponding to the physical topology of power system, by which the information resources are inherently highly distributed. The role of PCDC is thus to aggregate and manage those resources and provide web-based service interface to users. Such aggregating process can be achieved by the mature grid computing technology (e.g., by use of the well-known middleware 'Globus' [50]).

The aggregated physical information resources are virtualized to form a virtualized resource pool by PCDC. PCDC then delivers the virtualized resources (such as data, computational resource, storage disk, software, etc.) to different users as services. The web-based service interface can be implemented according to the web service standard.

PCDC can also utilize the third-party public cloud computing platforms as complement. For example, some non-sensitive data can be stored in Google BigQuery [101] to benefit from the fast data storage & retrieving. The load balancer of the power cloud data center can also dispatch the computational tasks among the leased third-party resources and the virtualized resource pool.

3.5.2 Cloud Computing-based Information Infrastructure

Based on the conceptual model of PCDC, the global picture of the cloud-based information infrastructure for next-generation power system is depicted in Fig. 3-21. The information physical resources are built on top of the power infrastructure. The information physical resources include data storage facilities, computing resources, database system, software, etc. Those resources are physically distributed and may be located in different organizations, such as control centers and substations. The various kinds of online/offline data of the power infrastructure are collected by the sensors and are stored in the physical information resources either locally or remotely.

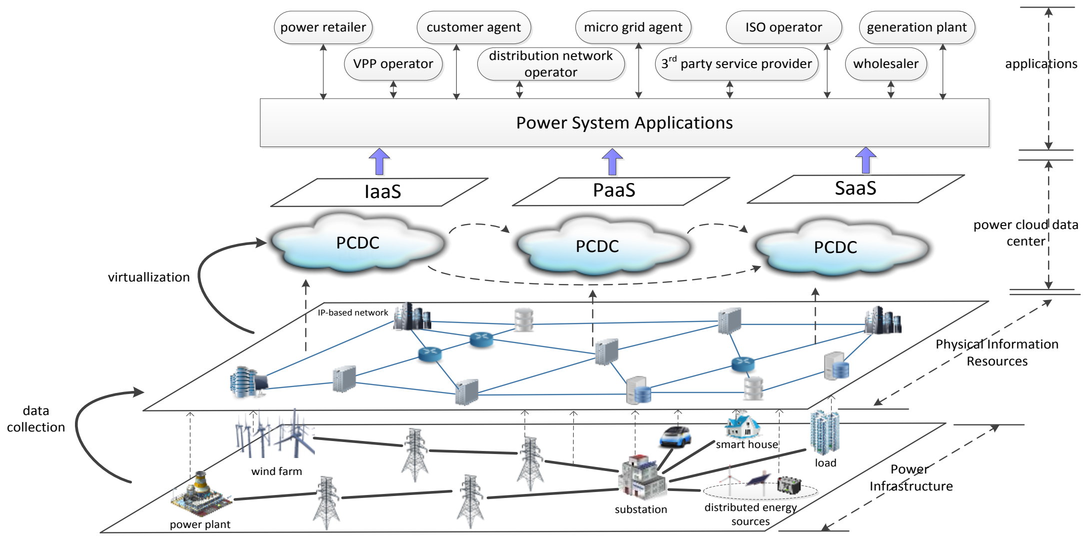


Fig. 3-21. Cloud computing infrastructure for smart grid

The physical information resources are named and addressed by IP-based network (normally Internet). They are aggregated by the grid computing technology, and virtualized and managed by the PCDCs. Since power grid is a large interconnected networked system and the information resources are scattered in the large-scale grid, in practical implementations, multiple PCDCs can be constructed and each of them covers a certain scope of information resources. The information resources are virtualized by the virtual machine manager (VMM) of PCDCs and form a virtualized resource pool. On the top layer of the information infrastructure is the service interface, through which the *IaaS*, *PaaS* and *SaaS* levels of services are delivered by PCDCs. By retrieving data and applying for virtualized resources, different power system applications can be developed by different market participants.

The cloud computing middleware such as FI-WIRE [102] can be utilized to construct the proposed infrastructure. The cloud middleware provides the mechanisms to support resource virtualization & management, and Internet-scale service delivery.

3.5.3 Characteristics of the Infrastructure Model

In this section, the main characteristics of the proposed conceptual model are introduced as follows.

1) **Infrastructure as Service (IaaS)**

The power cloud data center can directly provide the virtualized computing resources and data storage resources to the end users. Here, the resources are virtualized by utilizing the hardware virtualization technology, which is one important technology in the cloud computing domain. The main idea of hardware virtualizing technology is to create logic server images based on the physical hardware to allow users easily access and use the heterogeneous resources in a uniform way. The virtualization technology adds a software layer on top of the physical hardware, which is called virtual machine monitor (VMM). Based on the VMM, the multiple virtualized server images are created, while each server has its own operating system and software stack. The VMM is responsible for mapping the virtualized images to the underlying hardware.

Different actors of smart grid can apply and operate the virtualized computing power and storage devices simply through the web-based interface to satisfy their own needs. For example, the transmission system operator can apply the virtualized high performance Linux cluster to perform the long term transmission network planning analysis; the wind farm operator can apply the storage disks to store the historical wind speed information; the independent system operator (ISO) can apply the computing power to support online security assessment and control; the smart grid academic researchers can apply the computing power to perform large scale scientific simulations, etc.

2) Platform as Service (PaaS)

In addition to deliver the virtualized physical resources to users directly, the power cloud data center can deliver the development platform as the service to allow users develop and deploy smart grid applications through the web browser. The *PaaS*-level service mainly refers to the programming model, development toolkits and the deploy framework. By utilizing the *PaaS*-level service, different users can develop their specific applications and deploy them into the data center directly.

Compared with migrating the locally developed application into the virtualized resources applied by *IaaS*, it is much easier for users to utilize the development toolkits and deploy framework to develop cloud-enabled applications. Users do not need to care about the issues such as compilation and server load balancing, which is handled by the data center in the background.

3) Software as Service (SaaS)

In addition to deliver the virtualized resources and development platform as service, power cloud data center can deliver *SaaS*-level service to users, which means that the various kinds of software can be published by the data center and users can access the software through web browser. The software could be the developed applications by utilizing the *PaaS*-level service, which are deployed as the web-based applications to allow other users access; or it can be the legacy programs or software which are encapsulated as web-based applications. For example, the a academic researcher can develop a wind speed forecast application by utilizing *PaaS* and deploy it as online service, and then the win farm operator can retrieve the historical wind speed data through data center and access the online wind forecast service to do the wind speed forecasting.

4) Distributed Data Management and Parallel Processing

The mass data produced from various data sources in smart grid is uploaded to the data centers through the optical fiber network. Such uploading can follow different strategies. For example, the data sources can transfer the data to multiple data centers, or just transfer it to its nearest data center to improve the transfer efficiency. In the data centers, distributed data structures and some kind of data management algorithms are utilized to support the fast data retrieving, system robustness and large scale data-intensive applications. The distribution of the data management means that there could be multiple replicas of one data set scattered on different data storage nodes or different data centers. Such distributed data management allows for fast retrieval of information. For example, when a user acquires a data set, the data transfer connection can be established between the user and his nearest data replica.

One important requirement of smart grid operation is fast mass data analysis. Many analyses are based on the continuous data flow and can be real-time application. The virtualized computing resources provided by the cloud data center can support parallel data analysis and simulation easily. The virtualized resources are configured with different operating systems and software stacks to meet the needs of different computing tasks. Furthermore, unlike traditional parallel computing, the parallel analysis provided by cloud computing is elastic and scalable. This means users can apply as many resources to do the analysis as they need, and the size of the resources can be dynamically extended or reduced.

5) Robustness and Fault Tolerance

The reliability of the data center is a crucial condition. In addition to upgrade the performance and reliability of the physical equipment of the data centers, such reliability can be ensured by the distributed data management and hardware virtualization technology.

The multiple replicas of the data set improve the system robustness. Considering some analysis is performed on a given data set, if there is some hardware failures occurs, the analysis can be migrated to the storage nodes which store the replicas of the data set. On the other hand, the utilizing of the hardware virtualization technology makes the virtualized machine migrating become very easy. Because of the loose coupling between the virtualized images and the hardware, once the hardware failure occurs, VMM can re-map the image to other physical nodes.

6) Pay on Demand Pattern

In cloud computing domain, a notable feature is the so called ‘pay-on-demand’ computing mode, which implies users can apply the resources as they need, and finally pay for the service provider according to the amount of the resources they used. For the proposed power cloud data center, it is obviously that the capital investment and operation cost will be very high. Such cost can be shared by the users of the data center following the ‘pay-on-demand’ mode. For the investors of the data center, they can get the cost back by leasing the resources and providing the services; for the different data center users (such as the retailer, wholesaler, transmission system operator, etc.), they do need to buy the expensive IT equipment and software, and just need to pay for the data center investor according to their usage.

3.5.4 IaaS-based Application in Power Systems

This section illustrates how power system applications can be built based on the layered *IaaS* service model of cloud computing technology mentioned in Section 3.5.3.

Different power market participants and power researchers can significantly benefit from the *IaaS* service provided by the PCDCs. They can apply for the virtualized data storage and computing devices, and operate them just as local devices. Taking the Amazon EC2 [103] as an example, as the largest public *IaaS* service provider, Amazon EC2 provides a resizable compute capacity in the cloud side. Users can easily obtain the virtual machines with various kinds of operating systems and custom software configurations, and then run the virtual machine instances as they desired. Amazon EC2 provides various kinds of virtual machines, including standard instance, high-memory instance, high-CPU instance, and cluster compute instance, etc. For example, Fig. 3-22 shows the configuration information of an online created 8-core cluster compute instance.

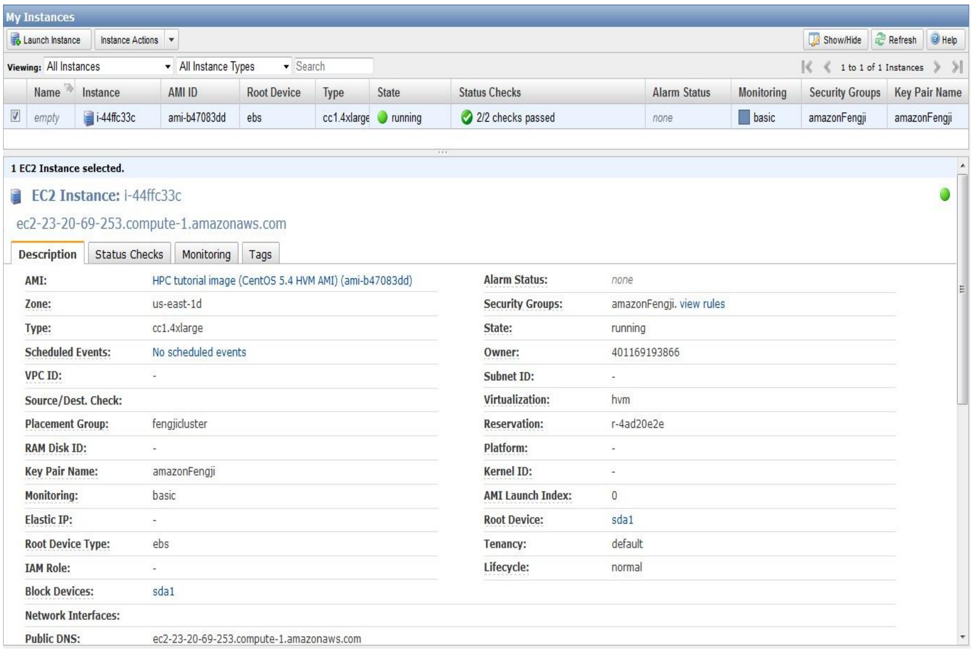


Fig. 3-22. Configuration profile of an 8-core cluster compute instance

Performing high performance computing in those virtualized compute instances is the same as that in the physical computing devices. Some batch management systems such as OpenPBS [104] or Condor [105] can be installed to manage the task queue and the parallel programming models such as MPICH [106] can be utilized to develop the applications. Some examples of applying such techniques in power system applications can be found in literature [107-109].

3.5.5 *PaaS & SaaS-based Application in Power Systems*

This section demonstrates how to build a dynamic security assessment (DSA) application according to the layered PaaS and SaaS service models of cloud computing, which utilizes Goole Application Engine (GAE) tool.

1) Google Application Engine

GAE [110] is a cloud service that can provide an integrated framework for users to develop and deploy scalable web applications directly on Google's infrastructure, rather than plain access to the physical hardware. Users only need to focus on the logic of the applications while GAE automatically handles the common issues such as load balancing, server environment set up and maintaining. Fig. 3-23 shows the architecture of the GAE.

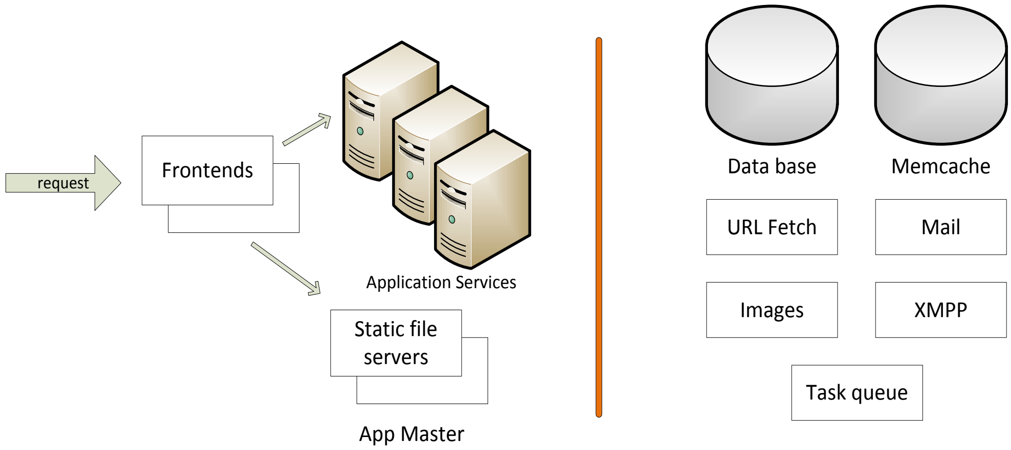


Fig. 3-23. Architecture of Google App Engine [111]

Basically, every request is handled firstly by the App frontends, and then the App frontends read the configuration information of the application and dispatch it to an application server according to a certain load balancing mechanism. The application then starts and runs in the application server.

Currently, GAE provides the configuration and deploy framework for two programming languages: *Java* and *Python*. In this research, the simulation program is developed by *Java* and the development environment is *Eclipse 3.7* [112], which integrates the Google App Engine plug-ins.

2) Pattern Discovery based Dynamic Security Assessment

Literature [113] applied a pattern discovery (PD) algorithm on dynamic security assessment (DSA). PD accepts *n*-dimension data points as the training data, and partitions each dimension into several segments. By combining the partitions segments of each dimension, multiple hyper rectangles are identified in the *data* space, which are called *event*s. Some events might be recursively partitioned. The hyper rectangles where the numbers of data points they contained are statistical meaningful are called *pattern*s. Each pattern is labeled as *secure*/*insecure* according the percentage of the secure training data points it contains. After discovering the patterns, each online operating point can be assessed to be *secure* or *insecure*.

3) Parallel Framework of PD-based DSA application

Although PD can provide high accurate assessment results [113], one performance bottleneck of PD is that when the dimension is high, the recursive process might be very time-consuming. Imaging a 20 dimension data space where each dimension will be divided into 2 segments during the partition, 2^{20} events will be formed. PD then needs to check the 2^{20} events one by one to see whether it should be further portioned. If so, then repeats the partition process on the sub space covered by that event. If there are too many events need to be recursive partitioned, the total execution time will become very large. Therefore, it is necessary to design some parallel mechanism to improve the performance of PD. On the other hand, it is desirable to publish the program as an online service, so that engineers can access it when they want to assess the OPs. Therefore, GAE is the idle choice.

The parallel strategy here is to share the event checking task with multiple processes. The master process starts multiple processes to conduct the event checking and recursive partition task. Meanwhile, each process checks part of the events and send the result back to master. The whole parallel framework is shown in Fig. 3-24.

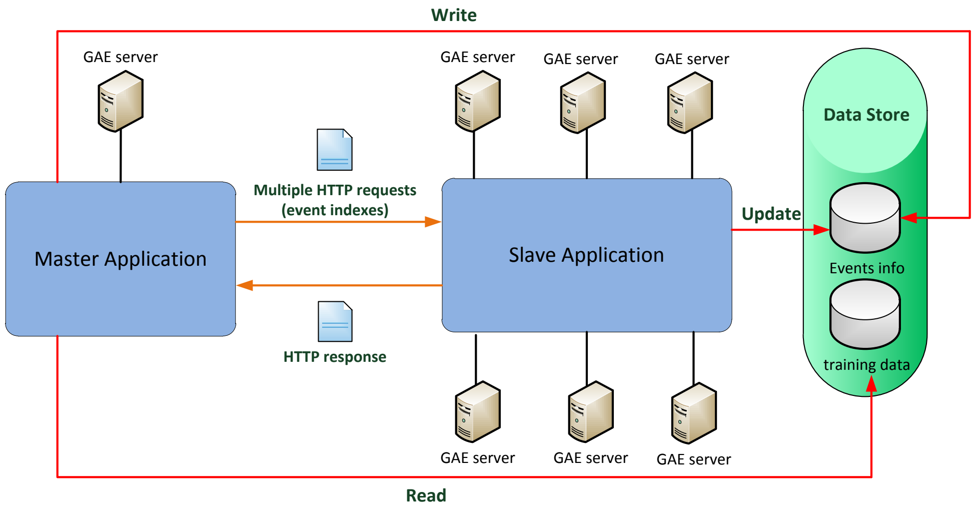


Fig. 3-24. Parallel framework of pattern discovery process

The training data is stored in the data store of GAE, which is a persistence database storing the data shared by all application instances. The master application reads the tanning data to do partition, and forms the event list. Then the master stores the information of the events into data store, and sends *n* HTTP requests to slave application, where each request encapsulates start and end indexes of the events to be processed. For every HTTP request, GAE starts and manages an instance of the slave application automatically. That instance reads the training data from data store, extracts the event indexes from the HTTP request, and do the checking & recursive partition. Finally, each slave updates the event information in data store and sends HTTP response back to the master.

4) Case Studies and Results

Firstly, a transient stability database is artificially generated as the training database. Totally 6,000 different OPs are simulated and the time-domain simulations are performed on the OPs under various disturbances. According to the time-domain simulation result, the OPs are labeled as *secure* or *insecure*. Then, 20 critical features are selected by employing the RELI-FF algorithm [114]. Afterwards, the PD algorithm is employed to discover patterns in the 20-dimension data space. The experiments were repeated 15 times on GAE by increasing the number of the parallel slave instances, and observe the computation time and network overhead shown in Fig. 3-25.

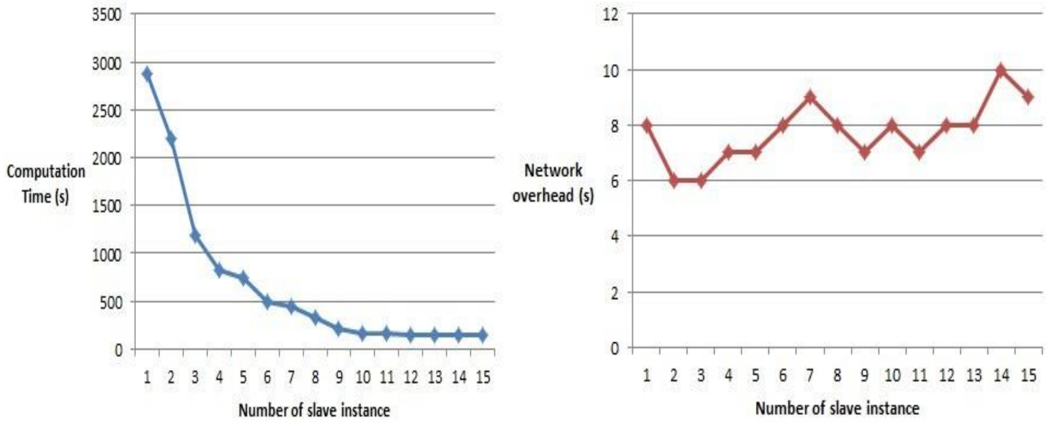


Fig. 3-25. The computation time (left) and network overhead (right) of PD on GAE

It can be seen that the computation time is dramatically reduced along with the increase of parallel jobs, while there exist a little overhead, about 6-10 seconds. Those overheads are mainly incurred by starting the instances, TCP communications, and database access.

The performance of the cloud-based PD and single PC-based PD is compared by setting the number of the slave instance of the cloud-enabled PD to be 10 and varying the number of dimensions from 10 to 20. The single PC-based PD is performed on a 64-bit, Dual CPU DELL PC, with Windows 7 operation system. The computation time comparison and the corresponding network overheads of the cloud-based PD are shown in Fig. 3-26. It can be seen that when the data dimension increases, cloud-based PD shows significantly better performance than the single PC-based PD.

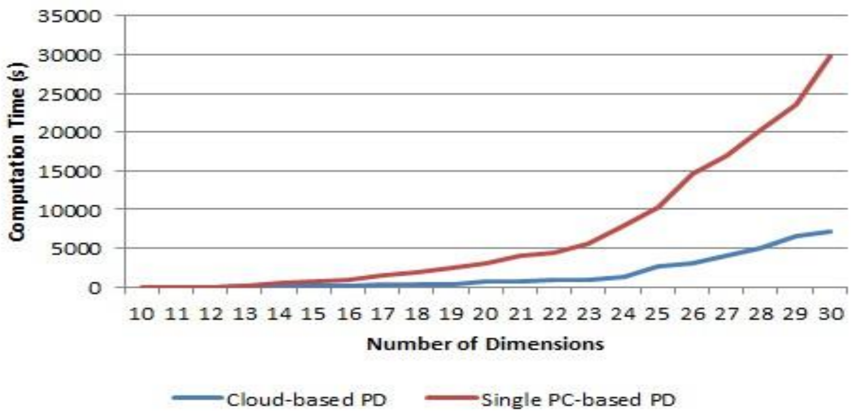


Fig. 3-26. The computation time comparison between the single PC-based PD and the cloud-based PD

When the data dimension is 2 or 3, the pattern discovery results can be visualized. Fig. 3-27 shows the patterns discovered on the 6, OPs with 3 selected critical features. Each rectangle represents a pattern. The technique guide of developing and publishing the web applications through GAE can be referred to [115].

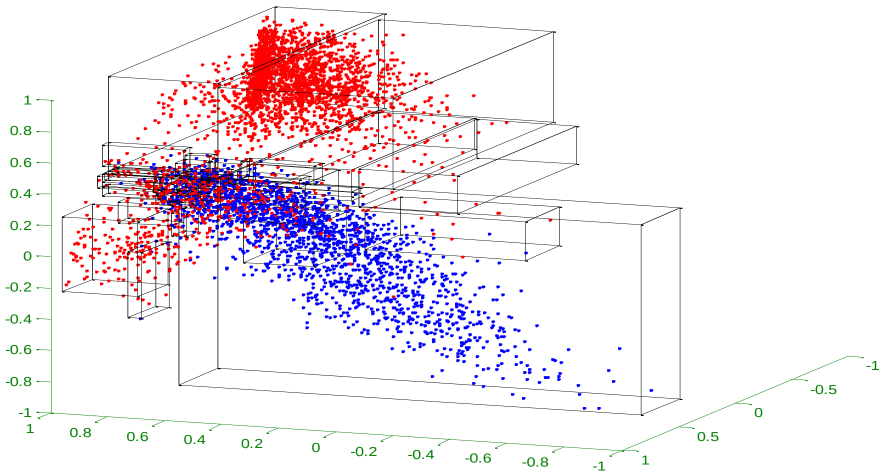


Fig. 3-27. PD demonstration on 3-dimension data space

3.6 Summary

This chapter summarized the computational platforms that will meet the needs of power system analysis, operation, and control, both today and in the future. In the following chapter, a computational intelligence based framework for one of the most essential power system computation tasks— optimal power flow, will be introduced.

Chapter 4

MULTI-CONSTRAINED OPTIMAL POWER FLOW

4.1 Introduction

Optimal power flow (OPF) is an essential component of power system operations. With increasing renewable energy connected to the grid, the challenges for OPF calculations have grown rapidly. It is desirable to have highly efficient and flexible OPF calculation capability with the system operator in order to better dispatch the system in connection with the increased renewable energy resources.

In real-world OPF problems, many practical operational constraints such as prohibited operating zones (POZ) and valve-point effects need to be considered. These problems are very difficult for conventional optimization methods to handle. Discrete control variables such as tap settings of the adjustable transformer are another difficulty. Although a lot of powerful tools such as nonlinear programming (NLP), quadratic programming (QP), linear programming (LP), the Newton method, and the interior point method (IPM) [116, 117] have been used for OPF computations, some of the practical operational constraints have been either ignored or simplified in most previous studies.

In this chapter, a multi-constrained OPF (MCOPF) problem will be proposed and solved with advanced differential evolution algorithms. This problem considers discrete control variables, as well as several practical constraints including transient stability constraints, valve-point effects, POZ of generators, and branch flow thermal constraints. Moreover, cloud computing techniques will be utilized for the parallelized optimization of this MCOPF problem, followed by simulation cases to demonstrate the practicability of the proposed approaches.

4.2 Traditional Optimal Power Flow

4.2.1 Formulation of OPF

Currently, basic OPF problem is mathematically described as a nonlinear programming problem which aims to minimize an objective function while satisfying a serial of equality and inequality constraints.

$$\text{Minimize} \qquad f(\mathbf{u}, \mathbf{x}) \tag{4.1}$$

Subject to $$g(\boldsymbol{u},\boldsymbol{x}) = 0 \tag{4.2}$$

$$h(\boldsymbol{u},\boldsymbol{x}) \leq 0 \tag{4.3}$$

where $\boldsymbol{u}$ is the vector of control variables of the system, $\boldsymbol{x}$ is the vector of dependent variables of the system. Objective function f may be system generating fuel cost, available transfer capability etc. Equality and inequality constraints (4.2) and (4.3) are system physical and operational constraints.

4.2.2 Optimization Techniques

As an alternative to conventional mathematical approaches, modern heuristic optimization techniques such as evolutionary algorithms (EAs), evolutionary programming (EP), and etc. [118] have been employed due to their ability to find almost globally optimal solutions.

Differential evolution (DE) algorithm, as a new branch of EA, proposed by Storn and Price [119], has received increasing attention recently. Studies have shown that DE is simpler, more efficient, and robust than other EAs in many benchmark problems [120]. Meanwhile, it can inherently perform in parallel, so as to obtain both computational speed and performance improvements [121]. There have been studies on the use of DE in power system applications [122-123, 69]. In [69], DE is used to solve transient-stability constrained OPF (TSOPF) with superior performance compared with other approaches. However, discrete control variables and practical constraints such as POZ and valve-point effects of generating units have not been considered.

Besides, in conventional DE, an appropriate trial vector generation strategy and its associated control parameters, which significantly affect optimization, have to be pre-determined through a time-consuming trial-and-error search process. Besides, the selected strategy may still not be the most effective one throughout the evolution process because DE population may move to different searching areas where other strategies may perform better. This problem is more serious in highly-constrained optimization problems such as multi-constrained OPF (MCOPF) with discrete control variables and the practical constraints discussed above. To solve the nonlinear, non-convex, discrete and non-differentiable MCOPF problem, more advanced algorithms are needed.

In this chapter, a self-adaptive DE (SaDE) [124] based approach is introduced to address limitations of the conventional DE in solving the MCOPF problem. SaDE determines the generation strategy and its associated control parameters values self-adaptively, learning from experience in producing improved solutions. Its performance has been validated on many benchmark problems.

4.3 Practical Operation Constraints

4.3.1 Generator Valve Point Effects

A fuel cost function is obtained based on a ripple curve for more accurate modeling. This curve contains higher order nonlinearity and discontinuity due to valve-point effects and can be refined by a sine function [125]. Therefore, f is accurately represented as,

$$f = \sum_{i=1}^{N_g} \left[a_i P_{gi}^2 + b_i P_{gi} + c_i + \left| w_i \sin\left(e_i \left(P_{gi}^{\min} - P_{gi} \right) \right) \right| \right] \tag{4.4}$$

where, a_i, b_i, c_i are the cost coefficients of unit i; w_i and e_i are constants of its valve-point effects.

4.3.2 Generator Prohibited Operating Zones

For convenience, unit generation output is usually assumed to have been adjusted smoothly. Practically, units may have some prohibited operating zones (POZs) in the input-output curve due to steam valve operation or vibration in a shaft bearing. Adjusting output P_{gi} of a unit must avoid operations in these POZs. Therefore, feasible operating zones of unit *i* can be described as,

$$\begin{cases} P_{gi}^{\min} \le P_{gi} \le P_{gi,1}^{l} \\ P_{gi,j-1}^{u} \le P_{gi} \le P_{gi,j}^{l} \\ P_{gi,N_{zi}}^{u} \le P_{gi} \le P_{gi}^{\max} \end{cases}, \quad \begin{array}{l} i = 1,2,\ldots,N_g \\ j = 2,3,\ldots,N_{zi} \end{array} \tag{4.5}$$

4.3.3 Transformer Discrete Control Region

Transformer taps should be adjusted step-by-step to ensure bus voltage is within allowable limits. The region of transformer tap settings, regarded as discrete control variables in this MCOPF model, is shown as,

$$T_i \in \left\{T_i^{\min}, \cdots, T_k, \cdots, T_i^{\max}\right\}, i = 1, 2, ..., N_t \tag{4.6}$$

4.3.4 Thermal Constraints

$$\left|S_i\left(V, \theta, T\right)\right| \le S_i^{\max}, i = 1, 2, ..., N_l \tag{4.7}$$

where, N_{zi} is the total number of POZs of generator *i*; $P_{gi,j}^{l}$ and $P_{gi,j}^{u}$ are lower and upper bounds of *j*th POZ of generator *i*; N_l is the total number of branches; S_i is the apparent power flow in branch *i*; $P_{gi}^{\min}$, $P_{gi}^{\max}$, $Q_{gi}^{\min}$, $Q_{gi}^{\max}$, $V_{gi}^{\min}$, $V_{gi}^{\max}$, $T_i^{\min}$, $T_i^{\max}$, $V_{di}^{\min}$, $V_{di}^{\max}$, and $S_i^{\max}$ are the corresponding lower and upper limits of these variables.

4.3.5 Transient Stability Constraints

The power system under transient conditions can be described by a set of differential-algebraic equations [126] referred to as "swing equations" corresponding to machine rotors. In this model, the swing equations denoted here are,

$$M_i \dot{\tilde{\omega}}_i = P_{mi} - P_{ei} - \frac{M_i}{M_T} P_{COI} \equiv PAC_i \tag{4.8}$$

$$M_T = \sum\nolimits_{i=1}^{N_g} M_i \,, \quad P_{COI} = \sum\nolimits_{i=1}^{N_g} \left(P_{mi} - P_{ei}\right) \tag{4.9}$$

where, M_i is the moment of inertia of generator i; ω_i is the rotor angular speed of generator *i*; P_{mi} and P_{ei} are the mechanical power input and electrical power output of generator i, respectively; and PAC_i is the accelerating power of generator *i*.

Meanwhile, the transient energy function (TEF) of a power system is described as:

$$KE = \sum\nolimits_{i=1}^{N_g} M_i \tilde{\omega}_i^2 / 2 \,, \quad PE = -\sum\nolimits_{i=1}^{N_g} \int_{\theta_i^{SEP}}^{\theta_i} PAC_i^P \, d\theta_i \tag{4.10}$$

$$TEF = KE + PE \tag{4.11}$$

where, θ_i is the rotor angle of generator i and θ_i^{SEP} is the rotor angle of post-fault system stable equilibrium point; PAC_i^P is the accelerating power of the post-fault systems; KE is the system kinetic energy, PE is the system potential energy, and TEF should be their sum according to the transient energy function (TEF) theory [127]. If a system is stable after a contingency, KE and PE are both positive and keep confined by TEF; otherwise, KE increases and PE drops rapidly. Therefore, the constraint in Eq. (4.11) can be used to determine the system operating transient stable condition.

Consideration of transient stability constraints in the OPF problems has become increasingly imperative [128]. It is very difficult to solve even for small power systems because transient stability is a dynamic concept and differential equations are involved. The existing approaches [129-132] are mainly to convert the differential equations into equivalent algebraic equations as inequality constraints; and then reformulate the OPF problem as an algebraic one that can be solved by conventional optimization techniques. However, these methods may introduce errors due to the approximation process, and some time may experience convergence problems because of the high dimensions of the reformulated problems.

4.4 Proposed Multi-Constrained OPF Model

4.4.1 Formulation of MCOPF

The MCOPF problem can be formulated as,

$$\underset{\boldsymbol{u_c},\boldsymbol{u_d}}{Min.}\ f\left(\boldsymbol{u_c},\boldsymbol{u_d},\boldsymbol{x}\right) \tag{4.12}$$

$$s.t.\ g\left(\boldsymbol{u_c},\boldsymbol{u_d},\boldsymbol{x}\right)=0 \tag{4.13}$$

$$h\left(\boldsymbol{u_c},\boldsymbol{u_d},\boldsymbol{x}\right)\leq 0 \tag{4.14}$$

$$KE\left(\boldsymbol{u_c},\boldsymbol{u_d},\boldsymbol{x}\right)<TEF\left(\boldsymbol{u_c},\boldsymbol{u_d},\boldsymbol{x}\right) \tag{4.15}$$

where $\boldsymbol{u_c}$ and $\boldsymbol{u_d}$ are vectors of continuous and discrete control variables, respectively; $\boldsymbol{x}$ is the vector of dependent variables; f is the objective function; g and h are vectors of functions, which model equality and inequality constraints respectively; KE and TEF are system kinetic energy and transient energy functions, respectively, calculated through transient stability analysis. In this research, $\boldsymbol{u}_c$, $\boldsymbol{u}_d$, and $\boldsymbol{x}$ can be expressed as,

$$\boldsymbol{u_c} = \left[P_{g2},...,P_{gN_g}, V_{g1},...,V_{gN_g} \right]^T \tag{4.16}$$

$$\boldsymbol{u}_d = \left[T_1,...,T_{N_t} \right]^T \tag{4.17}$$

$$\boldsymbol{x} = \left[P_{g1}, V_{d1},...,V_{dN_{pq}}, \theta_{g2},...,\theta_{gN_g}, \theta_{d1},...,\theta_{dN_{pq}}, Q_{g1},...,Q_{gN_g} \right]^T \tag{4.18}$$

where, N_g is the total number of units (unit 1 is the slack unit); N_t is the total number of adjustable transformers; N_{pq} is the total number of PQ-buses; P_{gi} and Q_{gi} are active and reactive power output of unit i; V_{gi}, θ_{gi} and V_{di}, θ_{di} are voltage magnitudes and angles of unit i and PQ-bus i, respectively; and T_i is the tap setting of adjustable transformer i.

1) Objective Function

In this research, the objective of MCOPF is to minimize the total fuel cost of generation units. A fuel cost function is obtained based on a ripple curve for more accurate modeling. This curve contains higher order nonlinearity and discontinuity due to valve-point effects and can be refined by a sine function [125]. Therefore, f is accurately represented as,

$$f = \sum_{i=1}^{N_g} \left[a_i P_{gi}^2 + b_i P_{gi} + c_i + \left| w_i \sin\left(e_i \left(P_{gi}^{\min} - P_{gi} \right) \right) \right| \right] \tag{4.19}$$

where, a_i , b_i , c_i are the cost coefficients of unit i; w_i and e_i are constants of its valve-point effects.

2) Equality constraints

Equality constraints are the nonlinear power flow equations,

$$\begin{cases} P_{gi} - P_{di} - \sum_{j \in i} P_{ij}\left(V, \theta, T \right) = 0 \\ Q_{gi} - Q_{di} - \sum_{j \in i} Q_{ij}\left(V, \theta, T \right) = 0 \end{cases}, \quad \begin{matrix} i = 1, 2, ..., N \\ j = 1, 2, ..., N \end{matrix} \tag{4.20}$$

where, N is the total number of all buses; P_{di} and Q_{di} are active and reactive power loads of bus i; $j \in i$ means bus j is connected with bus i; P_{ij} and Q_{ij} are active and reactive power output between buses i and j; V, θ and T are vectors of voltage magnitude, voltage angle, and transformer tap, respectively.

3) Inequality Constraints

a) Generator constraints

(i) Active power output limits considering POZs

For convenience, unit generation output is usually assumed to have been adjusted smoothly. Practically, units may have some prohibited operating zones (POZs) in the input-output curve due to steam valve operation or vibration in a shaft bearing. Adjusting output P_{gi} of a unit must avoid operations in these POZs. Therefore, feasible operating zones of unit *i* can be described as,

$$\begin{cases} P_{gi}^{\min} \le P_{gi} \le P_{gi,1}^{l} \\ P_{gi,j-1}^{u} \le P_{gi} \le P_{gi,j}^{l} \\ P_{gi,N_{zi}}^{u} \le P_{gi} \le P_{gi}^{\max} \end{cases}, \quad \begin{matrix} i = 1,2,\ldots,N_g \\ j = 2,3,\ldots,N_{zi} \end{matrix} \tag{4.21}$$

(ii) Reactive power output limits

$$Q_{gi}^{\min} \le Q_{gi} \le Q_{gi}^{\max}, i = 1,2,\ldots,N_g \tag{4.22}$$

(iii) Voltage limits

$$V_{gi}^{\min} \le V_{gi} \le V_{gi}^{\max}, i = 1,2,\ldots,N_g \tag{4.23}$$

b) Transformer constraints

Transformer taps should be adjusted step-by-step to ensure bus voltage is within allowable limits. The region of transformer tap settings, regarded as discrete control variables in this MCOPF model, is shown as,

$$T_i \in \left\{T_i^{\min},\cdots,T_k,\cdots,T_i^{\max}\right\}, i = 1,2,\ldots,N_t \tag{4.24}$$

c) PQ-bus voltage constraints

$$V_{di}^{\min} \le V_{di} \le V_{di}^{\max}, i = 1,2,\ldots,N_{pq} \tag{4.25}$$

d) Transmission line flow thermal constraints

$$\left|S_i\left(V,\theta,T\right)\right| \le S_i^{\max}, i = 1,2,\ldots,N_l \tag{4.26}$$

where, N_{zi} is the total number of POZs of generator *i*; $P_{gi,j}^{l}$ and $P_{gi,j}^{u}$ are lower and upper bounds of *j*th POZ of generator *i; N_l* is the total number of branches; S_i is the apparent power flow in branch *i*; $P_{gi}^{\min}$, $P_{gi}^{\max}$, $Q_{gi}^{\min}$, $Q_{gi}^{\max}$, $V_{gi}^{\min}$, $V_{gi}^{\max}$, $T_i^{\min}$, $T_i^{\max}$, $V_{di}^{\min}$, $V_{di}^{\max}$, and $S_i^{\max}$ are the corresponding lower and upper limits of these variables.

4) Transient Stability Constraints

The power system under transient conditions can be described by a set of differential-algebraic equations [126] referred to as "swing equations" corresponding to machine rotors. In this model, the swing equations denoted here are,

$$M_i \dot{\tilde{\omega}}_i = P_{mi} - P_{ei} - \frac{M_i}{M_T} P_{COI} \equiv PAC_i \tag{4.27}$$

$$M_T = \sum_{i=1}^{N_g} M_i \ , \qquad P_{COI} = \sum_{i=1}^{N_g} \left(P_{mi} - P_{ei} \right) \tag{4.28}$$

where, M_i is the moment of inertia of generator *i*; ω_i is the rotor angular speed of generator *i*; P_{mi} and P_{ei} are the mechanical power input and electrical power output of generator *i*, respectively; and PAC_i is the accelerating power of generator *i*.

Meanwhile, the transient energy function (TEF) of a power system is described as:

$$KE = \sum_{i=1}^{N_g} M_i \tilde{\omega}_i^2 / 2 \ , \ \ PE = -\sum_{i=1}^{N_g} \int_{\theta_i^{SEP}}^{\theta_i} PAC_i^P d\theta_i \tag{4.29}$$

$$TEF = KE + PE \tag{4.30}$$

where, θ_i is the rotor angle of generator *i* and θ_i^{SEP} is the rotor angle of post-fault system stable equilibrium point; PAC_i^P is the accelerating power of the post-fault systems; *KE* is the system kinetic energy, *PE* is the system potential energy, and *TEF* should be their sum according to the transient energy function (TEF) theory [127].

Fig. 4-1 and Fig. 4-2 illustrate the variation of *KE*, *PE*, and *TEF* along typical stable and unstable trajectories respectively. It is obviously that if a system is stable after a contingency, *KE* and *PE* are both positive and keep confined by *TEF*; otherwise, *KE* increases and *PE* drops rapidly. Therefore, the constraint in Eq. (4.15) can be used to determine the system operating transient stable condition.

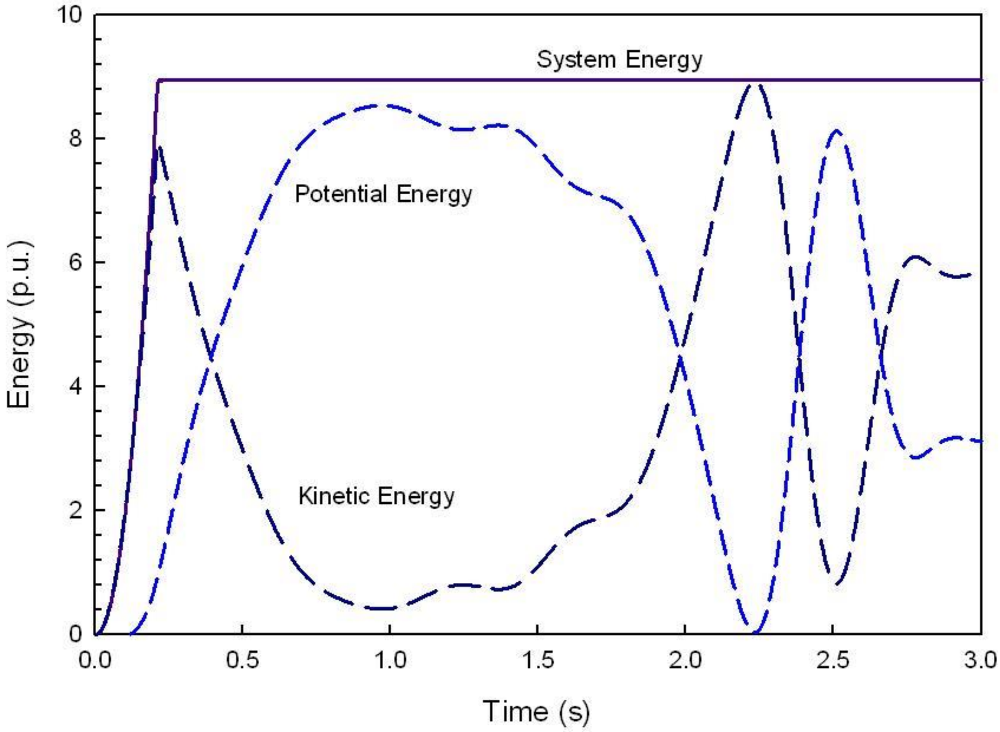


Fig. 4-1. Variation of *KE, PE*, and *TEF* along a stable trajectory

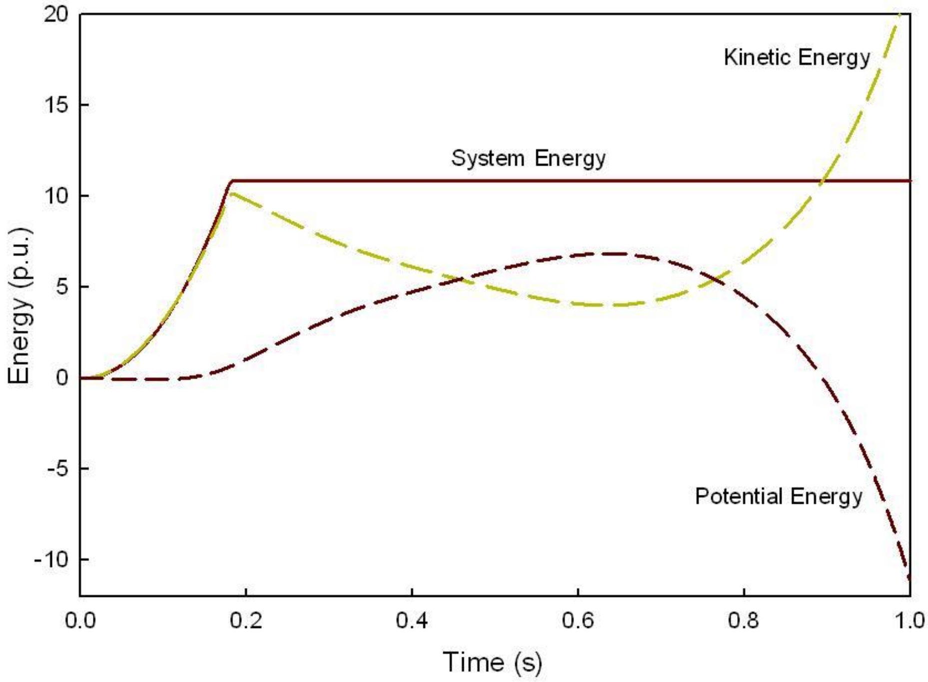


Fig. 4-2. Variation of *KE*, *PE*, and *TEF* along an unstable trajectory

4.4.2 Handling of Discrete Variables

As introduced in Section 4.4.1, the control variables transformer tap settings are discrete variables. During the optimization process, some measures have to be taken here to handle this. In this research, a heuristic approach is proposed which is able to cope with these discrete control variables in large-scale OPF computations. Specifically, this approach works as follows:

1) Firstly, the discrete variables are treated as continuous and a relaxed solution is computed by a classical OPF;
2) Then, an iterative procedure is applied which at each iteration moves, according to some rules, a subset of discrete variables (and all discrete variables that have continuous values) to one of their closest upper or lower discrete values; then an OPF computation is launched to re-optimize the continuous variables only;
3) Finally, the iterative procedure is terminated once a feasible solution is found whose integrity gap is sufficiently small or when no significant improvement of the objective is observed with respect to the previous iteration.

4.4.3 Fitness Evaluation Function

In MCOPF model, for each individual a power flow calculation is performed and a fitness value F_i is assigned to measure the quality of the individual. Violations of feasible regions are considered as punishments in the fitness function (4.31).

$$F_i = 1 / \left(f_i + K_V F_{Vi} + K_Q F_{Qi} + K_{BF} F_{BFi} + K_{PS} F_{PS} \right) \tag{4.31}$$

$$F_{Vi} = \sum_{j=1}^{N_{pq}} \left[\left(\left| V_{ij} - V_j^{\lim} \right| \right) / \left(V_j^{\max} - V_j^{\min} \right) \right] \tag{4.32}$$

$$F_{Qi} = \sum_{j=1}^{N_g} \left[\left(\left| Q_{ij} - Q_j^{\lim} \right| \right) / \left(Q_j^{\max} - Q_j^{\min} \right) \right] \tag{4.33}$$

$$F_{BFi} = \sum_{j=1}^{N_l} \left[\left(\left| S_{ij} - S_j^{\max} \right| \right) / S_j^{\max} \right] \tag{4.34}$$

$$F_{PS} = \left(P_{slack} - P_{slack}^{\lim} \right) / \left(P_{slack}^{\max} - P_{slack}^{\min} \right) \tag{4.35}$$

where, F_{Vi}, F_{Qi}, F_{BFi} are sums of violations of *PQ*-bus voltage limitations, generator reactive power output limitations, branch flow limitations of individual *i* respectively; F_{PS} is sum of violation of active power limitation of the slack bus; K_V, K_Q, K_{BF} and K_{PS} are the corresponding penalty coefficients of each violation punishment; and $V_j^{\lim}$, $Q_j^{\lim}$, and $P_{slack}^{\lim}$ are the violated upper or lower limits of voltage of *PQ*-bus *j*, reactive power output of unit *j*, and active power output of slack bus, respectively.

In this research, K_V, K_Q, K_{BF}, and K_{PS} of *G*th generation are dynamically set according to,

$$K_V(G) = K_Q(G) = K_{BF}(G) = K_{PS}(G) = K_{\min} + G(K_{\max} - K_{\min}) / G_{\max} \tag{4.36}$$

where, $G_{\max}$ is the maximum number of generations, $K_{\min}$ and $K_{\max}$ are set to 0.005 and 1.0, respectively. Each individual is assigned an index "*feasible*" to record whether it is feasible. If any of F_{Vi}, F_{Qi}, F_{BFi}, and F_{PSi} exceeds violation tolerance (set as 0.0001 here), individual *i* is considered unfeasible, i.e. assigns *feasible* as "NO"; otherwise, *feasible* is set to "YES".

4.4.4 Transient Stability Assessment

To satisfy transient stability constraints in MCOPF, each individual is also assigned an index "*stable*" to show if the system is transient stable under current system configuration defined by this individual. "*stable*" is calculated through a transient stability assessment (TSA) using a hybrid method which combines time simulation trial-and-error methods with TEF theory presented in Section 4.3.5 which makes TSA more practical to consider detailed system models [133]. Thus, a hybrid assessment procedure is developed as follows:

(a) Perform a time-domain simulation till the contingency is cleared; and calculate the value of TEF at this moment and denote it as TEF_{cl};

(b) Continue the time-domain simulation in Step (1). At each time step, calculate KE of the system, if KE surpassed the boundary of TEF_{cl}, assign stable as "NO", and stop the simulation;

(c) Otherwise, if the predefined simulation period reached and KE is still bounded within the value of TEF_{cl}, assign stable as "YES".

Besides, when a short circuit fault is cleared by disconnecting the faulted branch line, the generators that have accelerated will decelerate and come back into synchronism with the rest of the system. If they do not, and the system becomes unstable, there is a risk of widespread blackouts and of mechanical damage to generators. Critical clearing time (CCT) is the maximum time interval by which the fault must be cleared in order to preserve the system stability and it can be determined with time domain simulation [127]. It is used as a transient stability index.

4.5 Advanced Differential Evolution Algorithms for MCOPF

4.5.1 Conventional Differential Evolution Algorithm

DE is technically a population-based EA, which employs mutation and crossover operators as search mechanisms, and a selection operator to lead the search towards the most promising regions of the solution space. The basic principle of conventional DE is shown as follows:

1) Strategy & Parameter Setup:

To set up the reproduction strategy *DE/x/y/z*, the associated parameters *NP, MF, CR,* number of control variables *D* and generation times G_{max}.

2) Initialization:

To initial a *NP*-size population of *D*-dimensional individual corresponding to:

$$u_{i,j}^{(G)} = u_j^{\min} + rand_j(0,1)(u_j^{\max} - u_j^{\min}) \tag{4.37}$$

where *G* is the number of generation and in this step *G=0*, $i = 1, 2, \cdots, NP$, $j = 1, 2, ..., D$, $u_{i,j}$ is the *j*th control variable of *i*th individual, $u_j^{\min}$ and $u_j^{\max}$ are the prescribed minimum and maximum bounds of *j*th control variable. Generally, discrete control variable is then rounded off to its nearest discrete value.

3) Mutation:

To generate mutant individuals $\boldsymbol{v_i}$ of current generation with fixed *MF* and strategy (taking *DE/rand/1/bin* for instance here):

$$v_{i,j}^{(G+1)} = u_{r1,j}^{(G)} + MF(u_{r2,j}^{(G)} - u_{r3,j}^{(G)}) \tag{4.38}$$

where $r1, r2, r3 = 0, 1, ..., NP - 1$ are randomly chosen and mutually different and also different from the current index *i*.

4) Crossover:

For each individual $\boldsymbol{u_i}$, an offspring individual $\boldsymbol{u'_I}$ is generated by the predetermined *CR*:

$$u'^{(G+1)}_{i,j} = \begin{cases} v^{(G+1)}_{i,j}, \, if \;\; rand[0,1) < CR \; or \;\; j = j_{rand} \\ u^{(G)}_{i,j} \;\;, otherwise \end{cases} \tag{4.39}$$

where j_{rand} is a randomly chosen integer in [1, *NP*] to ensure that $u'_{i,j}$ will get at least one element from the mutation individual.

5) Evaluation & Selection/Competition:

Offspring individual $\boldsymbol{u'_i}$ generated in Step 4 will compete with its parent individual $\boldsymbol{u_i}$ by using the greedy selection criterion:

$$u^{(G+1)}_{i,j} = \begin{cases} u'^{(G+1)}_{i,j}, \, if \;\; f(u'^{(G+1)}_{i,j}) \le f(u^{(G)}_{i,j}) \\ u^{(G)}_{i,j} \;\;, otherwise \end{cases} \tag{4.40}$$

where $f(\cdot)$ is the objective function of evaluation and the condition $f(u'^{(G+1)}_{i,j}) \le f(u^{(G)}_{i,j})$ means that $u'^{(G+1)}_{i,j}$ is better than $u^{(G)}_{i,j}$.

6) Population Iteration & Termination:

Once the new population is created, the processes of mutation, crossover and selection are repeated until the number of generations reaches the preset G_{max}..

There exist many strategies for population reproduction resulting in different variants. Price and Storn proposed several DE strategies using the notation *DE/x/y/z*, where *x*, which is either a randomly chosen vector or the best vector of the current generation, is the vector to be mutated; *y* is the number of vectors used in the mutation; and *z* is the crossover method. The performance of DE depends on the selection of strategy and three critical control parameters: population size *NP*, mutation scaling factor *MF*, and crossover rate *CR*. *NP* is a user-specified parameter to address problems with different complexities; *MF* controls the amplification of differential variations; and *CR* controls the influence of parents.

Proper setting of the strategy and associated control parameters is crucial to the success of the algorithm [134]. To achieve this, a time-consuming trial-and-error search process needs to be performed to select the strategy and the parameters settings which are then applied to the evolution process. However, the population may move through different regions of the search space during the evolution process, within which certain strategies and control parameters may be more suitable than others. A better mechanism for parameter selection is highly desirable.

4.5.2 Self-Adaptive Differential Evolution Algorithm

SaDE algorithm takes the self-adaptability of DE one step further by choosing reproduction strategies and the associated control parameters with a predefined constant learning period (*LP*) for each individual in the current generation. The main characteristics of SaDE are summarized as follows.

1) Trial Vector Generation Strategy Adaptation

A strategy candidate pool (SCPool) is used in SaDE. Let *X* denote the total number of strategies in the SCPool. Firstly, a success-failure memory (SF-Memory) is designed to store success and failure counts of each strategy in the pool after a certain number of *LP* generations. At the G^{th} generation ($G = 1,2,\dots,G_{max}$), the total number of individuals generated by the x^{th} strategy ($x = 1,2,\dots,X$), which can defeat its opponents and thus successfully enters the next generation, is recorded as $ns_{x,G}$, and this x^{th} strategy is called *successful strategy*; otherwise, for the failed individuals, the number is marked as $nf_{x,G}$.

After *LP* generations, there is a strategy probability value $SP_{x,G}$ for each strategy at the G^{th} generation, calculated from its SF-Memory, for generating improved solutions,

$$SP_{x,G} = N_{x,G} / \sum_{x=1}^{X} N_{x,G} \tag{4.41}$$

$$N_{x,G} = \frac{\sum_{g=G-LP}^{G-1} ns_{x,g}}{\sum_{g=G-LP}^{G-1} ns_{x,g} + \sum_{g=G-LP}^{G-1} nf_{x,g}} + \varepsilon \tag{4.42}$$

and $\varepsilon = 0.01$ is set to avoid a possible zero value of $N_{x,G}$, which is the denominator in Eq. (4.41).

Besides, one strategy is chosen for each individual from the SCPool, according to $SP_{x,G}$. The stochastic universal selection method is used here for strategy selection.

2) Control Parameter Adaptation

Generally, population size NP relies on problem complexity (i.e. the larger the NP, the more robust will the search be, and higher will be the increase in computational cost); MF is often related to the speed of convergence; and CR is usually more problem-sensitive. Accordingly, SaDE adapts these control parameters as follows,

a) Assumes *NP* as a predefined value due to its problem-dependent nature;

b) *MF* for different individuals in a generation are different random values with normal distribution, mean of 0.5 and standard deviation 0.3, denoted as $N(0.5,0.3)$;

c) *CR* of different strategies for each individual in a generation are random values according to $N(CR_m,0.1)$, where CR_m is initialized with 0.5 and is changed with its learning experience via CR-Memories, which are created to store *CR* values of successful strategies and the new CR_m value of each strategy is the mean value of *CR* values in its CR-Memory.

4.5.3 Opposition-based Differential Evolution Algorithm

In this section, an Opposition-based DE algorithm is introduced based on Opposition-based Learning (OBL) method. OBL, introduced by Tizhoosh in 2005 [135], is a new concept in machine intelligence. Generally, evolutionary optimization methods often start with some random assumptions or guesses without any preliminary information on the direction of optimal solutions. The main concept behind OBL is to consider the opposite of an assumption or a guess. That is, by taking into account the opposite of an assumption, and compare it with the original assumption, we can increase our chances to find the solution faster. Of course the concept proposed in [135] was a general, high-level concept which can be utilized in specific optimization algorithms. And it has been proven to be an effective method to DE [136], Particle Swarm Optimization (PSO) [137] and other population-based algorithms in solving many optimization problems.

Opposition-based DE (ODE) is one of successful applications of OBL, which shows faster and more robust convergence than conventional DE by decreasing the distance from an unknown solution by comparing the candidate solution with its opposite and continuing with the better one. It utilizes the scheme of OBL for population initialization and new generation production (generation jumping) to accelerate the convergence rate of conventional DE. The main characteristics of ODE, which are different from conventional DE, are summarized as follows.

1) Definition of Opposite Point:

Let $x \in [a, b]$ be a real number. The opposite number $\breve{x}$ is defined as:

$$\breve{x} = a + b - x \tag{4.43}$$

This definition can be extended to higher dimensions as follows:

Let $P = (x_1, x_2, ..., x_D)$ be a point in a D-dimensional space, where $x_1, x_2, ..., x_D \in R$, $x_i \in [a_i, b_i] \forall i \in \{1, 2, ..., D\}$. Then, the opposite point $\breve{P} = (\breve{x}_1, \breve{x}_2, ..., \breve{x}_D)$ is defined by its components:

$$\breve{x}_i = a_i + b_i - x_i \tag{4.44}$$

When evaluating a solution x to a given problem, simultaneously computing its opposite solution will provide another chance for finding a closer solution to the global optimum. Fig. 4-3 illustrates the one-dimensional of solution x and its opposite solution $\breve{x}$, in [a, b].

2) Opposition-based Population Initialization:

By applying the definition of opposite point, the population initialization can be designed as:

Let $U = (u_1, u_2, ..., u_C)$ be an initial randomly-generated population in a C-dimensional space (i.e. a candidate solution). Then the corresponding opposite population is $\breve{U} = (\breve{u}_1, \breve{u}_2, ..., \breve{u}_n)$. Assume $f(\cdot)$ is the fitness function used to measure the candidate's fitness. If $f(\breve{U})$ is better than $f(U)$, then update population U with $\breve{U}$ as the initial population; otherwise keep the current population U.

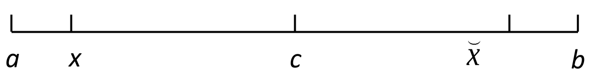


Fig. 4-3. The one-dimensional presentation of solution x and the opposite solution $\breve{x}$, in [a, b], where c is the center of the searching space, i.e., $c = (x + \breve{x})/2$

Hence, the current population and its opposite population are evaluated simultaneously to continue with the fitter one. This scheme obviously accelerates the convergence speed by starting with a better initial candidate solution.

3) Opposition-based Generation Jumping:

Another improvement to conventional DE, by ODE is the generation jumping based on a new checking variable called jumping rate J_r, which is constant value in the entire optimization. After generating a new population V by mutation, crossover and evaluation & selection of DE as stated in Section 4.5.1, the parameter J_r determines the probability of whether the opposite population $\breve{V}$ will be calculated and competed with the original population V. Unlike the opposition-based population initialization, the generation jumping happens dynamically according to the J_r value. I.e., if $rand(0,1) < J_r$, the generation jumping works with opposition-based concept. Besides, instead of using the predefined interval boundaries of the variables, ODE calculates the opposite of each variable based on the minimum ($Min_j^{(G)}$) and maximum ($Max_j^{(G)}$) values of that variable in the current generation G with NP-size population:

$$\breve{U}_{i,j} = Min_j^{(G)} + Max_j^{(G)} - U_{i,j}, \text{ where } i = 1,2,...,NP, \quad j = 1,2,...,C \tag{4.45}$$

This dynamic behavior of the opposite point calculation increases our chance to find fitter opposite points. Because by keeping variables' interval static boundaries, we will jump outside of solution space and the knowledge of current reduced space (converged population) is not utilized to find better opposite candidate.

4.5.4 SaDE-based MCOPF procedure

The flowchart in Fig. 4-4 illustrates the proposed SaDE-based MCOPF method. All contents in dashed boxes are the features that distinguish SaDE from conventional DE.

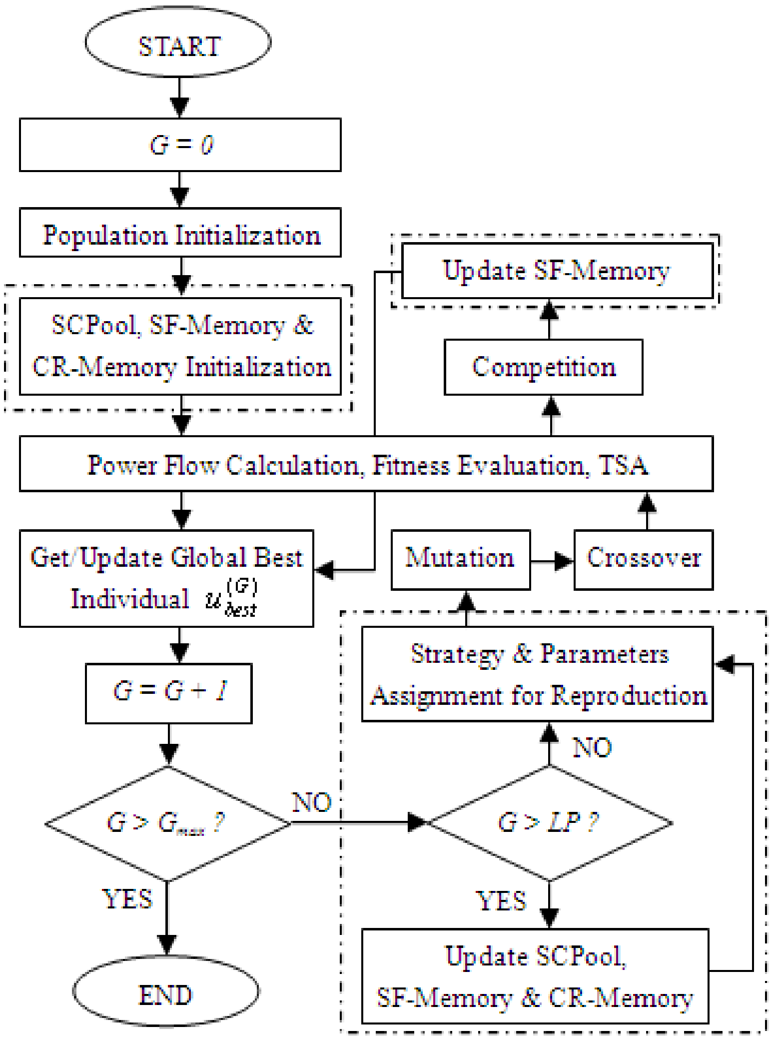


Fig. 4-4. Flowchart of the proposed SaDE-based MCOPF procedure

Specifically, there are mainly five steps for the SaDE-based MCOPF procedure, which are descripted respectively as follows.

1) Population Initialization

When applying SaDE to solve the MCOPF problem, the first step is to determine the control variables embedded in individuals. In our studies, both continuous and discrete control variables introduced in (4.16)-(4.17) are considered respectively and mixed into the control variables set $\mathbf{u}$, i.e. $\boldsymbol{u} = [\boldsymbol{u}_c^T, \boldsymbol{u}_d^T]^T$.

Generally, all control variables in individual *i* are initialized in generation *0*. Handlings of generator active power output control, as described in Section 4.4.2, need to be considered to satisfy the practical operation constraints. For generator active power output, if the initial value lies in POZs, the value must be modified to be the nearest limit of the feasible regions. Moreover, to reduce search space, an individual should be reinitialized if it meets any of the following conditions,

(i) $P_{ds} + P_{slack}^{\max} < P_L$: the candidate is an infeasible solution;

(ii) $P_L + P_{loss} < P_{ds} + P_{slack}^{\min}$: the candidate is bad because of its high power loss. P_{loss} is set to be 10% of PL.

Besides, P_{ds} is the sum of the power which has been dispatched to all generators excluding the slack unit; P_L is the total active load of the system; P_{loss} is the total power loss of the system; $P_{slack}^{\min}$, $P_{slack}^{\max}$ are the lower and upper P limits of the slack bus.

2) SCPool, SF-Memory & CR-Memory Initialization

As discussed in Section 4.5.2, there are mainly two SaDE-related initialization algorithms,

a) Construction of SCPool

Five strategies from [119] are included in the SCPool to achieve diversity of characteristics. Note that all strategies formulated here combine mutation and crossover operations concurrently. Details are shown as follows,

(i) DE/current-to-best/1/bin

$$u'_{i,j} = u_{i,j} + MF(u_{best,j} - u_{i,j}) + MF(u_{r1,j} - u_{r2,j}) \tag{4.46}$$

(ii) DE/rand/1/bin

$$u'_{i,j} = \begin{cases} u_{i,j} + MF(u_{r2,j} - u_{r3,j}), \, if \ rand[0,1) < CR \, or \ j = j_{rand} \\ u_{i,j}, \, otherwise \end{cases} \tag{4.47}$$

(iii) DE/rand-to-best/2/bin

$$u'_{i,j} = \begin{cases} u_{i,j} + MF(u_{best,j} - u_{i,j}) + MF(u_{r1,j} - u_{r2,j}) \\ \quad + MF(u_{r3,j} - u_{r4,j}), \, if \ rand[0,1) < CR \, or \ j = j_{rand} \\ u_{i,j}, \, otherwise \end{cases} \tag{4.48}$$

(iv) DE/rand/2/bin

$$u'_{i,j} = \begin{cases} u_{r1,j} + MF(u_{r2,j} - u_{r3,j}) + MF(u_{r4,j} - u_{r5,j}), \\ \quad if \ rand[0,1) < CR \, or \ j = j_{rand} \\ u_{i,j}, \, otherwise \end{cases} \tag{4.49}$$

(v) DE/current-to-rand/1/bin

$$u'_{i,j} = u_{i,j} + MF(u_{r1,j} - u_{i,j}) + MF(u_{r2,j} - u_{r3,j}) \tag{4.50}$$

The value $SP_{x,G}$ for strategy x in generation G is initialized as $1/X = 1/5 = 0.2$, $x = 1, 2, ..., X$, $G = 1, 2, ..., LP - 1$; after LP generations, the probabilities are updated according to Eq. (4.41).

b) Initializations of SF-Memory & CR-Memory

In this study, a SF-Memory with size $X * LP$ and a CR-Memory with size LP are used. Note that only *CR* values for the *successful strategy* (i.e. the one which helps the offspring defeat the parent) of the current generation will be stored.

3) Power Flow Calculation, Fitness Evaluation & TSA

In SaDE, for each individual a power flow calculation is performed and a fitness value F_i is assigned to measure the quality of the individual. According to Section 4.4.3, violations of feasible regions are considered as punishments in the fitness function (4.31). Also the TSA process is utilized as mentioned in Section 4.4.4.

4) Mutation & Crossover

With mutation and crossover operators, offspring individuals are reproduced successively from their parents based on the updated strategies and parameters. The assignment of strategies and parameters follows the procedures given in Section III. In addition, like the constrained initialization, control variables in offsprings must also be fixed to the margin of the feasible region so as to ensure the feasibility of the problem solution.

5) Competition

In conventional DE, parents and offsprings compete with each other according to their fitness values. Nonetheless, a competition rule is proposed here, along with introduction of violation tolerance settings and transient stability constraints, as shown in Table 4-1. Each competitor acts in accordance with three indices, viz. *feasible* index, *stable* index and fitness value. u_a and u_b are the two competitors, with FI_a, FI_b as their *feasible* indices; SI_a, SI_b as their *stable* indices; and F_a, F_b as their fitness values, respectively. The rule is also applied to updating of the global best individual of Gth generation $u^{(G)}_{best}$.

Table 4-1. Resource Definition of Small Scale Grid System

IF FI_a == FI_b==YES or FI_a==FI_b==NO THEN
IF $SI_a == SI_b$ ==YES or $SI_a == SI_b$ ==NO THEN
IF $F_a > F_b$ THEN u_a is better than u_b
ELSE, u_b is better than u_a
END IF
ELSE IF SI_a ==YES and SI_b ==NO THEN u_a is better than u_b
ELSE, u_b is better than u_a
END IF
ELSE IF FI_a ==YES and FI_b==NO THEN u_a is better than u_b
ELSE, u_b is better than u_a
END IF

4.5.5 ODE-based MCOPF procedure

The flowchart in Fig. 4-5 illustrates the proposed ODE-MCOPF method. All contents in dashed boxes in this flowchart are the features that distinguish ODE from conventional DE.

In this research, ODE-based MCOPF is also proposed based on the conventional DE method. In a nutshell, this approach works as follows:

1) Opposition-based Population Initialization
2) Power Flow Calculation, Fitness Evaluation & TSA
3) Mutation & Crossover
4) Competition
5) Opposition-based Generation Jumping

The first step is to determine the control variables embedded in individuals. By utilizing OBL, it can generate the population and opposite population respectively. The control variables introduced in (4.16)-(4.17) are considered and mixed into the control variables set (i.e. population) $\boldsymbol{u}$.Then the opposite population is $\breve{\boldsymbol{u}}$.

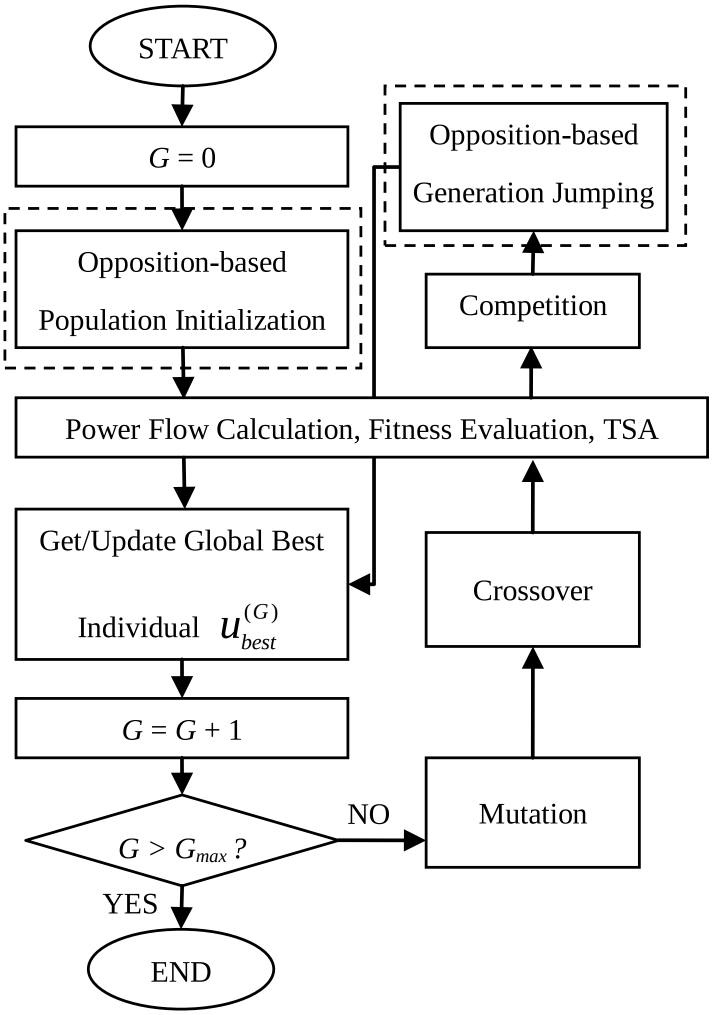


Fig. 4-5. Flowchart of the proposed ODE-based MCOPF procedure

Similarly, all control variables in individual i are firstly initialized in generation *0.* Moreover, special handlings on generator active power output control is considered for the practical operation constraints, as mentioned in Section 4.5.4. The following steps present the procedure of opposition-based population initialization.

(a) Initialize the population $\boldsymbol{u}$ randomly.

(b) Calculate the opposite population $\breve{\boldsymbol{u}}$.

(c) Select the NP-size fittest individuals from $\{\boldsymbol{u} \cup \breve{\boldsymbol{u}}\}$ as the final initial population.

The procedure of power flow calculation, fitness evaluation, TSA, mutation, crossover and competition in Step 2) – 4) are the same as the ones of SaDE-based MCOPF method, which are proposed in Section 4.5.4. It is to be noted that an opposition-based generation jumping step is introduced here. Based on a jumping rate J_r, instead of generating new population by evolutionary process, the opposite population is calculated and the *NP*-size fittest individuals are selected from the current population and the corresponding opposite population (exactly it is similar to what was performed for opposition-based population initialization), as stated in Section 4.5.3. Note that the comprehensive experiments show that J_r should be a small number ($J_r \in (0, 0.4)$).

4.6 Applying Cloud Computing Techniques for MCOPF Problem

4.6.1 Cloud Computing Platform

According to the mechanism of SaDE algorithm, it is intrinsically easy to be parallelized. In order to enhance the performance for large-scale system, the proposed MCOPF problem could be deployed on parallel computing equipment, such as multi-core server or PC cluster. However, the investment & maintenance costs of such equipment are often high, and for the researchers who do not own such equipment, it is difficult to perform such parallel computing. This project harnesses the Cloud Computing technology to parallelize and deploy SaDE to solve large-scale, complicated systems in a high performance but low cost pattern. The details of Cloud Computing can be referred to [97,98] while generally speaking, Cloud Computing provides a virtualized resource pool where users can apply the IT resources through Internet to perform tasks as they need and pay for the amount of resources they consumed. Amazon Elastic Cloud (Amazon EC2) [103] is utilized here, which is one of the largest Cloud Computing service providers. With Amazon EC2, users can operate the virtualized servers or clusters through Internet just like the local resources. The deployment model of the parallelized MCOPF on Amazon EC2 is shown in Fig. 4-6. In this research, a simple master-slave parallel scheme is used, where the master lunches different subpopulations on multiple virtualized machines and each subpopulation runs separately and sends the final result back to the master.

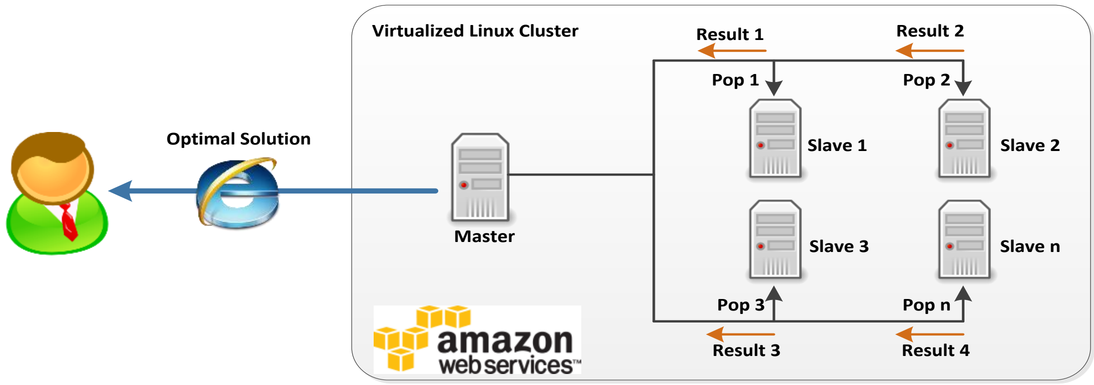


Fig. 4-6. Cloud Computing deployment model of SaDE-based MCOPF

4.6.2 Parallelized Optimization of MCOPF

The studies in Section 4.5 show that DE and the improved SaDE and ODE algorithms are effective and robust in solving the MCOPF problem. However, transient stability assessment is a CPU intensive task especially when system becoming large. Fortunately, the development in the world of modern computing technologies offers the possible way of parallel computing to decrease the computation time of engineering problem and meet the need of practical application.

As one of the evolutionary algorithms, DE is intrinsically a parallel searching algorithm. Thus, it is very suitable for parallel computation. So far, researches about parallel evolutionary algorithm on OPF problem are inadequate. This research parallelizes the DE based MCOPF problem on the proposed Cloud Computing platform.

4.7 Comparison Case Studies

The proposed MCOPF problem becomes nonlinear, non-convex, discrete, and non-differentiable, thus may degrade the quality of solution and convergence rate. To verify and compare the effectiveness of the proposed SaDE-based MCOPF method and ODE-based MCOPF method, the New England 10-generator, 39-bus system and IEEE 17-generator, 162-bus system are tested respectively. In these cases, the proposed two methods are also compared with other recently published methods used in OPF problems, including conventional mathematical method – Sequential Quadratic Programming (SQP, in

MATPOWER software) [138], and modern heuristic methods – DE [69], EGA [139], EP [140] and PSO [141], in terms of solution quality and computation efficiency using the same fitness function and individual definition described in Section 4.5.4 and Section 4.5.5. SQP is implemented in Matlab using MATPOWER toolbox, and other methods are coded with C Language. A total of 20 trial runs of all the heuristic algorithms are performed for each method on a computer with an Intel Core 2 CPU, 2 GB RAM.

4.7.1 Case A: New England 10-generator, 39-bus system

The system data are available in [142] and the single line diagram of this system is shown in Fig. 4-7. The lower and upper limits of all bus voltage magnitudes are 0.94 p.u. and 1.06 p.u. respectively. The unit ratings and fuel cost coefficients with valve-point effects are listed in Table 4-2. Lower and upper limits for transformer tap settings are 0.90 and 1.10 p.u., respectively, and the adjustment is in steps of 0.01 p.u. There are 31 control variables, comprising 9 generator active power outputs, 10 generator voltage magnitudes and 12 transformer tap settings. A three-phase to ground fault at Bus 21 is cleared by tripping Line 21-22 at 0.16 s. Transient simulation time step is set as 0.01 s, and the whole simulation period is 3.0 s. With the same control variable limits, initial conditions and system data, the parameters are given in Table 4-3.

Table 4-2. Unit Ratings and Fuel Cost Coefficients with Valve-point Effects for New England 10-Generator, 39-Bus System

Unit	P_{min}	P_{max}	a	b	c	w	e
30	0	350	0.0193	6.90	0.0	100	0.084
31	0	650	0.0111	3.70	0.0	150	0.063
32	0	800	0.0104	2.80	0.0	200	0.042
33	0	750	0.0088	4.70	0.0	150	0.063
34	0	650	0.0128	2.80	0.0	150	0.063
35	0	750	0.0094	3.70	0.0	150	0.063
36	0	750	0.0099	4.80	0.0	150	0.063
37	0	700	0.0113	3.60	0.0	150	0.063
38	0	900	0.0071	3.70	0.0	200	0.042
39	0	1200	0.0064	3.90	0.0	250	0.036

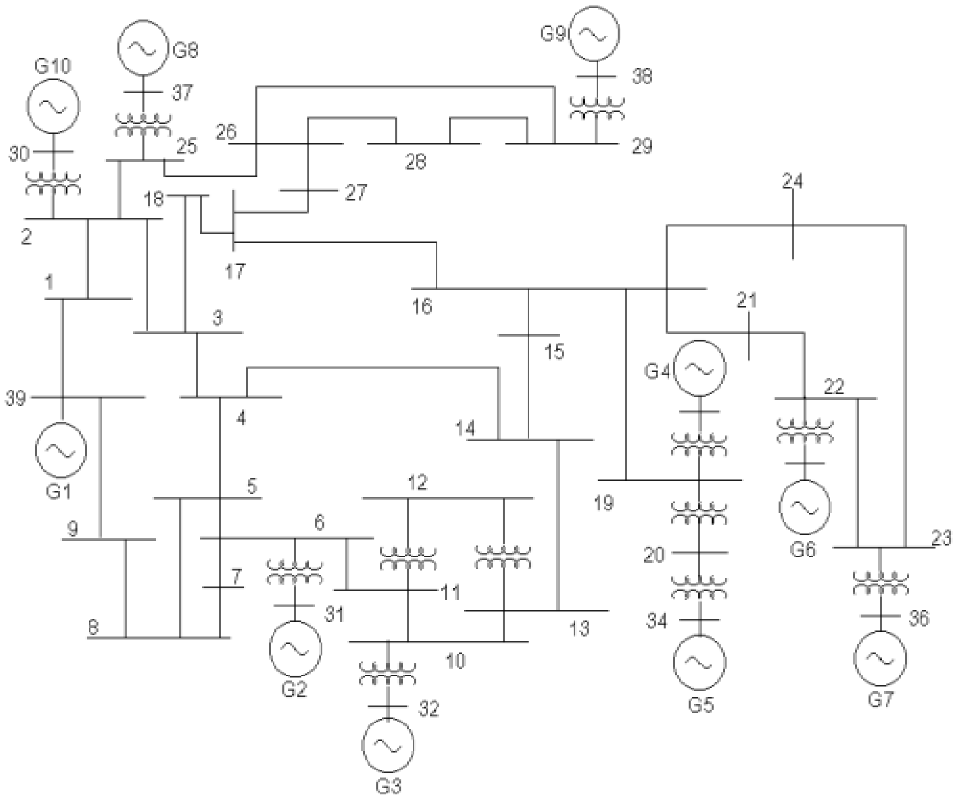


Fig. 4-7. The single line diagram of New England 10-generator, 39-bus test system.

Table 4-3. Parameters of EGA, EP, PSO, DE, SaDE & ODE

EGA	***EP***	***PSO***
NI = 31	NI = 31	NI = 31
NP = 60	NP = 60	NP = 60
CF = 0.6	G_{max} = 100	G_{max} = 100
MF = 0.05	*Scaling factor* = 0.1	*Inertia weight* = 1.2
G_{max} = 100		*Acceleration constants* = 2.5
DE	***SaDE***	***ODE***
NI = 31	NI = 31	NI = 31
NP = 60	NP = 60	NP = 60
CF = 0.7	$CF = N(CR_m,0.1)$,	CF = 0.7
MF = 0.7	$MF = N(0,5,0.3)$	MF = 0.7
G_{max} = 100	G_{max} = 100	G_{max} = 100
	LP = 40	Jr = 0.3

In this section, two cases are simulated. The former is the basic OPF, which just considers the discrete control variables; while the latter is the full MCOPF model proposed in this research. Detailed results and analyze are presented as follows.

1) Case A-1: Basic-OPF with Discrete Control Variables

Before the detailed study on MCOPF, a basic-OPF model is formulated and studied, in which the valve-point effects are not included in the objective function Eq. (4.19). Equality and inequality constraints include the load flow equation in Eq. (4.20) and the lower and upper limits of continuous and discrete control variables in Eqs. (4.22)-(4.25), respectively. Constraints for transient stability in Eq. (4.15), POZs of generators in Eq. (4.21) and transmission line flows in Eq. (4.26) are not considered.

In this basic-OPF model, the classic SQP method is first applied to solve the problem by considering the discrete variables, i.e. transformer tap control variables, as continuous variables, and the final optimal solution is obtained by rounding off the solution of QP method. Then, the same basic-OPF model is performed using EGA, EP, PSO, conventional DE, SaDE and ODE methods, as shown in Table 4-4. It can be seen that considering discrete control variables, solutions from heuristic methods are significantly more economical than that of the the classic SQP method, obtained in similar computational CPU time. All the methods can satisfy all the constraints in the basic-OPF model (*Feasible* is “Yes”), except SQP. The power flow result obtained by the SQP method shows that voltages of Buses 25 and 36 (1.062 p.u. and 1.063 p.u., respectively) violated the upper limit (1.06 p.u.), while with other heuristic methods, all constraints are satisfied. The above simulation shows not only the superiority of performance of heuristic methods over the classic mathematical method SQP, but also the importance of a solution approach, which can flexibly consider the detailed model of the problem in the optimization. The simulation verified that with the classic optimization method, the practical operational constraints cannot be satisfied indeed and has to be either ignored or simplified. The rest of the comparison studies are focused on heuristic methods.

Clearly SaDE is more capable of handling discrete control variables and it is better than other methods in terms of minimum, maximum and average fuel costs. Standard deviation (SD) of SaDE trials is also smaller. For similar computational time, SaDE has better convergence rate toward an acceptable solution than others. The best solutions obtained with SaDE are tabulated in Table 4-5 and the average convergence curves over 20 trials of EGA, EP, PSO, DE, SaDE and ODE are shown in Fig. 4-8.

Table 4-4. Comparison Results among Seven Methods in Case A-1

Method	***Evaluation Value ($/h)***			***CPU Time (s)***	***Performance***	
	Avg.	***Min.***	***Max.***		***SD(%)***	***Feasible***
SQP	---	63009	---	2.7	---	No
EGA	61302	61248	61350	3.8	0.049	Yes
EP	61003	60945	61075	4.0	0.041	Yes
PSO	60965	60935	60985	3.9	0.032	Yes
DE	60923	60909	60944	3.7	0.015	Yes
SaDE	60912	60904	60925	4.2	0.011	Yes
ODE	60920	60906	60933	5.1	0.014	Yes

Table 4-5. Best System Solution of Case A-1 Using SaDE

Unit	**Generation (MW)**	**Voltage (p.u.)**	**Branch**	**Tap Setting (p.u.)**
30	243.00	1.0253	12-11	1.03
31	571.93	1.0600	12-13	1.02
32	641.33	1.0035	6-1	1.10
33	631.13	1.0161	10-32	1.08
34	503.41	1.0431	19-33	1.06
35	649.47	1.0483	20-34	0.95
36	555.89	1.0390	22-35	1.03
37	533.42	1.0438	23-36	1.04
38	829.87	1.0411	25-37	1.02
39	978.95	1.0292	2-30	1.01
Fuel Cost ($/h): 60904			29-38	1.02
			19-20	1.08

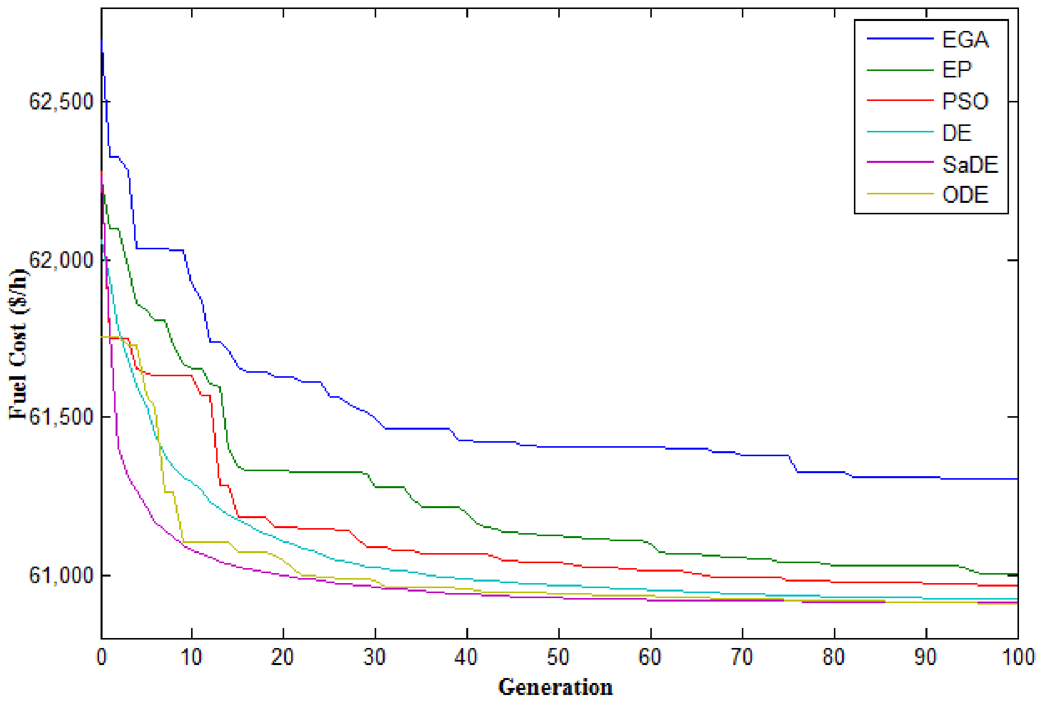


Fig. 4-8. Average convergence curves of six heuristic methods for Case A-1.

2) Case A-2: MCOPF

The full MCOPF model formulated in Section 4.5 is studied here. Three units are selected to have POZs. The data of generator's POZs are listed in Table 4-6. Branch flow apparent power limit is set at 800MVA for each line; Branch 29-38 has a flow of 852.61MVA, which violates the limit.

Table 4-6. Unit POZs Data for New England System

Unit	P_i^{min} (MW)	P_i^{max} (MW)	POZs (MW)
30	0	350	[200, 300]
32	0	800	[400, 500] [600, 700]
39	0	1200	[700, 800] [900, 1100]

Critical clearing time (CCT) is used for transient stability constraints. CCT of the test system under the original system configuration is 144ms which is less than the fault clearing time (FCT, 160ms), and then it is unstable. For the trade-off of fuel cost and computational time, only 40% of individuals are randomly selected to TSA.

Simulation results are summarized in Table 4-7 and Fig. 4-9, and best solution of SaDE is listed in Table 4-8.

Table 4-7. Comparison Results among Five Methods in Case A-2

Method	*Evaluation Value ($/h)*			*CPU Time(s)*	*Performance*			
	Avg.	*Min.*	*Max.*		*SD (%)*	*Feasible*	*Stable*	*Avg. CCT (ms)*
EGA	62785	62503	62891	58.5	0.59	Yes	Yes	175.5
EP	62692	62489	62880	55.0	0.51	Yes	Yes	185.0
PSO	62558	62268	62837	58.9	0.34	Yes	Yes	179.8
DE	62260	62047	62565	57.4	0.20	Yes	Yes	187.1
SaDE	61505	61393	61724	60.3	0.12	Yes	Yes	188.2
ODE	61787	61565	61902	71.2	0.15	Yes	Yes	186.1

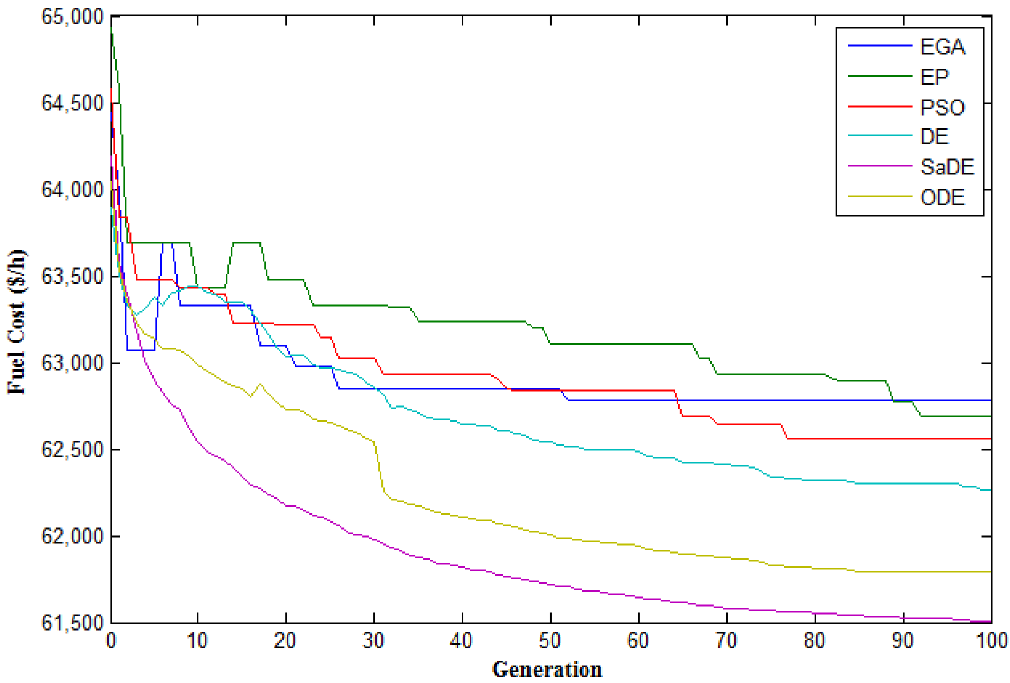


Fig. 4-9. Average convergence curves of five methods for Case A-2

Table 4-8. Best System Solution of Case A-2 Using SaDE

Unit	Generation (MW)	Voltage (p.u.)	Branch	Tap Setting (p.u.)
30	300.10	0.9932	12-11	0.97
31	549.72	0.9998	12-13	0.95
32	599.76	1.0342	6-1	1.09
33	648.74	1.0206	10-32	1.06
34	498.75	1.0086	19-33	1.05
35	599.31	1.0224	20-34	1.07
36	548.77	1.0046	22-35	1.06
37	504.18	0.9847	23-36	1.04
38	749.39	1.0308	25-37	1.06
39	1135.13	1.0128	2-30	1.04
Fuel Cost ($/h):	61393		29-38	1.02
CCT (ms):	177		19-20	1.00

As seen from the results, generation power solutions of units with POZs have moved away from the zones to feasible regions. Branch flow in Branch 29-38 has been decreased to 765.67MVA without any other branch flow violations. The average fuel cost is slightly higher than that in Case A-1, which is outside the practical operations in this case. Although with the simplified problem model, the fuel cost seems lower, a trade-off between economics and security should be considered in reality. According to CCT, feasible and stable values (in Table 4-7), it is obvious that all these methods can operate the system in a stable and economical condition. However, SaDE achieves a smaller standard deviation and performs more robustly. The study also reveals that in some generations, although the fuel cost of individual *i* is lower, constraints violations may lead to fitness value F_i being worse (higher) than others. Besides, comparing with SaDE and ODE methods, there are more constraints violations that result in fluctuation of the convergence in EGA, EP, PSO and DE simulations. The self-adaptation of SaDE handles this problem well, as well as the opposition learning ability of ODE.

4.7.2 Case B: Cloud Computing of SaDE-based MCOPF on IEEE 17-generator, 162-bus system

An 8-core high performance cluster computing instance is created in Amazon EC2 over the IEEE 17-generator, 162-bus system here, to parallelize SaDE, ODE, as well as the other five heuristic algorithms respectively for the performance comparison. System data is available in [143]. The POZs, unit ratings and fuel cost coefficients data are shown in Table 4-9 and Table 4-10, respectively. 10 transformers are selected as adjusting transformers, with changes in steps of 0.10 p.u. There are 43 control variables: 16 generator active power outputs, 17 generator voltages and 10 transformer tap settings. For this large system, set NP = 120, G_{max} = 200, leaving other control parameters the same as in the New England system. A three-phase to ground fault at Bus 1 is applied and cleared by tripping Line 1-4 at 240ms (i.e. FCT=240ms here), which is greater than the initial CCT 220ms, so the original state is unstable.

Table 4-9. Unit POZs Data for IEEE 17-Generator, 162-Bus System

Unit	$P_i^{\min}$ **(MW)**	$P_i^{\max}$ **(MW)**	**POZs (MW)**
3	1000	2300	[1250,1350] [1700,1800] [2100,2200]
15	1000	1800	[1200,1300] [1550,1650]
76	500	1355	[700,800] [900,1100]
118	0	473	[200,300]
131	200	875	[400,500]

All individuals are involved in TSA calculation in this simulation; results are shown in Table 4-11 and Table 4-12. All the methods can make the system stable after the contingency; but SaDE incurs lower cost and standard deviation, and has prominent convergence speed. The average cost reduction with SaDE is 10.72% in contrast with the original system operating state (31339.34 $/h).

Table 4-10. Unit Ratings & Fuel Cost Coefficients with Valve-point Effects for IEEE 17-Generator, 162-Bus System

Unit	P_{min}	P_{max}	*a*	*b*	*c*	*w*	*e*
3	1000	2300	0.00064	0.50	0.0	200	0.042
6	500	1094	0.00098	0.30	0.0	100	0.084
15	1000	1800	0.00076	0.50	0.0	120	0.077
27	1000	1800	0.00076	0.50	0.0	120	0.077
73	200	747	0.00150	0.20	0.0	100	0.084
76	500	1355	0.00088	0.30	0.0	150	0.063
99	0	450	0.00200	0.40	0.0	80	0.098
101	0	382	0.00200	0.40	0.0	80	0.098
108	0	1200	0.00084	0.30	0.0	200	0.042
114	0	431	0.00200	0.40	0.0	80	0.098
118	0	473	0.00200	0.40	0.0	80	0.098
121	200	920	0.00150	0.30	0.0	100	0.084
124	1000	2851	0.00640	0.52	0.0	300	0.035
125	1000	2688	0.00640	0.67	0.0	300	0.035
126	1000	2767	0.00640	0.42	0.0	300	0.035
130	200	755	0.00150	0.30	0.0	100	0.084
131	200	875	0.00150	0.30	0.0	100	0.084

Table 4-11. Simulation Results of IEEE 17-Generator, 162-Bus System

Method	***Evaluation Value ($/h)***			***CPU Time (s)***	***Performance***			
	Avg.	***Min.***	***Max.***		***SD (%)***	***Feasible***	***Stable***	***Avg. CCT (ms)***
EGA	29762	29163	30389	1334	6.12	Yes	Yes	261
EP	29850	29164	30498	1335	4.76	Yes	Yes	259
PSO	29577	29122	30150	1342	3.02	Yes	Yes	256
DE	29348	28679	30190	1325	1.51	Yes	Yes	259
SaDE	27981	27685	28326	1328	0.54	Yes	Yes	256
ODE	28322	28015	28615	1390	0.63	Yes	Yes	256

Table 4-12. Best Solution of IEEE 162-Bus System Using SaDE

Unit	Generation (MW)	Voltage (p.u.)	Branch	Tap Setting (p.u.)
3	2048.01	0.9821	108-6	0.98
6	687.58	1.0188	12-2	0.98
15	1652.02	0.9709	26-76	0.95
27	1448.87	1.0103	66-11	0.98
73	611.35	0.9691	96-101	1.10
76	699.30	0.9961	128-72	0.96
99	385.04	1.0156	144-141	1.06
101	224.71	1.0080	153-70	0.97
108	748.08	0.9901	110-112	0.92
114	224.62	1.0452	93-42	0.94
118	417.14	0.9735		
121	611.80	1.0019		
124	2077.08	0.9786	**Fuel Cost ($/h)**: 27685.56	
125	2256.71	0.9418		
126	2166.70	1.0571	**CCT (ms)**: 251	
130	611.26	0.9789		
131	573.84	0.9912		

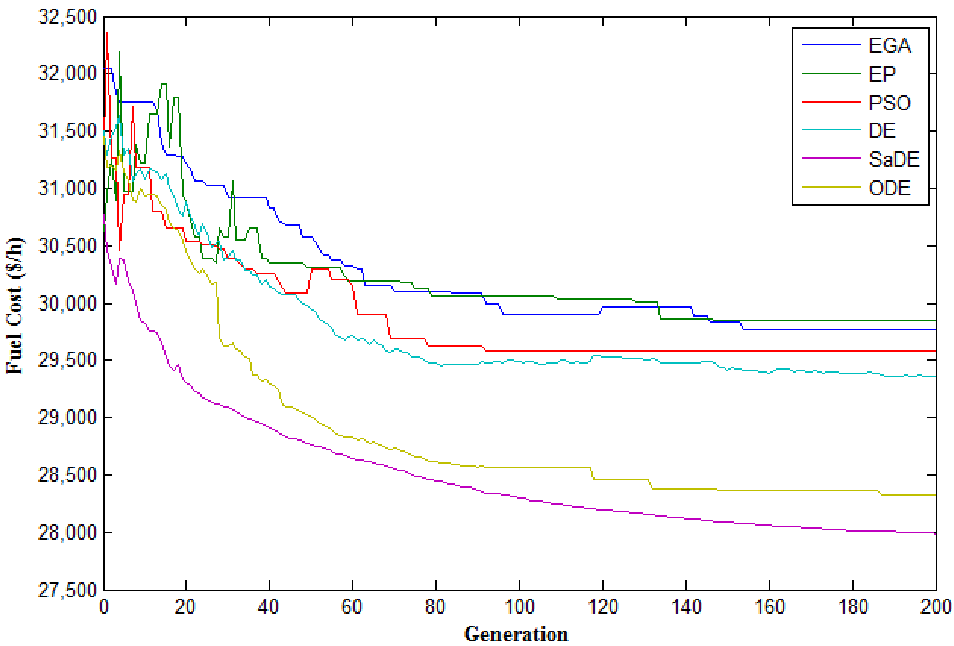


Fig. 4-10. Average convergence curves of IEEE 17-Generator, 162-Bus system

Obviously, in large-scale systems, the computational time costs of all the methods are almost same, and the longer computational time of SaDE's self-adaptation can be ignored. And according to Fig. 4-10, SaDE gets the final solution of other methods in 20th generation (around 90% time-saving). And the cost of performing above cloud computing simulations over Amazon EC2 is just $1.3/hour.

4.8 Summary

In this chapter, a computational framework and tools have been developed for a specific OPF system operations and planning task. The framework is based on multi-constrained optimal power flow (MCOPF), which explores the computational power from the improved DE algorithms to achieve fast yet reliable optimisation computations. Transient stability constraints, constraints due to generator POZs, valve-point effects, and branch flow thermal constraints are all included in the MCOPF calculations. Simulation results show that compared to the conventional mathematical method, SQP, and some modern heuristic methods such as EGA, EP, PSO, conventional DE and ODE, SaDE is very effective and robust. Its superior advantages are most significant when the problem becomes acute. Within the simulation results of cloud computing for large-scale power systems, it is shown that the proposed framework can be easily paralleled and has a large potential to be applied to real-world large-scale MCOPF problems.

Chapter 5

COLLECTOR SYSTEM LAYOUT OPTIMIZATION FOR OFFSHORE WIND FARMS

5.1 Introduction

With the fundamental operational planning tools, as described in Chapter 4, this chapter outlines the research for a new and efficient collector system layout optimisation (CSLO) model for large-scale offshore wind farms, which considers multiple substations and cable types, and focuses on cable topology optimisation among wind turbines and substation(s) with the objective to minimise the overall investment and maintenance costs, as well as the levelized power losses cost, while simultaneously considering the network reliability and operational constraints.

5.2 Problem Description and Formulation

5.2.1 Collector System Layout of Offshore Wind Farms

Each possible alternative connection scheme between substations and wind turbines produces a new layout to which cables must be assigned to them that minimize the total cost for a specific planning schedule and comply with the technical constraints. Therefore, this CSLO problem is formulated as an optimization problem in (5.1)-(5.11). The primary objective function (5.1) is composed of annualized investment cost (5.2), maintenance cost (5.3), and the levelized power losses cost (5.4) and, at the same time, to meet all the security constraints imposed on the collector system for reliability purposes, which are listed below including the active power flow constraints (5.5), reactive power flow constraints (5.6), bus voltage limits (5.7)-(5.8), substation capability limits (5.9), cable capability limits (5.10), and network radiality constraints (5.11).

Objective:

$$Min.\ F = C_1\left(\boldsymbol{I}\right) + C_2\left(\boldsymbol{I}\right) + C_3\left(\boldsymbol{I}\right) \tag{5.1}$$

$$C_1\left(\boldsymbol{I}\right) = \sum_{c\in\Psi^C} \sum_{i\in\Psi^N} \sum_{j\in\Psi^N} c_c^I l_{ij} I_{ij,c} \tag{5.2}$$

$$C_2\left(\boldsymbol{I}\right) = kC_1\left(\boldsymbol{I}\right) \tag{5.3}$$

$$C_3(\boldsymbol{I}) = \sum_{h\in\Psi^H}\sum_{c\in\Psi^C}\sum_{i\in\Psi^N}\sum_{j\in\Psi^N} c^L P_{ij,h}^L I_{ij,c} \tag{5.4}$$

Subject to:

$$P_i^g = V_i \sum_{j\in\Psi^N} V_j\left[G_{ij}\cos\left(\delta_i-\delta_j\right)+B_{ij}\sin\left(\delta_i-\delta_j\right)\right], \forall i\in\Psi^N \tag{5.5}$$

$$Q_i^g = V_i \sum_{j\in\Psi^N} V_j\left[G_{ij}\sin\left(\delta_i-\delta_j\right)-B_{ij}\cos\left(\delta_i-\delta_j\right)\right], \forall i\in\Psi^N \tag{5.6}$$

$$V_i^{\min} \le V_i \le V_i^{\max}, \forall i\in\Psi^N \tag{5.7}$$

$$\delta_i^{\min} \le \delta_i \le \delta_i^{\max}, \forall i\in\Psi^N \tag{5.8}$$

$$\left|S_{ij}\right| \le \sum_{\mathrm{c}\in\Psi^C} I_{ij,c} S_c^{\max}, \forall i,j\in\Psi^N \tag{5.9}$$

$$SN_i \le SN_i^{\max}, \forall i\in\Psi^S \tag{5.10}$$

$$\sum_{c\in\Psi^C}\sum_{i\in\Psi^N, i\ne j} I_{ij,c} \le 1,\ \forall j\in\Psi^N \tag{5.11}$$

where, c is the index of cable types; i and j are the index of buses; h is the index of time period; c_c^I is the investment cost of cable with type c, which includes cable purchasing cost and installation cost; l_{ij} is the cable length between bus i and bus j; C_1 is the overall investment cost; C_2 is the overall maintenance cost; C_3 is the overall levelized power losses cost; $I_{ij,c}$ is the binary decision variables, $I_{ij,c}=1$ if cable with type c is connected between bus i and bus j, $I_{ij,c}=0$ otherwise; I is the set of binary decision variables $I_{ij,c}$; k is the coefficient for maintenance cost; c^L is the levelized cable losses cost; $P_{ij,h}^L$ is the cable losses between bus i and bus j in time period h; P_i^g is the active power output of wind turbine at bus i; Q_i^g is the reactive power output of wind turbine at bus i; S_{ij} is the apparent power flow in the cable between bus i and bus j; V_i is the voltage magnitude at bus i; δ_i is the voltage angle at bus i; G_{ij} is the real components of element i, j in bus admittance matrix; B_{ij} is the imaginary components of element i, j in bus admittance matrix; SN_s is the number of turbines in substation s; $V_i^{\min}$ is the lower bound on the voltage magnitude at bus i, $V_i^{\max}$ is the upper bound on the voltage magnitude at bus i, $\delta_i^{\min}$

is the lower bound on the voltage angle at bus *i*, $\delta_i^{\max}$ is the upper bound on the voltage angle at bus *i*, $S_c^{\max}$ is the maximum allowable apparent power flow at cable type *c*.

Note that for offshore wind farms, the submarine cables are utilized whose capacitances are much higher compared to the overhead cables. The effect of reactive power on bus voltage cannot be neglected. Therefore, the AC power flow equations are used in this research to calculate the network security.

5.3 Self-Adaptive Allocation Method

Considering multiple substations in the proposed CSLO model, this section presents a self-adaptive allocation (SAA) method for wind turbines in large-scale offshore wind farms.

5.3.1 Wind Turbine Clustering

In a large and complex engineering system, clustering is an effective way to reduce the complexity of model. It is the process of grouping a data set in a way that the similarity between data within a cluster is maximized while the similarity between data of different clusters is minimized [144]. There have already been some studies considering the clustering technique as a tool in power systems [145-148]. For example, a clustering method proposed in [145] grouped wind turbines based on their power output dynamics to simplify controller design, operation and forecast of wind farm output. However, to the best of our knowledge, there has no research and application of clustering technology for offshore wind farm collector system layout. Normally, there are two popular approaches to clustering: crisp clustering (or hard clustering), and fuzzy clustering [149]. In the crisp clustering method, the boundary between clusters is fully defined. However, in many practical cases, boundaries between natural classes may be overlapping. So, certain input patterns do not completely belong to a single class, but partially belong to the other classes too. In such cases, the fuzzy c-means (FCM) clustering method provides a better and more useful method to classify these patterns. FCM is a data clustering method in which a dataset is grouped into *N* pre-specified clusters, with every data point in the dataset belonging to every cluster to a certain degree [150]. By iteratively updating the cluster centres as well as the membership grades for each data point, FCM moves the cluster centres to the right location within the dataset. In this research, a self-adaptive allocation (SAA) method for wind turbines is proposed in the CSLO model, which optimizes the locations of all substations and then groups wind turbines to their nearest substations based on FCM clustering method.

Fuzzy C-means clustering algorithm (FCM) is one of the most important and popular fuzzy clustering algorithms. It allows one piece of data point to belong to one or more cluster centres. This algorithm is frequently utilized in pattern recognition [151,152]. For some clustering methods, including FCM, the number of clusters k needs to be given in advance. By iteratively updating the cluster centres as well as the membership grades for each data point, FCM moves the cluster centres to the right location within the dataset.

Suppose a collection of data points $Y=\{y_1,y_2,...,y_n\}$, *n* is the number of elements in the dataset. The set is divided into *k* clusters $1<k<n$, and $C=\{c_1,c_2,...,c_k\}$ is the cluster center matrix. The membership matrix indicating the subordination degree of *i*th element to *j*th element is $U=\{u_{ij}\,|\,i\in[1,n],\, j\in[1,k]\}$.

The FCM algorithm performs clustering by solving,

$$Min.\ \ J_m(U,C)=\sum_{i=1}^{n}\sum_{j=1}^{k}u_{ij}^{m}\left\|\boldsymbol{y}_i-c_j\right\|^2 \tag{5.12}$$

$$s.t.\ \ u_{ij}\subset[0,1] \tag{5.13}$$

$$\sum_{j=1}^{k}u_{ij}=1 \tag{5.14}$$

$$0<\sum_{i=1}^{n}u_{ij}<n \tag{5.15}$$

where, u_{ij} is the membership degree; *m* is the weighted index; J_m is the center of each data point to the class and the square of the Euclidean distance.

The algorithm is composed of the following steps:

Step-1: Give the number of clusters *k*, the weighted index *m*, and the iterative standard ε.

Step-2: Fix the membership matrix *U* with a random number between 0 and 1. Set the iteration number $t = 1$.

Step-3: Calculate the cluster centre *C*.

$$c_i(t) = \left[\sum_{j=1}^{m} u_{ij}^{m}(t) y_i(t) \right] / \left[\sum_{j=1}^{m} u_{ij}^{m}(t) \right] \tag{5.16}$$

Step-4: Calculate the new membership matrix *U*.

$$u_{ij}(t) = \sum_{i=1}^{k} \left[d_{ij}(t-1) / d_{rj}(t-1) \right]^{2(m-1)} \tag{5.17}$$

where $d_{ij} = \left\| \boldsymbol{y}_i - \boldsymbol{c}_j \right\|$ is the Euclidean distance between y_i and $\boldsymbol{c}_j$.

Step-5: Calculate the objective function J_m. If the maximum of iteration is reached or tolerance $\left\| u_{ij}(t) - u_{ij}(t-1) \right\| \le \varepsilon$ is met, then stop; otherwise, return to step-3.

It can be seen that the entire optimization process is to revise the cluster centers and the membership degrees repeatedly until satisfying the stop criterion. The FCM algorithm has been collected in the fuzzy logic toolbox in Matlab. In this research, based on FCM, a fuzzy clustering approach is adopted to automatically allocate N_{WT} wind turbines to N_S substations, where N_{WT} is the number of substations, N_S is the number of wind turbines, and their Cartesian coordinate locations in an offshore wind farm are also pre-specified.

5.3.2 Wind Turbine Allocation

Based on FCM, all the wind turbines could be allocated to their nearest substations respectively. However, with the aim of considering network operation reliability constraints, the capacity of connected turbines in one substation is limited, and each turbine should belong to a substation exclusively. Therefore, in this phase, a maximum problem with binary linear constraints is developed. The objective function here is to maximize the overall membership degree calculated from FCM and reallocate turbines to their nearest substations while satisfying the capacity limits.

With the above considerations, the model for finding the optimal turbine allocation could be formulated as:

$$Max.\ \sum_{i=1}^{Nwt}\sum_{j=1}^{Ns} u_{ij} z_{ij} \tag{5.18}$$

$$s.t.\ z_{ij} \subset [0,1] \tag{5.19}$$

$$\sum_{i\in\Psi^{WT}} z_{ij} = SN_j^{\max}, \quad \forall j \in \Psi^S \tag{5.20}$$

$$\sum_{j\in\Psi^{S}} z_{ij} = 1, \quad \forall i \in \Psi^{WT} \tag{5.21}$$

This problem could be regarded as a BIP problem, which is a special case of Linear Programming (LP) where the variables have only 0-1 binary integer values. It is NP-hard and can be solved by methods like Branch and Bound (BNB) and Gomory Cut algorithms [153]. In this research, the BIP is solved with BNB method.

5.4 CSLO using Minimum Spanning Tree Algorithm

5.4.1 Introduction of Minimum Spanning Tree Algorithm

To compare with the performance of the proposed methods, a minimum spanning tree (MST) algorithm is also introduced here. Considering the radiality constraints for the network topology, the collector system layout problem for an offshore wind farm can be considered as finding a tree to meet required design characteristics in a graph $G=(V,E)$, where V is the set of vertices and E is the set of edges. MST constructs a tree of minimum total length between specified nodes, where the tree is defined as a graph G with one and only one path between every two nodes. Therefore, MST algorithm can be used to address the problem of finding the minimum total length layout design for cable connection in a wind farm, where the minimum total length gives the minimum total investment for cable connection which relies on the cable length, and the wind turbines' locations are the nodes.

There are three commonly used MST algorithms, which are Boruvka's algorithm, Kruskal's algorithm, and Prim's algorithm [154,155]. In this research, Prim's algorithm is utilized to find the optimal layout, which is a greedy algorithm that finds a minimum spanning tree for a connected weighted undirected graph. It finds a subset of the edges that forms a tree that includes every vertex, where the total weight of all the edges in the tree is minimized. This algorithm works by attaching a new edge to a single growing tree at each step: start with any vertex (arbitrary one) as a single-vertex tree; then add V-1 edges to it, always taking next minimum-weight edge that connects a vertex on the tree to a vertex not yet on the tree. The process is repeated until all the vertices are included in the spanning tree. After that, the transmission power losses can be obtained according to the optimal cable layout solution.

5.4.2 *Optimization Procedure*

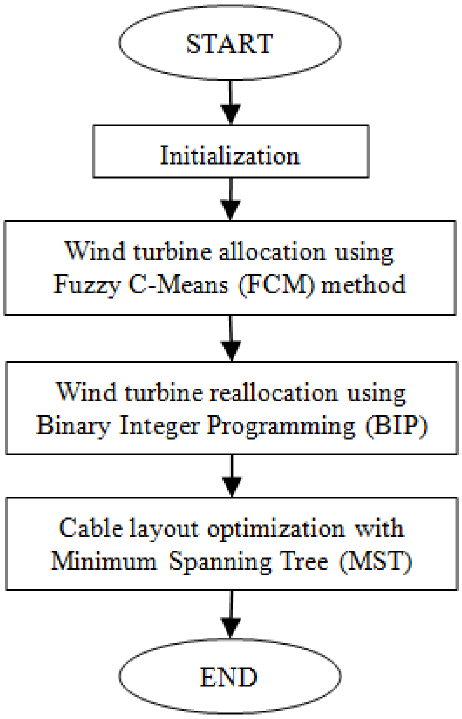


Fig. 5-1. The proposed method for the optimal design of electrical layout

The flowchart in Fig. 5-1 illustrates the proposed method for the optimal design of electrical layout for an offshore wind farm.

Specifically, the procedure could be presented as follows:

1) Initialization;
2) Wind turbine allocation using fuzzy c-means (FCM) method;
3) Wind turbine reallocation using binary integer programming (BIP) algorithm;
4) Cable layout optimization with the MST algorithm.

5.5 Benders Decomposition Method for the Collector System Layout of Large-Scale Offshore Wind Farms

5.5.1 Introduction of Benders Decomposition

Benders decomposition is a popular optimization technique. Using the Benders decomposition method, an original large-scale optimization problem could be decomposed into a master problem and several slave subproblems, which are then iteratively solved by generation of dual variables from the subproblem to form the benders cut and add it into the master problem until the stopping criteria are met. In general, the master problem is an integer program, and subproblems are linear or non-linear programs with real variables.

5.5.2 Benders Reformulation of the Proposed Optimizer

The CSLO problem addressed in this research is formulated as a mixed-integer nonlinear programming (MINLP) problem, with a non-linear objective function, binary decision variables, continuous variables, linear and non-linear constraints such as power flow equations and transmission line capacity limits. Small instances of this MINLP problem can be solved using available branch-and-cut solvers. However, the size of this problem is directly related to the number of considered wind turbines and the size of wind farms. For large-scale wind farms, the problem becomes complicated and compute-intensive, which forces us to make use of partitioning techniques such as Benders decomposition [156,157].

Benders decomposition is a popular optimization technique. Using the Benders decomposition method, an original large-scale optimization problem could be decomposed into a master problem and several slave subproblems, which are then iteratively solved by generation of dual variables from the subproblem to form the benders cut and add it into the master problem until the stopping criteria are met. In general, the master problem is an integer program, and subproblems are linear or non-linear programs with real variables.

In this research, based on the predetermined locations of substations and wind turbines, the master problem optimizes the annualized total cost, which includes the investment cost, maintenance cost and levelized power losses cost, for the collector system layout of the wind farm; while the subproblems deal with the network security problem for each period under study, with the cable capacities and power flow being fixed. This method could appropriately deals with the proposed non-convexity associated with binary variables and to divide the global problem into two smaller problems, which are easier to solve. Therefore, the CSLO problem is reformulated as the following standard Benders decomposition formulation:

$$Min.\ c^T x \tag{5.22}$$

$$s.t.\ \ Ax \geq b \tag{5.23}$$

$$Ex + Fy \geq h \tag{5.24}$$

where, x represents the cable connection state $\boldsymbol{I}$; y represents the system control variables $\boldsymbol{P}$, $\boldsymbol{Q}$, $\boldsymbol{V}$ and $\boldsymbol{\delta}$; A, E, and F are the coefficient matrices; (5.23) represents constraints (5.9)-(5.11); and (5.24) represents the remaining constraints (5.5)-(5.8).

Accordingly, the initial master problem MP1 can be represented as:

$$\begin{aligned} &Min.\ c^T x \\ &s.t.\ \ Ax \geq b \end{aligned} \tag{5.25}$$

If MP1 is infeasible, the optimization will stop. Otherwise, based on the solution $\hat{x}$ obtained in MP1, constraint (5.24) is checked in subproblem SP. In this level, a slack vector s is introduced and SP is formulated as:

$$\begin{aligned} &Min.\ \omega(\hat{x}) = 1^T \mathrm{s} \\ &s.t.\ \ Fy + s \geq h - E\hat{x} \quad \rightarrow \lambda \end{aligned} \tag{5.26}$$

where, 1 is the unit vector; λ is the dual multiplier vector of inequality constraints in (5.26). $\omega(\hat{x}) > 0$ means that violations occur in the subproblem, i.e. the constraints are uncoverged. To eliminate the violations, the Benders cut (5.27) is introduced and added back to the master problem, and the new master problem MP2 with Bender cuts is then generated for next iteration.

$$\omega(x) = \omega(\hat{x}) - \lambda^T E(x - \hat{x}) \leq 0 \tag{5.27}$$

The final solution based on the Benders decomposition may require an iterative process between the master problem and subproblems. Therefore, a maximum iteration time M is introduced to eliminate this process.

The procedure followed includes the steps illustrated in the flowchart Fig. 5-2.

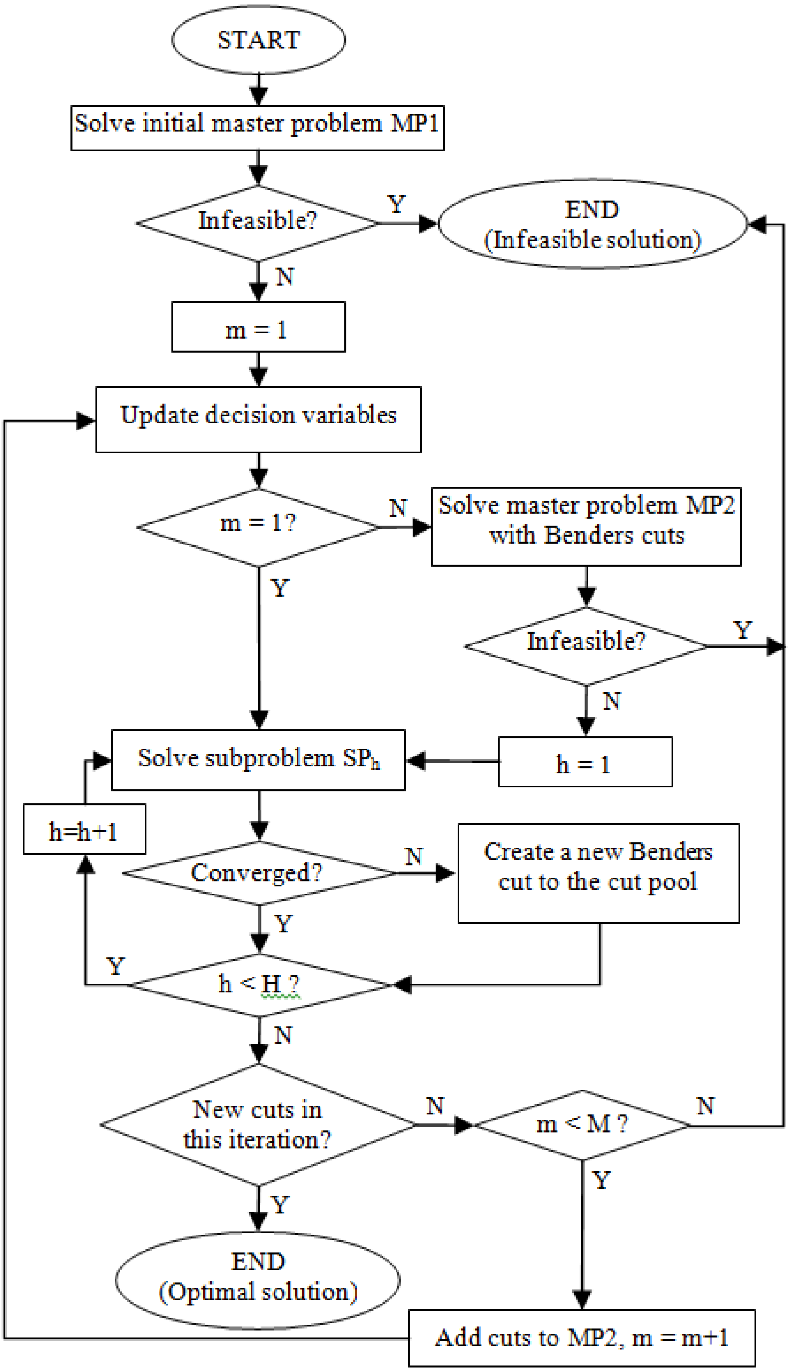


Fig. 5-2. The proposed method for the CSLO problem

1) Master Problem:

The master problem, which is composed of the objective function (5.1) and constraints

(5.10)–(5.11), determines the cable layout scheme of the wind farm, to connect between two wind turbines or one wind turbine and one substation. All binary variables $\boldsymbol{I}$ are included in the optimization problem of this level, and CPLEX [158] solver is applied to solve this mix-integer linear programming (MILP) problem.

The objective function minimizes:

$$F^{*} = C_1\left(\boldsymbol{I}\right) + C_2\left(\boldsymbol{I}\right) + C_3\left(\boldsymbol{I}\right) + \alpha^{*} \tag{5.28}$$

subject to the constraints (9)-(11), and the Benders cuts

$$\alpha^{*} \geq \alpha\left(I^{m-1}\right) + \sum_{c\in\Psi^{C}} \sum_{i\in\Psi^{N}} \sum_{j\in\Psi^{N}} \lambda_{ij,c}^{m-1}\left(I_{ij,c} - I_{ij,c}^{m-1}\right) \tag{5.29}$$

The term α^{*} represents an unfeasibility cost of the slave subproblem. $\alpha\left(I^{m-1}\right)$ in Benders cuts (5.29) is the optimal value of the subproblem in iteration *m-1*. Note that for the initial master problem MP1, there is no any Benders cut constraint and α^{*} equals zero.

2) Subproblem:

The slave subproblem level checks the feasibility of the master problem solution. In order to check the transmission flows in the collector system and gain a feasible solution, several slack variables are added into the power flow equations and the cable line constraints. Therefore, the hourly network security check subproblem is formulated as:

Objective:

$$\begin{aligned} Min. \sum_{i\in\Psi^{N}} \Big(& s_i^{P1} + s_i^{P2} + s_i^{Q1} + s_i^{Q2} + s_i^{V1} + s_i^{V2} + s_i^{V3} \\ & + s_i^{\delta 1} + s_i^{\delta 2} + s_i^{\delta 3} \Big) + \sum_{i\in\Psi^{N}} \sum_{i\in\Psi^{N}} \left(s_{ij}^{S1} + s_{ij}^{S2} + s_{ij}^{S3} + s_{ij}^{S4} \right) \end{aligned} \tag{5.30}$$

Subject to:

$$P_i^{g} + s_i^{P1} = V_i \sum_{j\in\Psi^{N}} V_j G_{ij}\left[\cos\left(\delta_i - \delta_j\right) + B_{ij}\sin\left(\delta_i - \delta_j\right)\right] + s_i^{P2}, \forall i \in \Psi^{N} \tag{5.31}$$

$$Q_i^{g} + s_i^{Q1} = V_i \sum_{j\in\Psi^{N}} V_j \left[G_{ij}\sin\left(\delta_i - \delta_j\right) - B_{ij}\cos\left(\delta_i - \delta_j\right)\right] + s_i^{Q2}, \forall i \in \Psi^{N} \tag{5.32}$$

$$V_i^{\min} + s_i^{V1} \leq V_i + s_i^{V2} \leq V_i^{\max} + s_i^{V3}, \forall i \in \Psi^{N} \tag{5.33}$$

$$\delta_i^{\min} + s_i^{\delta 1} \leq \delta_i + s_i^{\delta 2} \leq \delta_i^{\max} + s_i^{\delta 3}, \forall i \in \Psi^{N} \tag{5.34}$$

$$S_{ij} + s_{ij}^{S1} \leq \sum_{c\in\Psi^{C}} I_{ij}^{c} S_c^{\max} + s_{ij}^{S2}, \forall i, j \in \Psi^{N} \tag{5.35}$$

$$-S_{ij} + s_{ij}^{S3} \geq \sum_{c\in\Psi^C} I_{ij}^c S_c^{\max} + s_{ij}^{S4}, \forall i, j \in \Psi^N \tag{5.36}$$

where, s_i^{P1}, s_i^{P2}, s_i^{Q1} s_i^{Q2}, s_i^{V1}, s_i^{V2}, s_i^{V3}, $s_i^{\delta 1}$, $s_i^{\delta 2}$, $s_i^{\delta 3}$, s_{ij}^{S1}, s_{ij}^{S2}, s_{ij}^{S3}, and s_{ij}^{S4} are the slack variables, which represent the requirements of active power outputs, reactive power outputs, bus voltage magnitudes and angles, and cable line limits, are included in the original constraints (5.5)-(5.9) to make the optimization solution feasible. And the new constraints are as shown in (5.31)-(5.36).

It could be regarded as a nonlinear programming (NLP) problem, and solved with MINOS solver [159] in this research. In iteration *m*, if the subproblem SP_h in hour *h* is uncoverged, a new Benders cut will be created and added into the cut pool CP_m. And the iteration will go on for the next hour subproblem SP_{h+1}. After the subproblems are finished for all the hours, the cuts stored in CP_m will be added to master problem MP2 for the optimization of next iteration. Note that the Benders cuts used in the previous *1*, *2*, …, *m-1* iterations will also be included as constraints in the current iteration *m* for master problem MP2. If there is no any new cut obtained in the solution of subproblem in current iteration *m*, i.e. the cut pool CP_m is empty, the iteration will stop with the optimal solution.

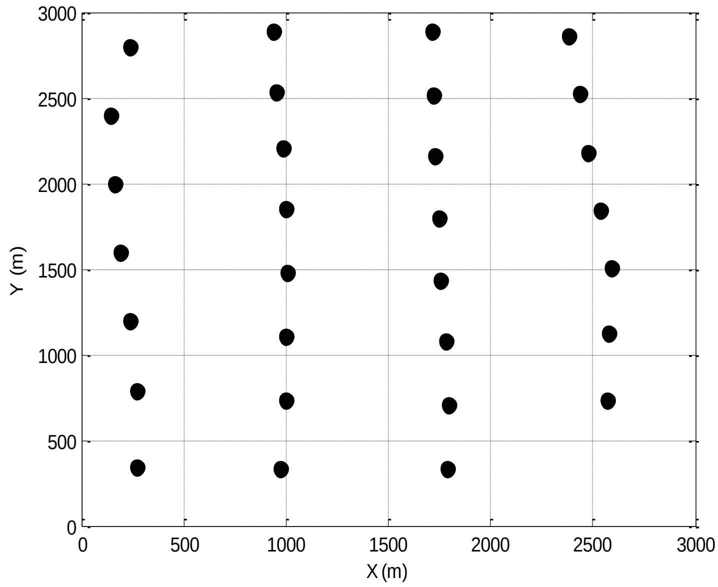
5.6 Comparison Case Studies

5.6.1 Case Settings

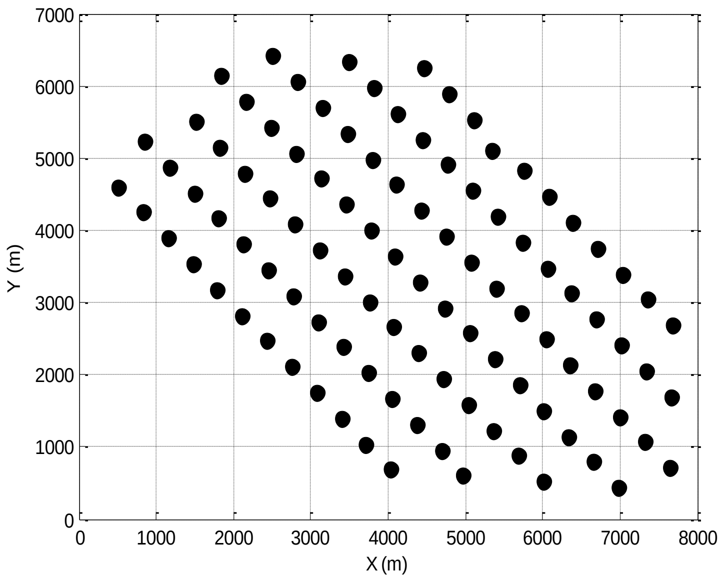
The proposed algorithms have been tested on two benchmark systems. One is a 90MW offshore wind farm consisting of thirty 3.0MW wind turbines; the other one is a 300MW farm with one hundred 3.0MW turbines. Optimal wind turbine placement considering effects of local wind patterns, land topography, wind turbine wake effects is a widely researched area. Therefore, for the purpose of this work, it has been assumed that the wind turbine locations in the wind farm are already available, which has been justified by the availability of abundant literature in this area [160]. It is also assumed that the availability of these wind turbines is 100% here, and the bus voltage level is set to be 33kV.

The turbine layouts, marked as the black dots, of these two wind farms are represented in

Fig. 5-3. The Vestas V112 3.0MW offshore wind turbine model [161] is selected here. The cut-in, cut-out, and rated speeds of this type of turbine are 3m/s, 25m/s, and 12m/s respectively, and the mechanical power captured by a given wind turbine is a function of wind speed, which is shown in Fig. 5-4. Besides, the historical wind data in one year was obtained from one wind observation station, which was provided by the Australian Bureau of Meteorology [162]. The wind speed distribution and Weibull fitting are given in Fig. 5-5.



(a) 30-WT offshore wind farm



(b) 100-WT offshore wind farm

Fig. 5-3. The turbine layouts of 30-WT and 100-WT offshore wind farms

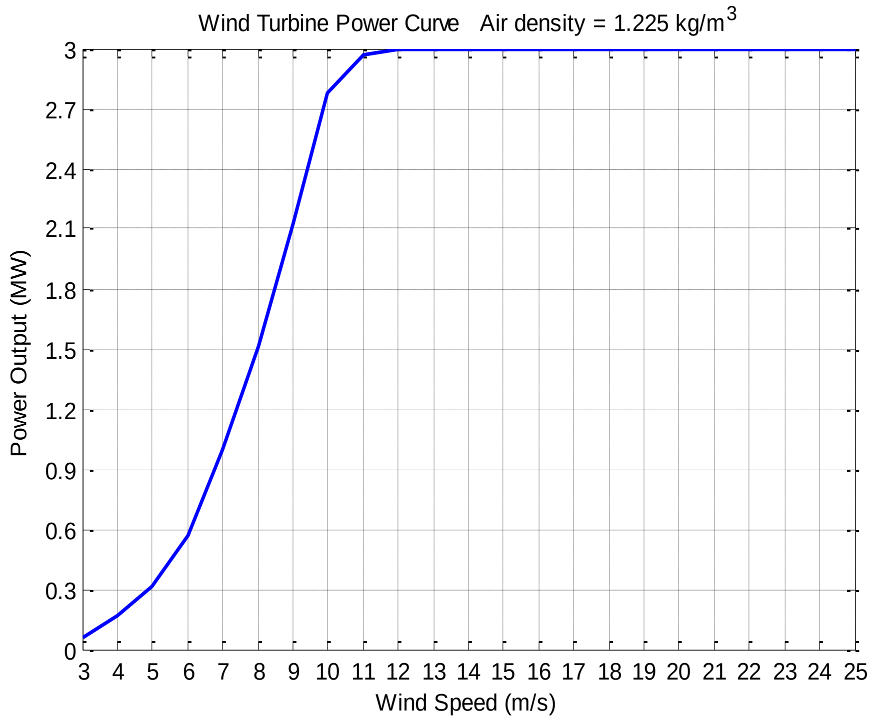


Fig. 5-4. The power output curve of Vestas V112/3.0MW offshore wind turbine

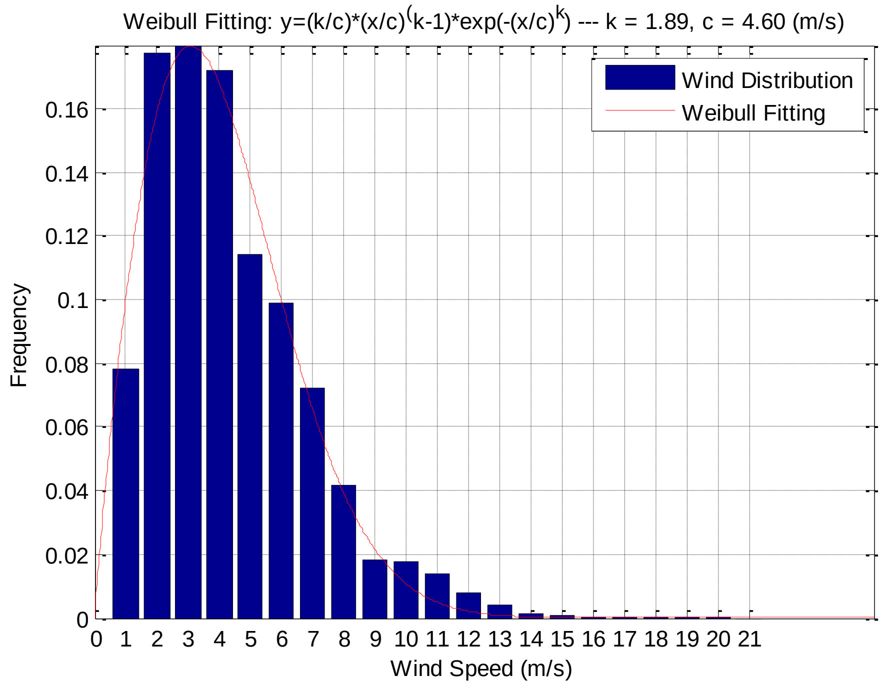


Fig. 5-5. Wind speed distribution and Weibull fitting

Three cases are studied for the simulations. Case 1 is the base case which just considers single substation and single cable type in the CSLO problem. In Case 2, the consideration of multiple substations with different capacities is included in the optimization. And in Case 3, multiple substation with different capacities, and two types of cables with different electric parameters are considered simultaneously. A detailed comparison is made between these three cases.

Table 5-1. Electric Parameters of Cable
(A: HEPRZ1 18/30 kV 1x150 mm^2, B: HEPRZ1 18/30 kV 1x400 mm^2)

Item	**A**	**B**
Rmax (Ω/km)	0.264	0.100
X (Ω/km)	0.118	0.103
Capacitance (μF/km)	0.253	0.366
Investment Cost ($/m)	7.60	12.25
Power Loss Cost ($/kWh)	0.50	0.50
Maximum current (amp)	248.4	426.42

In the simulation, two types of cable are chosen and their electric parameters are tabulated in Table 5-1. Some of other common cable sizes used in large-scale offshore wind farms can be found in [163]. All the computational experiments were obtained using CPLEX and MINOS solvers with default options under AMPL [164] and Matlab tools on a Windows-based PC with 2.40 GHz and 4 GB of RAM.

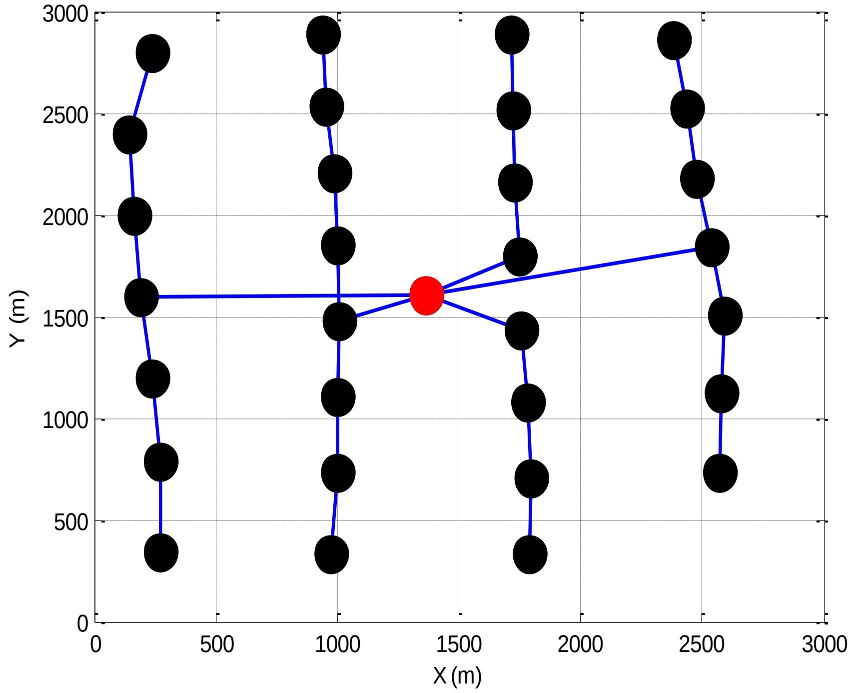
5.6.2 Case I: 30-WT Offshore Wind Farm

The 30-turbine offshore wind farm system in Fig. 5-3(a) is used in illustrate the proposed method. The totally layout boundary of the wind farm is 3-by-3 km, and the farm area is around 9 km^2. In Case 2 and 3, the maximum number of turbines for each substation is set to 15, and the locations of substations are obtained using the proposed SAA method presented in Section 5.3.

Besides, the MST method presented in Section 5.4 is also simulated to compare with our proposed CSLO model in Case 1. The comparison results for three cases are listed in Table 5-2. The cost of investment cost C_1 and maintenance cost C_2, power losses cost C_3, and the total cost is all included. The optimal layouts of the 30-turbine offshore wind farm for the three cases are illustrated in Fig. 5-8, in which the black dots are the location of turbines, red dots are substations, blue lines are the cable connections of type A, and red lines are the connections of cable type B with higher capacity.

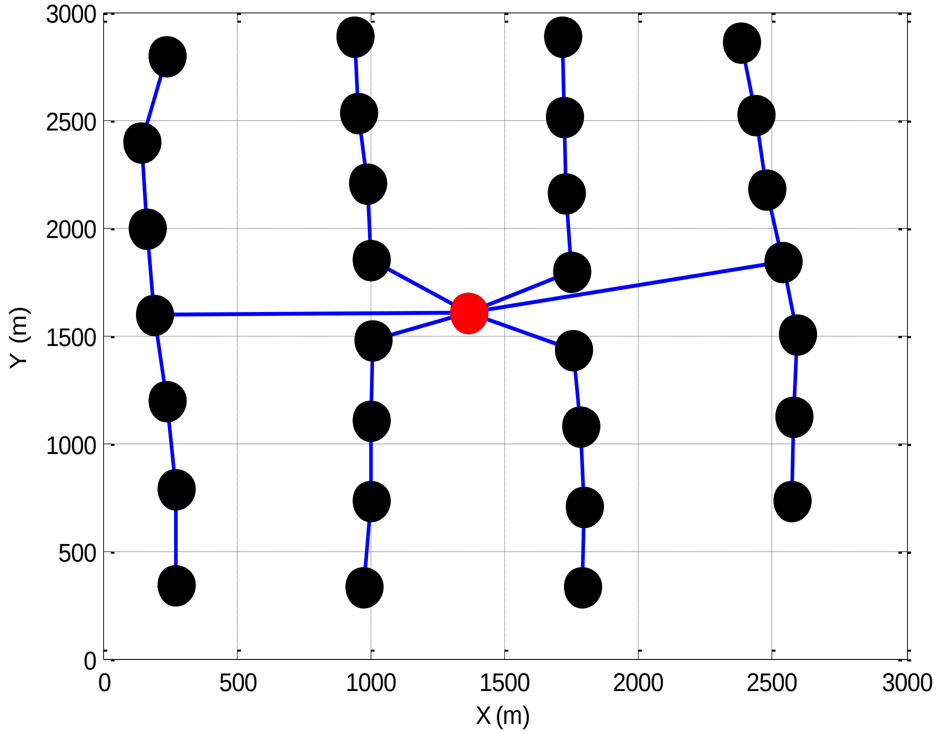
Table 5-2. Simulation Results for the 30-WTs Offshore Wind Farm

Case	1 (MST)	1	2	3
C_1+ C_2 ($)	119,197.9	121,135.5	118,868.2	120,458.2
C_3 ($)	134,678.6	134,721.6	71,577.6	70,972.5
Total Cost ($)	253,876.5	255,857.1	190,446.8	191,430.7

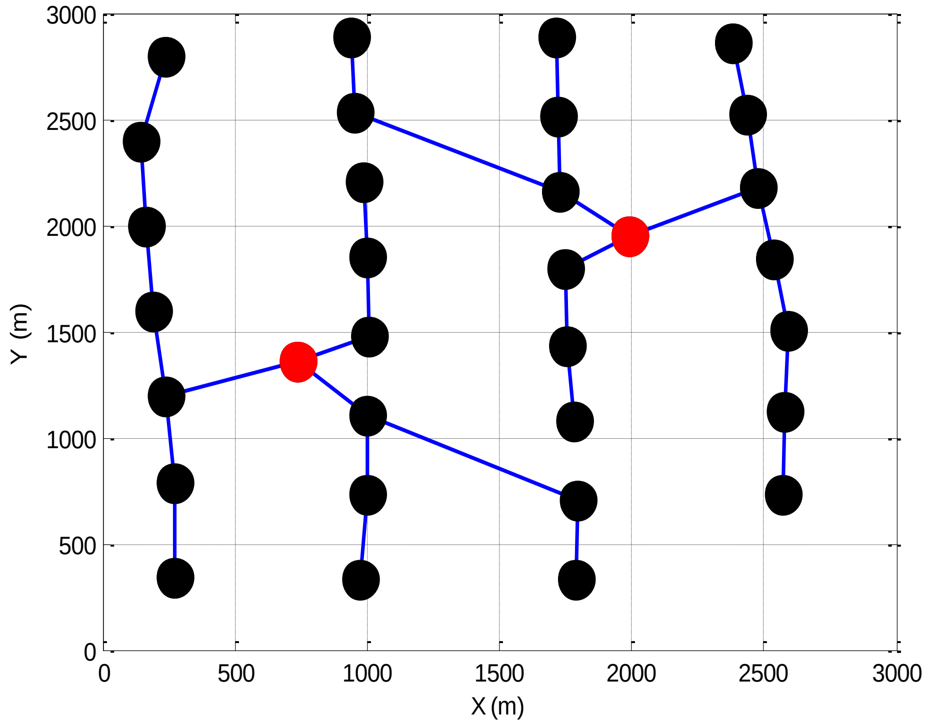


(a) Case 1 with MST method

Fig. 5-6. The optimal layouts of 30-WT offshore wind farm

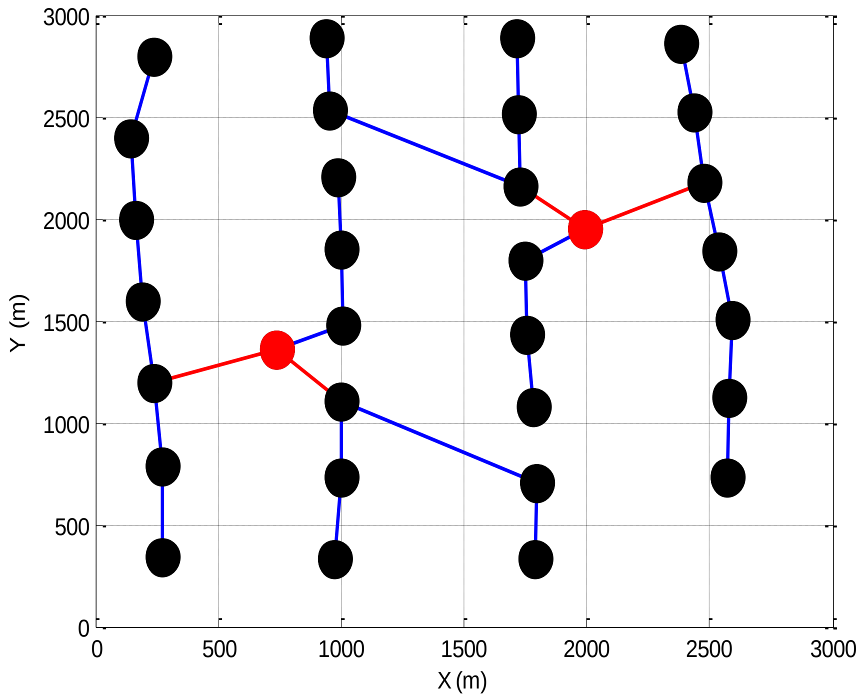


(b) Case 1 with CSLO method



(c) Case 2 considering two substations

Fig. 5-7. The optimal layouts of 30-WT offshore wind farm



(d) Case 3 considering two substations and two cable types

Fig. 5-8. The optimal layouts of 30-WT offshore wind farm

It can be seen from Table 5-2 that the optimal costs of MST method are less than the ones of CSLO method. However, according to Fig. 5-8 (a), with the MST method, two cable lines cross over each other in the optimal solution, which is not permitted in the collector system layout design for offshore wind farms. While this problem is solved automatically using our proposed CSLO method.

When considering two substations, although the investment cost of substation is increased doubly, the total length of cables is decreased from 12.6 km to 12.3 km, and thus the total power losses cost is decreased correspondingly. And in Case 3, the turbines near the substation are connected with the higher capacity cables (type B). Since a cable with higher capacity is more expensive, the investment cost of cable in Case 3 is higher than that in Case 2. However, the total power losses cost is cheaper in Case 3 when considering two different cable types and the total costs of two cases are almost equal.

5.6.3 *Case II: 100-WT Offshore Wind Farm*

To demonstrate the efficiency of the proposed CSLO model for large-scale offshore wind farms, a 100-turbine system in Fig. 5-3 is simulated here. The totally layout boundary of the wind farm is 8-by-7 km, and the farm area is around 56 km^2. In this study, the farm is divided into two zones. In each zone, the maximum number of turbines for each substation is set to 50. The simulation results are shown in Table 5-3 and Fig. 5-9 and Fig. 5-10 for Case 2 and Case 3.

Table 5-3. Simulation Results for the 100-WTs Offshore Wind Farm

Case	**2**	**3**
C_1+ C_2 ($)	556,531.6	560,228.2
C_3 ($)	884,752.4	796,462.1
Total Cost ($)	1,441,284.0	1,356,690.3

The result indicates that the consideration of two substations and two cable types makes possible to decrease the total power losses. Even the investment cost is higher than before, the total cost is decreased. In Case 3, the total length of optimal cables is 58.4 km and the average power loss in the cables is 181.8 kW per hour, which is 0.06% of the installed wind farm capacity. Although the network power loss cost is high, when spread out over the lifetime of the cables, the annual costs do not vary significantly.

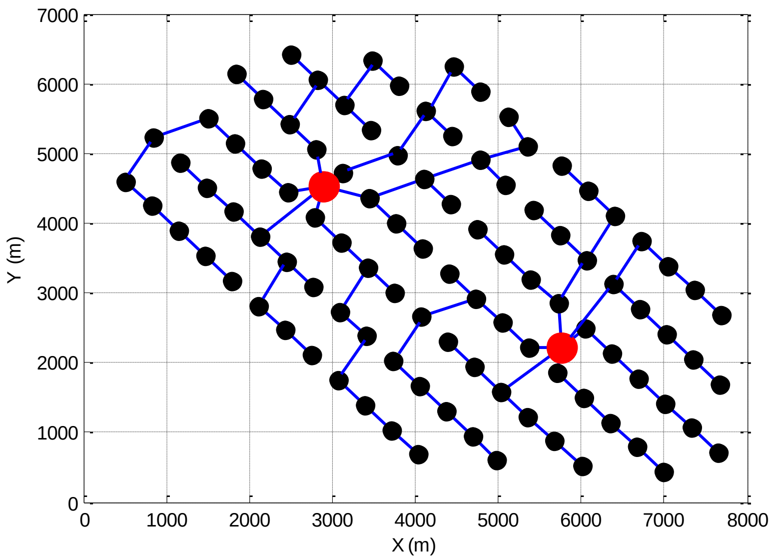


Fig. 5-9. The optimal layout of 100-WT OWF for Case 2

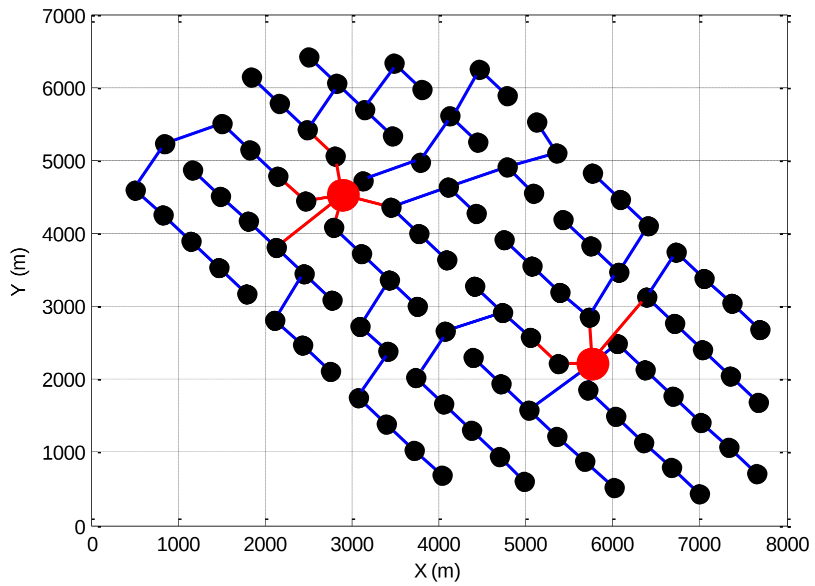


Fig. 5-10. The optimal layout of 100-WT OWF for Case 3

5.7 Summary

This chapter presents a new and efficient collector system layout optimisation (CSLO) model for large-scale offshore wind farms. The objective of this model is to minimise the annualised investment cost, maintenance cost, and the levelized power losses cost, while satisfying network reliability and operational constraints such as AC power flow constraints, capacity limits of substations and wind turbines, and network radiality constraints. Multiple substations and cable types are also considered in this model. Benders' decomposition algorithm is applied here to solve this mixed-integer nonlinear programming (MINLP) problem. The master problem optimises the annualised total cost, which includes the investment cost, maintenance cost, and power losses cost for the collector system layout of the wind farm; while the sub-problem deals with network security problem while the cable capacities are being fixed. Moreover, a self-adaptive allocation (SAA) method for wind turbines is proposed based on fuzzy c-means (FCM) clustering algorithm to exclusively allocate wind turbines to their nearest substations and obtain the topology structure of cables which are utilised to connect turbines and substations.

Two offshore wind farms with 30 and 100 turbines respectively are introduced for the case studies. A minimum spanning tree (MST) algorithm is also utilised to solve the CSLO problem and to compare with the proposed methods based on Benders' decomposition. The study results clearly demonstrate the feasibility of the proposed methods and show that it can be used as a reliable optimisation tool for collector system layout in large-scale offshore wind farms.

Chapter 6

OPTIMAL POWER DISPATCH FOR WIND FARM INTEGRATED POWER SYSTEMS

6.1 Introduction

The penetration of renewable energy into the power gird has been increasing rapidly in recent years due to the environmental concerns. However, because of the necessity for grid stability, the intermittency of these renewable sources is a major factor inhibiting their integration. The development of battery energy storage systems (BESS) enables renewable energy generation with flexible operation to meet the requirements of the grid.

This chapter will propose an optimal short-term wind farm dispatch model and an efficient method with BESS for better integration of wind energy. The effectiveness of the proposed method will also be tested with wind farm case studies to demonstrate that the optimal plan of charging and discharging processes and wind energy shedding can help reduce the fast intermittency and high fluctuation of wind power to meet grid requirements. Based on these case study results, the wind farm operator can make beneficial decisions through optimal dispatch planning.

6.2 Wind Farm System with BESS

6.2.1 System Modeling

The basic investigated system is a grid-connected wind farm with a BESS. The schematic of this system is showed as Fig. 6-1. The power of wind generator and battery are converted through convertors and inject to the grid at the common coupling point. The BESS is applied to smooth the fluctuation of wind farm output. To achieve this target, the control system generate the reference charging power signal for the battery and pitch angle for the wind machine according to the forecast information and the status of the system. This algorithm will be introduced in this research.

The total wind farm output is express as equations below,

$$P_{Total}^{t} = P_{Wind}^{t} + P_{BESS}^{t} \tag{6.1}$$

where P_{Total}^{t} is the total power injected to grid of wind farm at time t; P_{Wind}^{t} is the output of wind machine at time t; and P_{BESS}^{t} is the power output of BESS at time t.

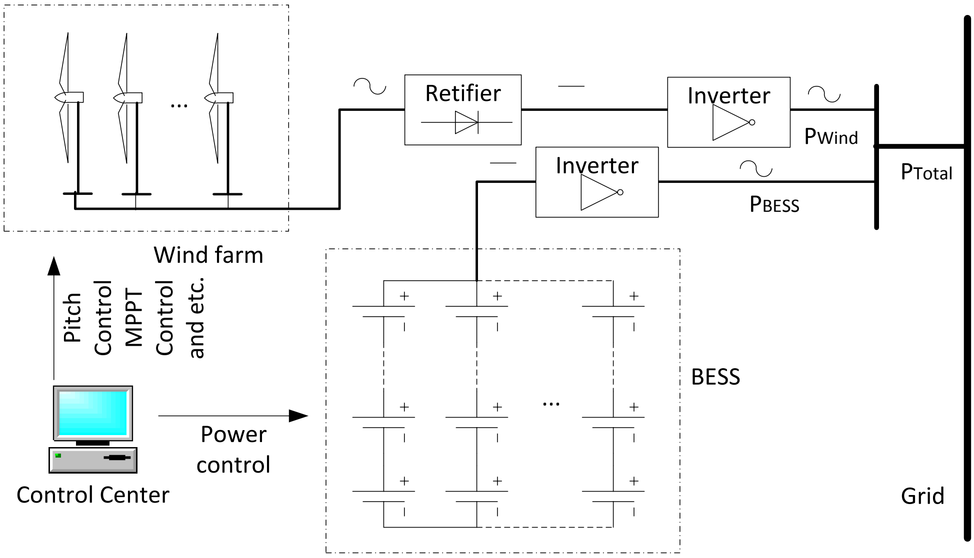


Fig. 6-1. Schematic of the grid-connected wind farm with BESS

6.2.2 *Battery Charging Issues*

Suitable capacity storage of energy and charging at appropriate time can help to smooth the output of wind farm and to maintain system stability. In the following study, battery operation issues are performed for better utilization of BESS in the wind farm. Table 6-1 compares the parameters of Lithium-Ion battery and Lead-Acid battery [165]. These two types of batteries are considered in this research.

Table 6-1. Power change limit of wind farm

Item	**Li-Ion Battery**	**Lead-Acid Battery**
Energy Density	250 Wh/kg	30 Wh/kg
Power Density	500 W/kg	30 W/kg
Life Cycle	1500 times	1000 times
Self-discharge	2% per month	3% per month
Reliability	Medium	High
Investment cost *CI* ($)	$300000E_r$	$80000E_r$

The energy changing of BESS, the key issue for battery operation strategy, can be described as,

$$E_{BESS}^{t+1} = E_{BESS}^{t} + \Delta t \cdot P_{BESS}^{t} - | P_{BESS}^{t} | \cdot \eta_c \cdot \Delta t - E_{BESS}^{t} \cdot \eta_l \cdot \Delta t \tag{6.2}$$

The state-of-charge (SOC) is expressed as follows,

$$SOC^{t} = E_{BESS}^{t} \Big/ E_{BESS}^{r} \tag{6.3}$$

And the operation of BESS is constrained to,

$$\text{Power limits:} \quad P_{BESS}^{Dis,Max} \le P_{BESS}^{t} \le P_{BESS}^{Chr,Max} \tag{6.4}$$

$$\text{SOC limits:} \quad SOC^{Min} \le SOC^{t} \le SOC^{Max} \tag{6.5}$$

6.2.3 *Wind Speed Forecasting*

Wind speed forecast plays a key role in the operation of a wind farm power dispatching. It is a basic task to facilitate renewable penetrations [166,167]. The forecasting software, *OptiWind*, is used for wind speed forecasting in this research [168]. Several important factors considered in the forecasting model include temperature, seasonal weather, public holidays, and historical load data. *OptiWind* incorporates highly customized numerical weather prediction models and the latest statistical methods. The mean absolute percentage error (MAPE) is used to assess the forecast accuracy:

$$MAPE = \frac{1}{N_h} \sum_{p=1}^{N_h} \frac{| P_p - \overline{P_p} |}{P_p} \times 100\% \tag{6.6}$$

Where N_h is the sampling quantity; P_p is the actual demand; and $\overline{P_p}$ is the forecasted demand. The curve of forecast wind speed and real one within the time horizon are compared in Fig. 6-2.

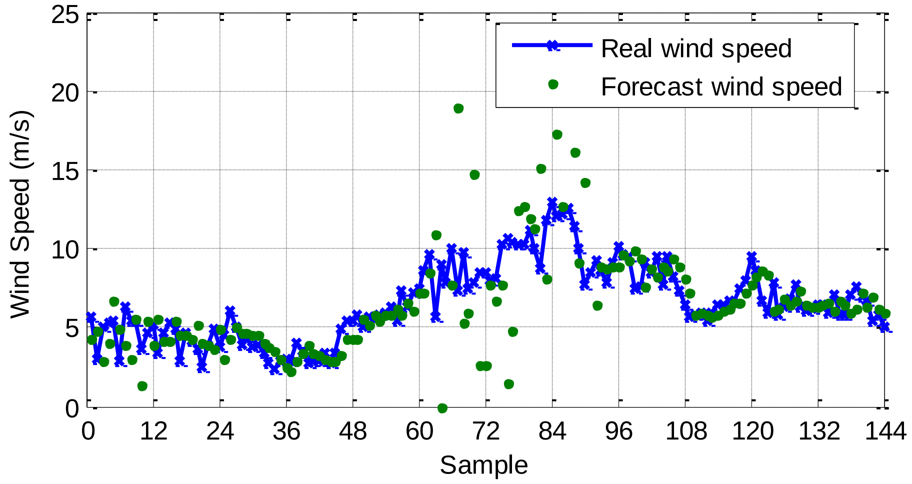


Fig. 6-2. Wind speed forecasting curve

Wind power is converted form kinetic energy by wind turbine and the output strongly depends on the wind speed. The wind turbine output power is given by [169],

$$P_M = \frac{1}{2}\pi r^2 \rho v^3 \frac{\left(1+\frac{v_0}{v}\right)\left[1-\left(\frac{v_0}{v}\right)^2\right]}{2} \tag{6.7}$$

Where v and v_0 is the upstream wind speed and downstream wind speed, respectively. ρ is the air density and r is turbine rotor radius. The real wind power is controlled through rotor speed by maximum power point tracking (MPPT) method. Meanwhile, the pitch angle of the blade can be controlled to adjust the output of wind machine. The wind machine will started when the wind speed reach the lower limit and will be stopped under high wind speed as Eq. (6.8).

$$P_{Wind} = \begin{cases} 0, & v_0 < v_{low} \\ P_M, & v_{start} \le v_0 \le v_{cut} \\ 0, & v_{cut} < v_0 \end{cases} \tag{6.8}$$

6.3 Proposed Power Dispatch Scheduling Method based on Model Predictive Control

6.3.1 *Model Predictive Control based Dispatch Scheduling*

With the framework of wind farm and the forecast wind speed, the MPC strategy is proposed to pursue the optimal wind power dispatch. Within the MPC-based dispatch scheme, a finite time horizon and time interval is decided first. Before the dispatch, the optimization process is performed to generate a series power dispatch in this time horizon. After the first action of the optimal plan, the system information is updated with actual data and repeat the optimization for next step with the fixed time horizon. In this section, detailed step of the MPC-based dispatch scheduling is implemented as Fig. 6-3.

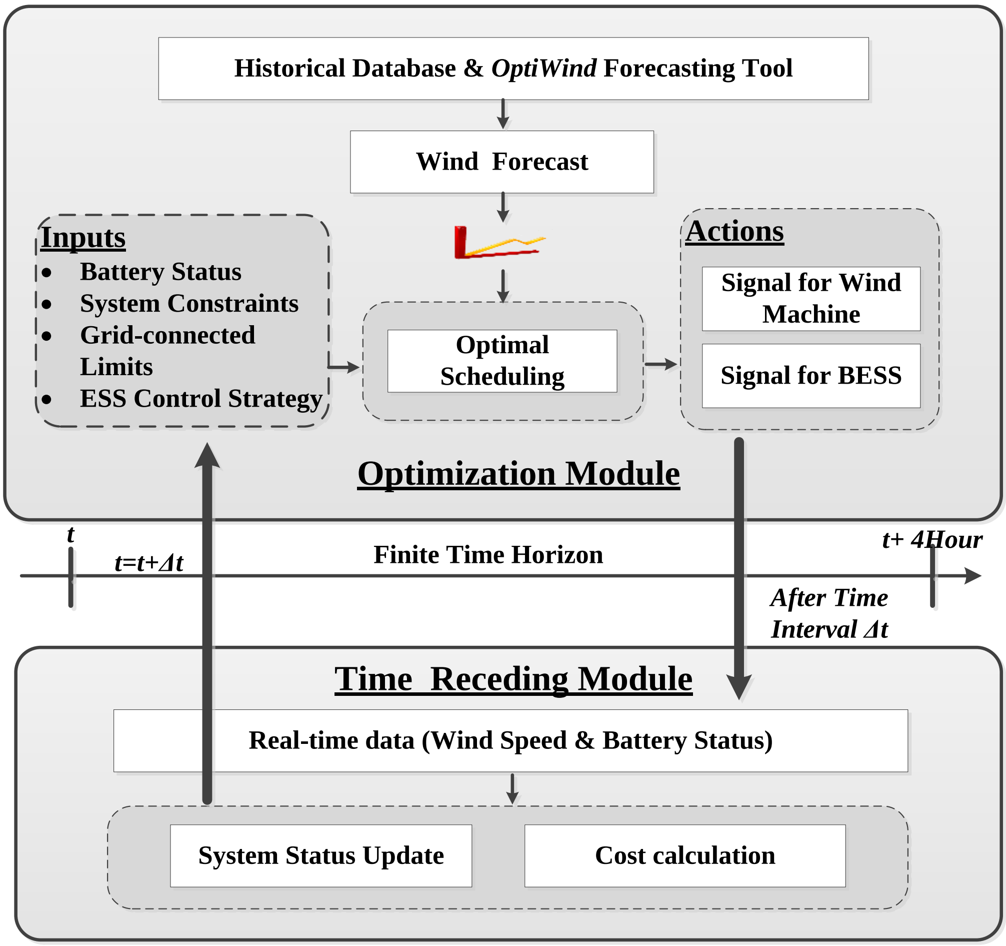


Fig. 6-3. Schematic of MPC-based dispatch scheduling step

6.3.2 *Objective and Constraints*

Based on the frame of MPC-based wind farm power dispatch scheduling, the reference signal is send to BESS and wind machine to adjust total output of the wind farm. During each step, the optimal reference signal is generate for the whole time horizon. After the time interval, this optimization problem is resolved with updated information for next time horizon.

In this research, given the forecasted wind speed and the configuration of the battery, the optimal dispatch schedule aims to control the charging/discharging currents of the battery over the dispatch horizon. The whole objective of the wind farm dispatch is to minimize the sum of the following three aspects of the operation cost:

a) The profit loss incurred by cutting off the wind power;

b) The cost of the battery's lifetime loss;

c) Penalty cost incurred by violating the ramp rate constraint.

Specifically, the objective and constraints are introduced as follows.

1) Objective function

The objective function of the wind farm dispatch model is as follows.

$$f = \min(\sum_{t=1}^{T}(C_{Cut}^{t} + C_{BESS}^{t} + C_{penalty}^{t}) \tag{6.9}$$

where C_{cut}^{t} is the cost of wind power curtailed at time t; C_{bes}^{t} is the cost of depression of battery lifetime at time t; $C_{penalty}^{t}$ is the penalty cost by violating the ramp rate constraint at time t. Detail cost function of C_{cut}^{t} , C_{bes}^{t} , and $C_{penalty}^{t}$ are expressed as Eqs. (6.10) - (6.12).

$$C_{cut}^{t} = \begin{cases} 0, & if \ \left|P_{Total}^{t} - P_{Total}^{t-1}\right| \le P_{ramp} \\ p_{violate} \cdot (\left|P_{Total}^{t} - P_{Total}^{t-1}\right| - P_{ramp}) \cdot \Delta t, & if \ else \end{cases} \tag{6.10}$$

$$C_{BESS}^{t} = \beta \cdot P_{BESS}^{t} \cdot \Delta t + \beta \cdot E_{BESS}^{t} \cdot \eta_{l} \cdot \Delta t \tag{6.11}$$

$$C_{Cut}^{t} = \begin{cases} 0, & \text{if } P_{wind}^{t} = P_{wind,\max}^{t} \\ \delta \cdot (P_{wind,\max}^{t} \text{-} P_{wind}^{t}) \cdot \Delta t, & \text{else} \end{cases} \tag{6.12}$$

where $P_{BESS}^{t} > 0$ indicates the battery is discharged, and $P_{BES}^{t} < 0$ indicates the battery is charged. β is the cost coefficient of the battery lifetime depression. δ is the penalty coefficient of violating the ramp power constraint. It should be noted that, the battery cost is calculated by the total energy usage. According to the impacts of discharge rate on battery life in [170], the total energy usage of battery is remained stable within the reason depth of discharge (DOD). Base on this conclusion, the β can be calculated as,

$$\beta = \frac{CI}{E_r \cdot (Life\ Cycle)} \tag{6.13}$$

2) Constraints

The dispatch model is constrained by following constraints.

a) Battery rated power capacity constraints, which are show as Eqs. (6.4)- (6.5);

b) Wind machine constraints, which is operated under appropriate wind speed.

c) Power Limits, which forbids the reverse power flow in this research. I.e.,

$$P_{Total}^{t} \geq 0 \tag{6.14}$$

6.4 Distributed Imperialist Competitive Algorithm

6.4.1 Introduction of ICA method

The proposed optimal power dispatch model is a binary, non-convex, high dimension, combinational optimization problem, which is hard to be handled by the conventional programming methods. In this research, a new heuristic searching algorithm proposed by recent years, ICA, is employed to solve the proposed model. And in order to enhance its searching performance and computational speed, a distributed processing architecture for ICA is proposed.

ICA is first proposed in 2007 [171]. In recent years, ICA has been successfully applied in unit commitment [172], economic dispatch [173], and other industrial applications [174,175]. ICA mimics the competition among the imperialists. Each imperialist possess some colonies to form an empire, and tries to expanse its power by possessing the colonies of other empires. During the competition, weak empires collapse and powerful ones take possession of their colonies.

1) Empire Initiation

As other heuristic searching algorithms, ICA maintains a population of individuals with the number of N_{pop}. In ICA, each individual is called a *country*. For a *N*-dimension minimization problem, the *i*th country is a $1 \times N$ vector with the form of $country_i = [x_i^1, x_i^2, ..., x_i^N]$. Each country represents a solution for a given problem, and its cost c_i can be obtained by evaluating it as $c_i = f(country_i) = f([x_i^1, x_i^2, ..., x_i^N])$.

In the start, N_{imp} of the countries are selected as imperialists. The other $N_{pop} - N_{imp}$ countries act as the colonies and are assigned to the N_{imp} imperialists in proportional to the powers of the imperialists. To calculate the imperialist powers, the cost of an imperialist (denoted as c_n) is firstly normalized,

$$C_n = c_n - \max_i \{c_i\} \tag{6.15}$$

where C_n is the normalized cost. Then the normalized power of *n*th imperialist can be calculated by Eq. (6.16).

$$p_n = \left| \frac{C_n}{\sum_{i=1}^{N} C_i} \right| \tag{6.16}$$

The initial number of the colonies possessed by each imperialist (denoted as NC_n) is in proportional to its power,

$$NC_n = round\{p_n \cdot N_{col}\} \tag{6.17}$$

where N_{col} is the total number of the colonies, and then NC_n colonies are randomly selected and assigned to *n*th imperialist.

2) Moving the Colonies towards the Imperialists

As other heuristic searching algorithms, ICA has an iterative process to mutate the individuals. In each iteration, each colony moves moved toward the relevant imperialist by *x* units,

$$x = \omega \cdot (I_n - col_n^i) \tag{6.18}$$

where col_n^i and I_n are the positions of *i*th colony and imperialist of *n*th empire, respectively; ω is the weight factor which is usually set as a uniform random number within (*0*, *2*), so as to make the colony move towards the imperialist in both sides. A random angle θ is also added as the deviation.

$$\theta \sim U(-\gamma, \gamma) \tag{6.19}$$

where γ is the control parameter that adjusts the deviation from the original movement direction. Generally, the movement of the colony can be depicted by Fig. 6-4.

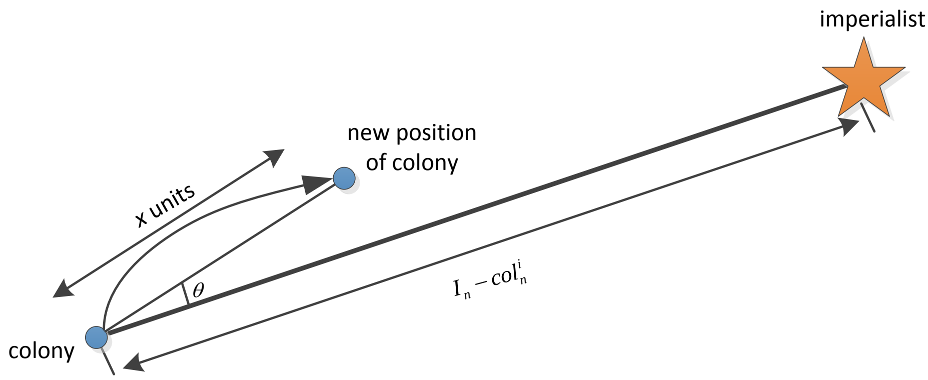


Fig. 6-4. Movement of the colony [171]

3) Updating the Positions of Imperialist and a Colony

After the movement, a colony may reach to a position with lower cost than the imperialist. In this case, the colony and the imperialist must exchange positions. The rest colonies of this empire should move forward the new imperialist position.

4) Imperialist Competition

In each iteration, all empires compete with each other to try to take possession of colonies of other empires. The competition is based on the empires' powers. The total power of *n*th empire (denoted as TC_n) is calculated by Eq. (6.20).

$$TC_n = c_n^{imp} + \xi \cdot \frac{\sum_{i=1}^{NC_n} c_n^i}{NC_n} \tag{6.20}$$

where c_n^{imp} is the cost of the imperialist of *n*th empire; NC_n is the number *n*th empire's colonies; c_n^i is the cost of the *i*th colony of *n*th empire; ξ is the weight factor. The normalized total power (denoted as NTC_n) and possession probability (denoted as p_{p_n}) of *n*th empire is represented as below.

$$NTC_n = TC_n - \max_i \{TC_i\} \tag{6.21}$$

$$p_{p_n} = \left| \frac{NTC_n}{\sum_{i=1}^{N_{imp}} NTC_i} \right| \tag{6.22}$$

To divide the mentioned colony among empires based on their possession probability, vector **P** is formed as **P**=$[p_{p_1}, p_{p_2}, ..., p_{p_{imp}}]$. Then the vector **R** is formed as **R**=$[r_{r_1}, r_{r_2}, ..., r_{r_{imp}}]$, with the elements are uniform distributed random numbers within [*0*, *1*]. Then vector **D** is formed by simply subtracting **R** from **P**,

$$\mathbf{D}=\mathbf{P}\text{-}\mathbf{R}=[p_{p_1} - r_{r_1}, p_{p_2} - r_{r_2}, ..., p_{p_{imp}} - r_{r_{imp}}] \tag{6.23}$$

The mentioned colony then will be assigned to the empire whose relevant index in **D** is maximum. After each iteration, ICA checks whether there exists an empire which has lost all the colonies. If so, the empire will collapse and be eliminated.

5) Termination

The algorithm terminates when either of following two conditions is satisfied:

a) All the empires collapse except for the most powerful one, and all the colonies have the same cost with the imperialist.

b) The preset maximum iteration number is reached.

6.4.2 Binary Structure of ICA

The original ICA algorithm is for the real-value optimization problem. For the proposed DLC model, the decision variables are the on/off states of the ACL groups, which represent a binary searching space. In this research, a binary searching strategy is proposed for ICA.

The key problem is how to interpret the colony movement in a binary space. In ICA, each colony moves by control parameters, which will finally leads to a new position. In a binary space, this movement distance can be modeled as the Hamming distance. A colony with zero bits flipped does not move, while it moves to the 'farthest place' by reversing all of its bits. Therefore, this research interprets the movement distance of each dimension as the bit change probabilities. Specifically, the probability represented by the movement distance of dimension of *j*th dimension of *i*th colony (denoted as d_{ij}) is calculated by using a logistic transformation shown in Eq. (6.18), and the decision of the corresponding new state is shown as Eq. (6.25).

$$\Pr(d_{ij}) = \frac{1}{1+\exp(-d_{ij})} \tag{6.24}$$

$$x_{ij} = \begin{cases} 1, & \text{if } rand() < \Pr(d_{ij}) \\ 0, & \text{otherwise} \end{cases} \tag{6.25}$$

6.4.3 Distributed Process Architecture of ICA

In ICA, multiple empires are formed and evolved separately. The communications among the empires occurs at the end of each iteration, where the weakest colony is re-assigned and the empty empire is eliminated. This characteristic makes ICA suitable for parallel and distributed processing in nature. In this research, a distributed computing architecture is proposed for running ICA across networked processors, depicted in Fig. 6-5.

As depicted by Fig. 6-5, each computing node is assigned to maintain the countries of $round(N_{emp} / N_{node})$ empires. The coordination node is responsible for coordinating the weakest colony re-assignment, empty empire eliminating, and statistics. A central database is established to store the input data of the optimization problem. The computing nodes retrieve data from the database through the network to perform the empire initialization, fitness evaluation and constraint handling.

With the evolution, the weakest empires are gradually eliminated. To balance the computing load, when $N_{emp} < N_{node}$, the idle computing nodes are assigned to part of the colonies of the empires which have largest number of colony. This is depicted by Fig. 6-6. The load balance algorithm is an active research topic in distributed computing. In future, more elaborate load balance strategy for the distributed ICA can be further studied. And the procedure of the distributed ICA is shown in Table 6-2.

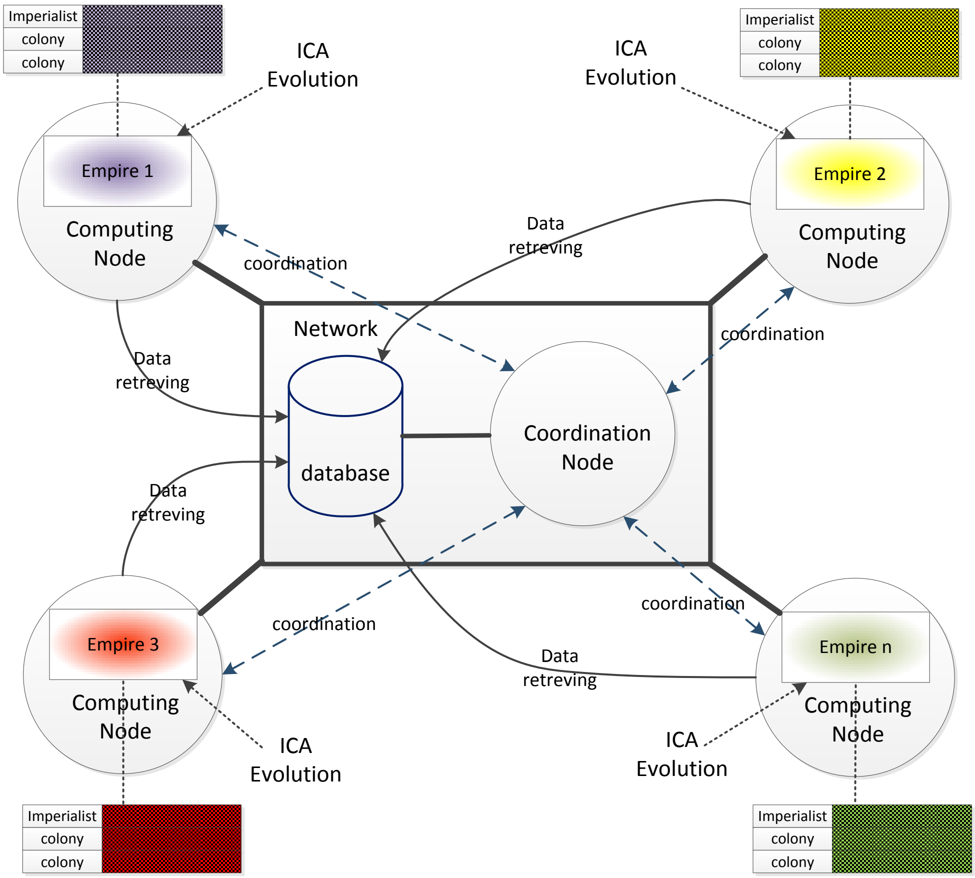


Fig. 6-5. Distributed processing architecture of the ICA

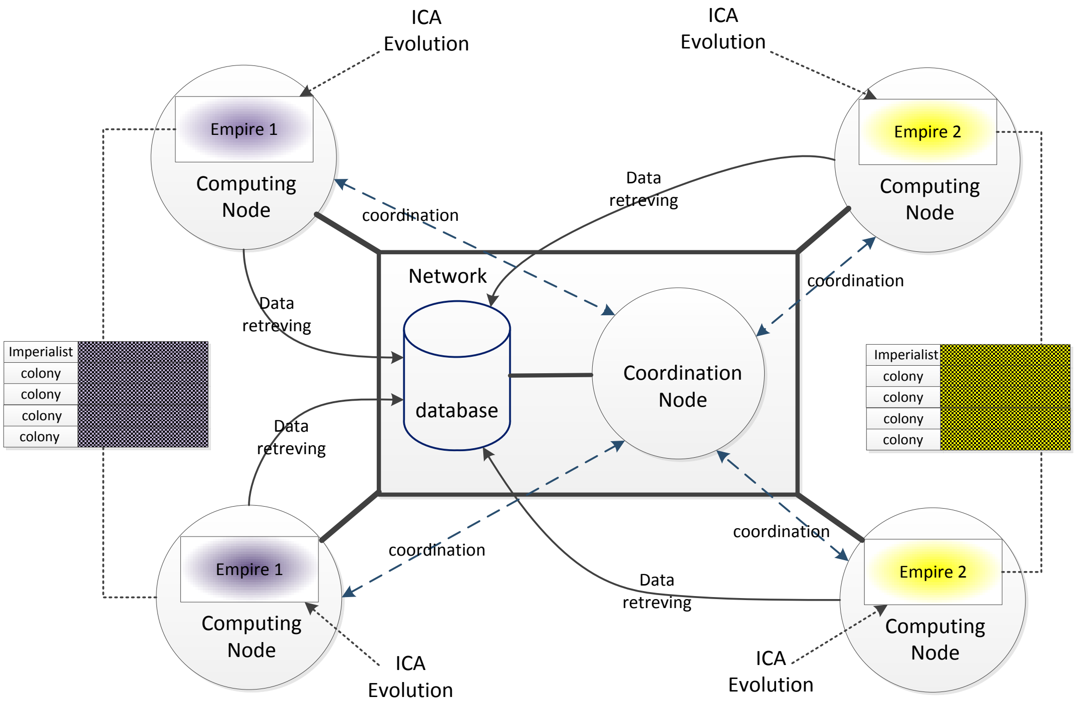


Fig. 6-6. Re-assignment of the computing nodes

Table 6-2. Procedure of the Distributed ICA

prepare the input parameters of the optimization model
assign the empire indexes for all *computing nodes*
for all *computing nodes* **do in parallel**
    retrieval model data from the *database*;
    initialize the imperialist and colonies of the empire;
end parallel
for each iteration
    for all *computing nodes* **do in parallel**
        retrieve the imperialist information;
        move each colony;
        evaluate each colony;
        constraint handling for each colony;
        send the fitness values to the *coordination node*;
    end parallel
    do routine on the *coordination node*
        calculating the powers of the empires;
        choose the weakest colony from the weakest empire;
        choose a powerful empire to possess that colony;
        migrate that colony to the computing node which maintains that empire;
        if there is empty empire **do**
            migrate half of the colonies of the *computing node* which has the

largest number of colony to the *computing node* which maintains the empty empire;

eliminate the information of the empty empire;

end if

synchronize all the *computing nodes*;

end routine

end iteration

do routine on the *coordination node*

output the final optimal imperialist and its fitness value;

end routine

6.5 Solution to the Dispatch Model based on Distributed Imperialist Competitive Algorithm

6.5.1 ICA Coding

The proposed dispatch scheduling model above is a continuous, nonlinear, constrained mixed integer programming problem which cannot be solved by classical deterministic integer programming techniques. In this research, the distributed ICA, as proposed in Section 6.4, is adopted here to solve the problem.

The solution to the model is represented as a vector, called coding. The output power of BESS and blade angle of wind machine can be coded in a vector. In this research, each vector is a matrix with fixed-size 2A, the coding rules are showed in Fig. 6-7.

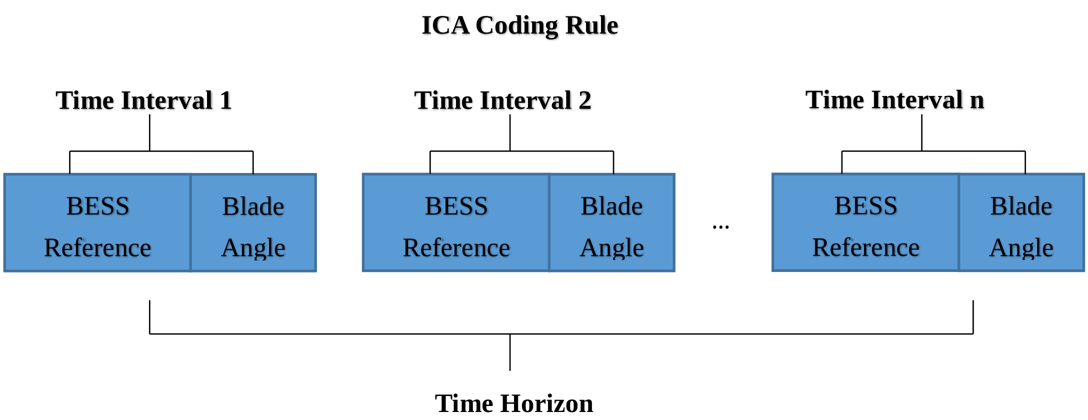


Fig. 6-7. Coding rules of the ICA solution method

6.5.2 Optimization Procedure

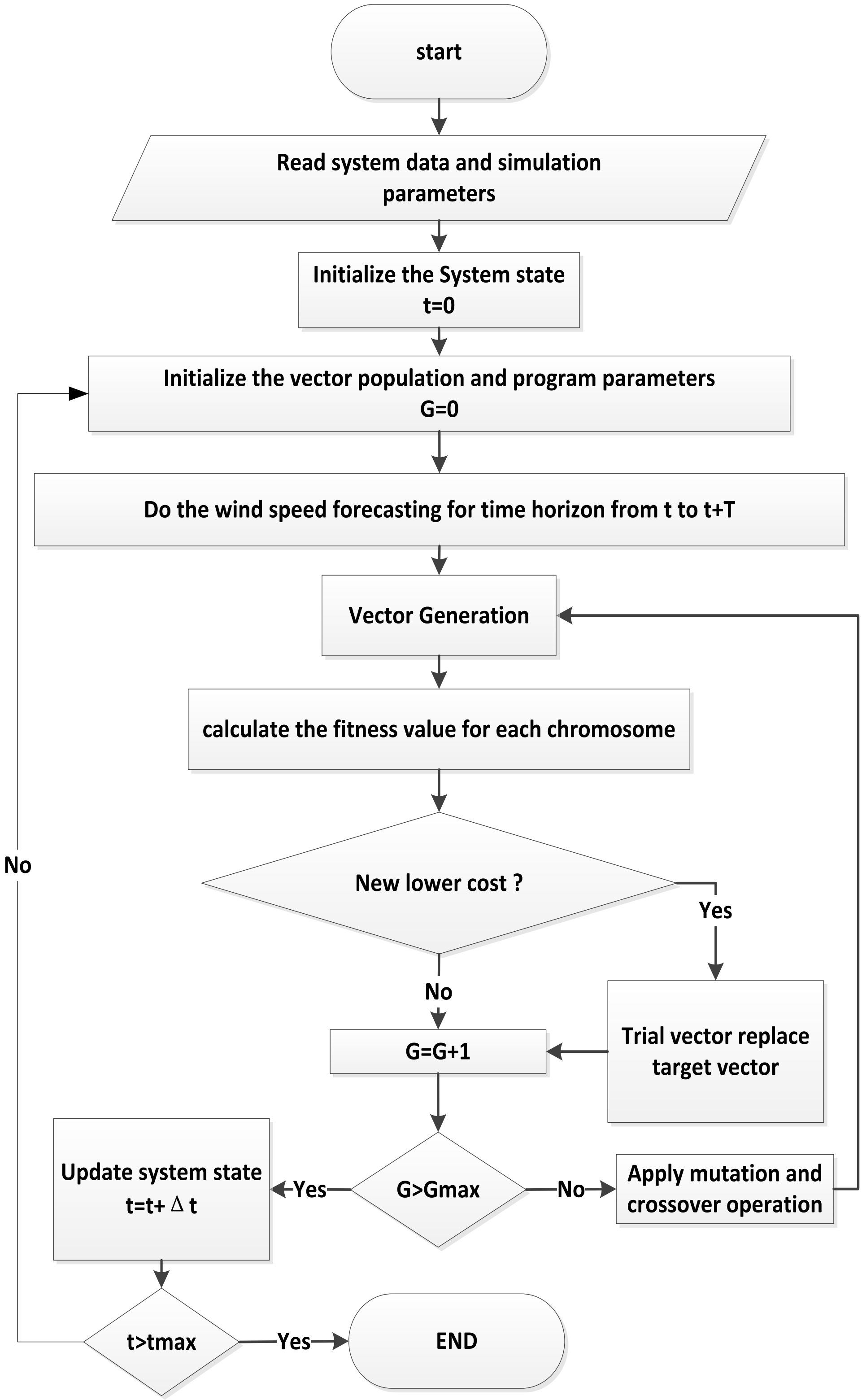


Fig. 6-8.Flow chart of solution program

During the process of calculating the fitness, the benefits part of the objective function is affected by the charging power of BESS and blade angle of wind machine. The flow chart is showed as Fig. 6-8.

6.6 Case Studies

6.6.1 Parameter Settings

The proposed methodology has been tested on a 30 MW wind farm simulation. The results testify the proposed scheduling methodology. The benefit of optimal BESS control scheme is calculated. After the simulation, the operation cost of the proposed optimal dispatch strategy is compared with the wind farm without BESS. The parameter settings are illustrated in Table 6-3.

Table 6-3. General Parameters of test system

Item	**Value**
Wind energy price p	0.05$/MWh
v_{start} / v_{cut}	4m/s / 25 m/s
Δt	10 mins
T	4 hrs
SOC^{Max} / SOC^{Min}	90% / 20%
Penalty price δ	0.5$/MWh
Wind farm capacity	30 MW
BESS capacity	10 MW
Power rating of BESS	3 MWh
Ramp limit	5MW/10mis
Initial wind power	0KW
Initial SOC of BESS	0.6
Battery type	Lead-ion battery

6.6.2 Case I: without Forecasting Error

In the first case, a 30MW wind farm is constructed with BESS. The detail parameters are shown as below. The ramp limit is adjusted to make the output of wind farm smoother.

First, the forecasting wind speed is assumed to be the same as real one. Fig. 6-9 shows the expected wind farm output of the wind farm and penalty cost without dispatch scheduling in a sample day.

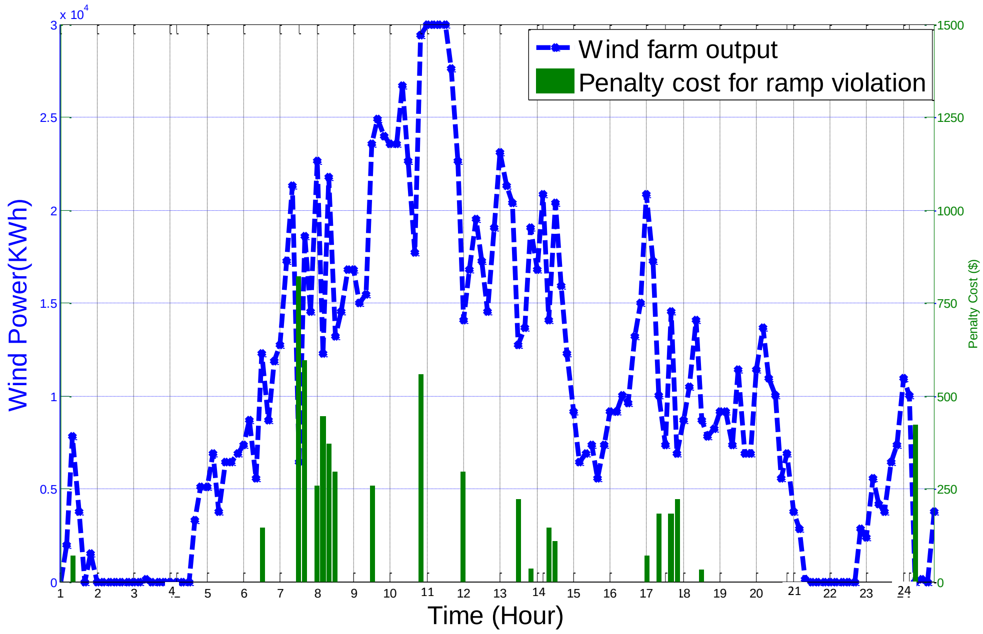


Fig. 6-9. Wind power and penalty cost of wind farm for one day

Table 6-4. Optimal results at step for time horizon

Time Interval	1	2	3	4	5	6
Charging Power / Wind Percentage	0/1	460/1	-430/1	0/1	0/1	0/1
Time Interval	7	8	9	10	11	12
Charging Power / Wind Percentage	0/1	0/1	0/1	0/1	0/1	0/1
Time Interval	13	14	15	16	17	18
Charging Power / Wind Percentage	0/1	0/1	0/1	0/1	0/1	0/1
Time Interval	19	20	21	22	23	24
Charging Power / Wind Percentage	0/1	0/1	0/1	0/1	0/1	0/1

Base on the above assumption that the forecasting data is accurate, the proposed wind farm dispatch scheduling method is proposed. During the first step, the optimal dispatch plan is set for time horizon. As shown in Table 6-4, the charging power and wind output percentage are optimized for minimum total operation cost. From the table, it can be found that through the power charging of battery at time interval 2, the penalty cost can be mitigated. During this horizon, the total forecast penalty cost is $71, the optimal dispatch scheme is $6.5. According to step 1, the battery will not be charged at first time interval. After the time interval, the new optimization problem is implemented with updated information. Based on this time horizon receding method, the scheduling for a whole day is calculated and the dispatch curve is drawn as Fig. 6-10.

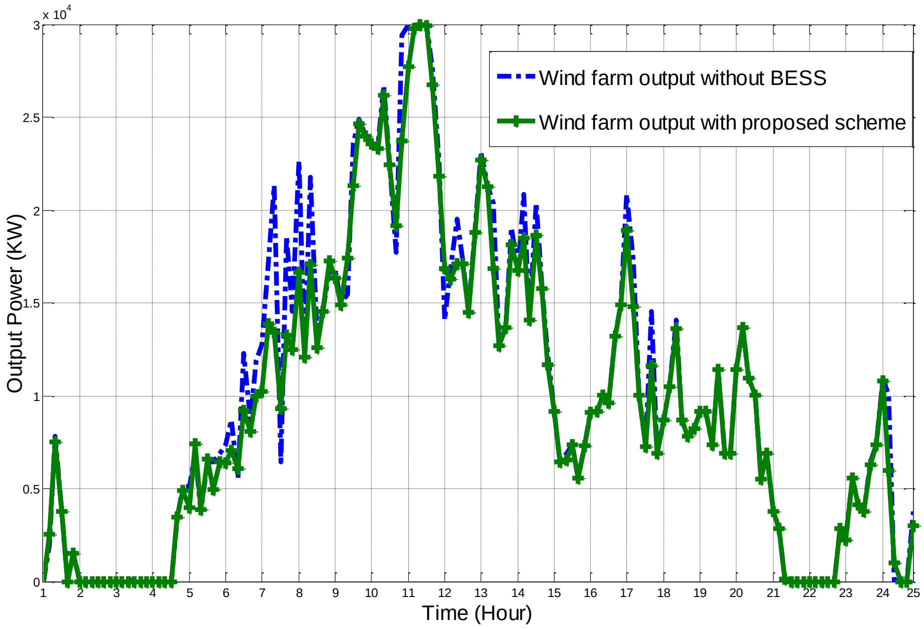


Fig. 6-10. Wind farm output comparison through a day

From the output curve, the proposed dispatch scheme, the output of the wind farm is smoother and better integrated into the grid. The operation cost of the wind farm is shown in Table 6-5. The results prove the feasibility when the forecasting wind speed is the same as real one.

Table 6-5. Operation cost comparison for Case I

Item (a sample day)	Without BESS	With Proposed Dispatch Scheme
Penalty Cost($)	5737	0
BESS Operation Cost	0	1863.6
Wind Cutting Loss	0	466.43
Total Cost	5737	2330.03

6.6.3 *Case II: with Forecasting Error*

In this case study, the parameters of 30MW wind farm is the same as before. The forecast error will be considered in this simulation. As shown in Fig. 6-2, the real wind speed is varied from the forecast one. In order to mitigate the impact of forecast error, the proposed dispatch scheme can update the system state and modify the dispatch schedule. Fig. 6-11 is the optimal wind farm dispatch plan at step 1 and step 2. After step 1, the schedule of first time interval is implemented and the system state is updated. Based on MPC model, the new dispatch problem in solved in step 2. It can be found that, the second plan is different to the first for the same time. The difference is caused by forecast error and the updating of time horizon.

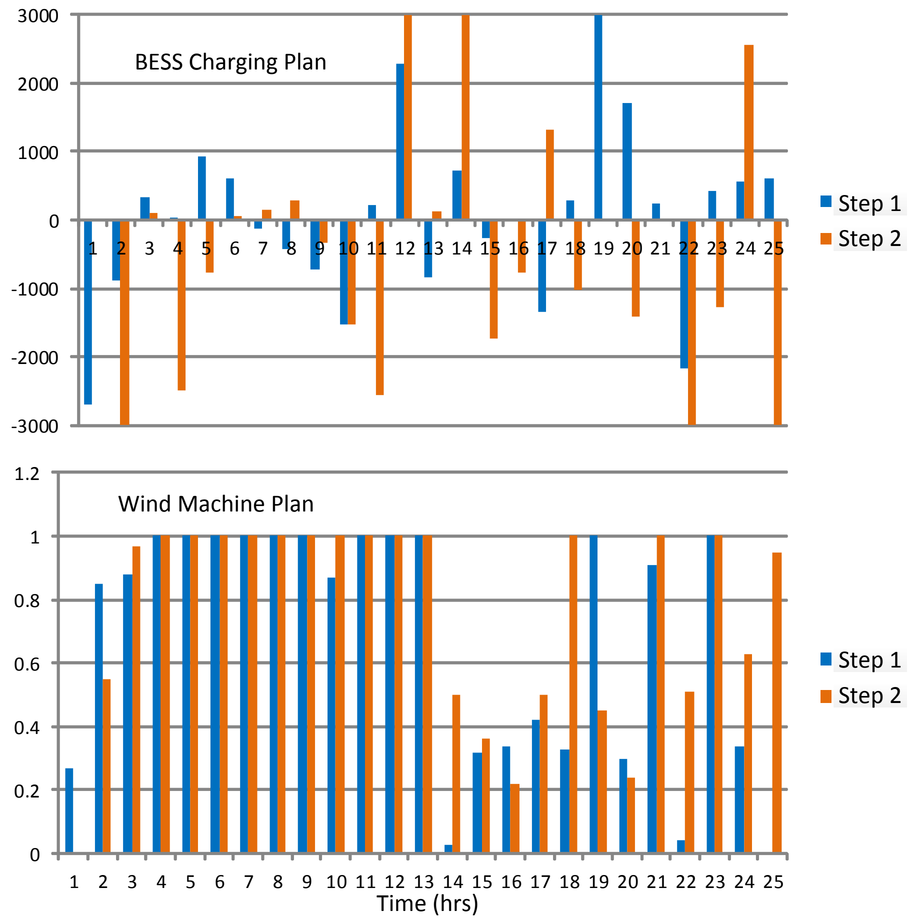


Fig. 6-11. Dispatch plan of step 1 and step 2 for time horizon

A full dispatch plan is performed in this case study. The historic wind speed data are utilized here for simulation. The forecast is done 4 hours in advance. The continuous operation plan and system state of a week is shown in following figures. Fig. 6-12 is the SOC curve of BESS through a week. The wind farm output with proposed dispatch scheme is compared with original wind power in Fig. 6-13. The detail operation cost of the wind farm per day is shown in Fig. 6-14. Total cost of the wind farm through a week is listed and compared in Table 6-6. From the table, the total saving through a week is $21263 which is 38.35% of original cost. The proposed method can increase the benefit of the wind farm and mitigate the risk of grid connected wind energy.

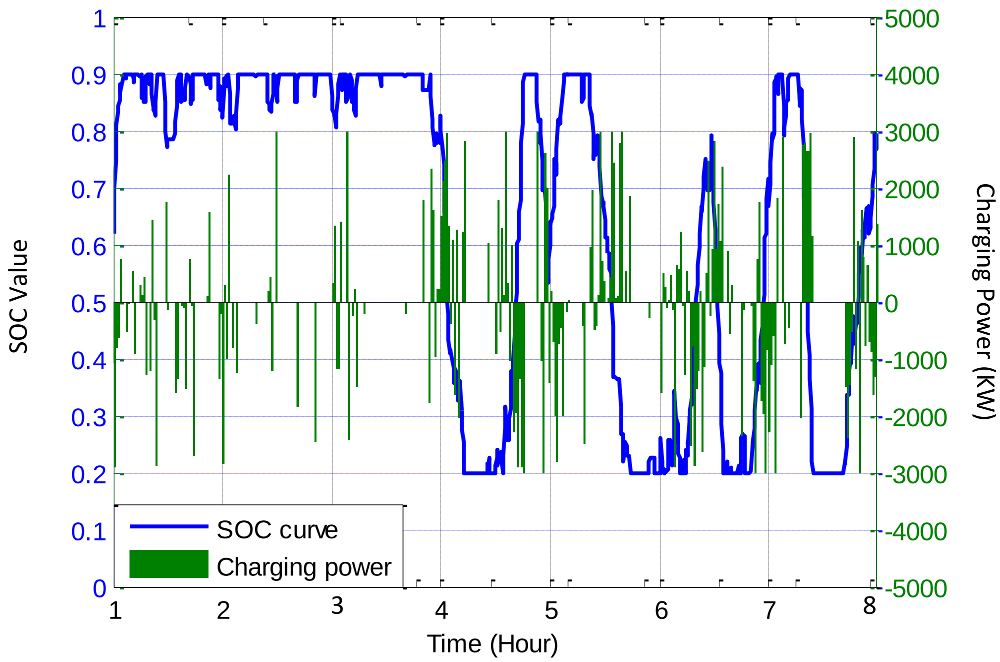


Fig. 6-12. Charging power and state of BESS

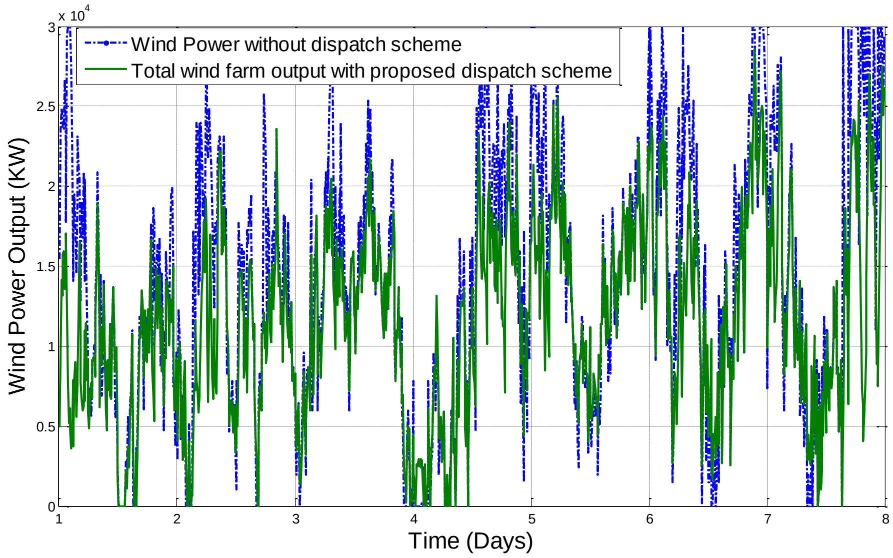


Fig. 6-13. Wind farm output comparison through a week

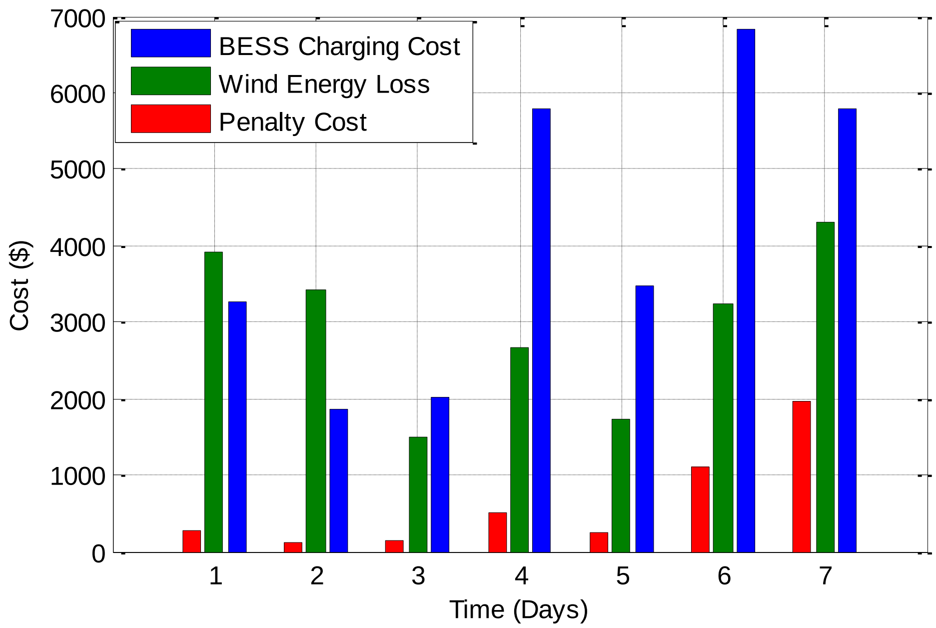


Fig. 6-14. Cost of the wind farm dispatch

Table 6-6. Operation cost comparison for Case II

Item (a week)	Without BESS	With Proposed Dispatch Scheme
Penalty Cost($)	55446	4388
BESS Operation Cost	0	19004
Wind Cutting Loss	0	10791
Total Cost	55446	34183

6.7 Summary

In this chapter, an optimal power dispatch scheduling is proposed for a wind farm with a battery energy storage system. The model predictive control-based (MPC) method is applied to operation decisions in order minimize the energy loss from wind farm and battery usage, while meeting grid constraints. The wind farm output data and battery state are updated continuously, and short-term power dispatch is scheduled by wind speed forecast within optimal horizon. The proposed optimization problem is solved by the distributed imperialist competitive algorithm (ICA). Finally, the novel short-term wind power dispatch scheme is demonstrated in a 30MW wind farm with historic data of wind speed.

Chapter 7

CONCLUSIONS AND FUTURE RESEARCH

7.1 Conclusions

Economic and environmental concerns are encouraging the application of renewable energy generation technologies, and the growing demand has sped up the development of renewable energy, especially wind power and solar energy. In recent years, the technology for wind power has become more mature and the penetration has been increasing rapidly around the world. However, the intermittent nature of wind power will affect the operation stability of the grid as more and more grid-connected wind farms are integrated into the system. In this situation, supplementary measures and operation strategies should be studied and employed to make the wind farm meet grid requirements.

This research developed a series of alternative and efficient computing methods and tools for the problems mentioned above. The major contributions are summarized as follows.

1) A computational grid platform for compute-intensive applications of modern power systems and a cloud computing platform for future power systems are proposed. Useful guidelines are drawn for power engineers to construct the practical grid system platform and cloud computing based information infrastructure.
2) A multi-constrained optimal power flow (MCOPF) model is proposed, which considers discrete control variables and practical operation constraints, including transient stability constraints, valve-point effects, prohibited operating zones of generators, and branch flow thermal limits. The simulation studies demonstrate that the algorithms utilized can address the limitations of conventional approaches, and that the classic differential evolution algorithm can solve this non-linear, non-convex, discrete, and non-differentiable optimization problem.
3) A cloud computing based platform for the MCOPF problem is proposed. The platform structure is presented and the simulation shows its practicability for large-scale power systems.
4) A new and efficient collector system layout optimization (CSLO) model for large-scale offshore wind farms is proposed, which considers multiple substations and cable types, and minimizes the annualized investment cost, maintenance cost, and the levelized power losses cost, while satisfying network reliability and operational constraints. The proposed SAA method can automatically obtain the locations of substations in a wind farm and allocate wind

turbines exclusively to their nearest substations. In addition, a Benders' decomposition based method is proposed to solve the CSLO model, and the simulation results show that the proposed CSLO model can be utilized as a reliable collector system layout tool for large-scale offshore wind farms.

5) An optimal power dispatch scheduling model for a wind farm integrated power system with battery energy storage system (BESS) is proposed. The development of BESS enables generation of renewable energy with flexible operation to meet the requirements of the electric grid. A wind speed forecasting method based on the forecasting software, *OptiWind*, which customizes numerical weather prediction models and the latest statistical methods, is also presented

6) A distributed imperialist competitive (ICA) algorithm is proposed to solve the proposed optimization model. A model predictive control (MPC) based method is also applied to operation decisions to minimize the energy loss from wind farm and battery usage. The simulation studies demonstrate that with BESS and the proposed policies and methods, the wind farm operator can make beneficial decisions through the proposed optimal model.

7.2 Future Research

The following research is expected to be conducted as a continuation of this thesis.

7.2.1 Computing Platform Improvement

With growing economies and populations, and concurrent greenhouse gas restrictions, electrical power systems play a crucial role in meeting the challenge of global energy demands, especially with the penetration of renewable energies such as wind power. We need to develop the capability to handle the large volumes of data generated by power system components and data acquisition devices like SCADA, PMUs, as well as the capacity to process these data at high resolution to ensure a reliable, stable, and secure power system network. Advanced research on the development and implementation of leading-edge computing platforms, and the related technologies and algorithms, will be required for future development.

This research proposed a computational grid platform for compute-intensive applications of modern power systems, and presented useful guidelines for electrical engineers and researchers on how to construct the practical platforms. However, many more simulations and analyses of the platform techniques on the test system are needed in order to insure that the future computing platforms will not endanger system stability and security. It should be clear that the test environment simulated in this research is simple because of the lack of data about future networks with large-scale wind farms integration. Great efforts are needed to perform simulations with more detailed data of the interconnected power systems.

7.2.2 CSLO Model Extension

In this research, a new and efficient collector system layout optimization (CSLO) model was developed for large-scale offshore wind farms, which considers multiple substations and cable types, and focuses on the cable topology optimization among wind turbines and substations. For the future work, the wake effect and the terrain of wind farms should be considered in order to improve the proposed model, from which more accurate simulation results could be expected.

7.2.3 Reliability Consideration

There is a growing concern over the effect of deregulation on the reliability of future power systems. In further research, more operational constraints on the grid should be involved in the optimal power dispatch model proposed in this research. Instead of assuming a selected set of cases beforehand, it is recommended that power interruptions, frequency control, and voltage control (reactive power) issues be considered.

7.2.4 Wind Speed Data Refinement

As indicated in Chapter 6, wind speed data for the optimal power dispatch model are developed based on the proposed forecasting software, *OptiWind*. In future work, the *OptiWind* tool should be improved with more realistic wind speed information for each wind farm, with various characteristics prepared for future dynamic simulations that consider both onshore and offshore wind farms installed in different locations. Interconnecting multiple offshore wind farms to a bulk power system is another challenge for future research to address.

BIBLIOGRAPHY

[1] F. Yao, Z.Y. Dong, K. Meng, Z. Xu, H. Iu, and K.P. Wong, “Quantum-inspired particle swarm optimization for power system operations considering wind power uncertainty and carbon tax in Australia,” *IEEE Trans. Industrial Informatics*, vol. 8, no. 4, pp. 880-888, Nov. 2012.

[2] Danish Ministry of Transport and Energy, Energistrategi 2025 (Energy Strategy 2025), 2007.

[3] Z.Y. Dong, K.P. Wong, K. Meng, F.J. Luo, F. Yao, and J.H. Zhao, “Wind power impact on system operations and planning,” *IEEE PES Gen. Meeting*, Minneapolis, USA, Jul. 2010.

[4] Y.V. Makarov, C. Loutan, J. Ma, and P. de Mello, “Operational impacts of wind generation on california power systems,” *IEEE Trans. Power Syst.*, vol. 24, no. 2, pp. 1039-1050, May 2009.

[5] J. Kaboouris and F.D. Kanellos, “Impacts of large-scale wind penetration on designing and operation of electric power systems,” *IEEE Trans. Sustain. Energy*, vol. 1, no. 2, pp. 107-114, Jul. 2010.

[6] T. Bookman, “Wind energy’s promise, offshore,” *IEEE Technol. Soc. Mag.*, vol. 24, no. 2, pp. 9-15, Jun. 2005.

[7] LORC. Offshore Wind Statistics. http://www.lorc.dk/offshore-wind-farms-map/statistics. Accessed 5 Dec 2013.

[8] R.E. Torres-Olguin, M. Molinas, and T. Undeland, “Offshore wind farm grid integration by VSC technology with LCC-based HVDC transmission,” *IEEE Trans. Sustain. Energy*, vol. 3, no. 4, pp. 899-907, Oct. 2012.

[9] P. Fuglsang and K. Thomsen, “Cost optimization of wind turbines for large scale off-shore wind farms,” Riso National Laboratory, Roskilde, 1998. http:// orbit.dtu.dk/fedora/objects/orbit:87965/datastreams/file_7731717/content. Accessed 5 Dec 2013.

[10] S. Dutta and T.J. Overbye, “Optimal wind farm collector system topology design considering total trenching length,” *IEEE Trans. Sustain. Energy*, vol. 3, no. 3, pp. 339-348, Jul. 2012.

[11] E.H. Camm, M.R. Behnke, O. Bolado, M. Bollen, M. Bradt, C. Brooks, W. Dilling, M. Edds, W.J. Hejdak, D. Houseman, S. Klein, F. Li, J. Li, P. Maibach, T. Nicolai, J. Patino, S.V. Pasupulati, N. Samaan, S. Saylors, T. Siebert, T. Smith, M. Starke, R. Walling, "Wind power plant collector system design considerations," *in Proc. 2009 IEEE Power Eng. Soc. General Meeting*, Calgary, Canada, Jul. 2009.

[12] M. Lange and U. Focken, Physical Approach to Short-Term Wind Power Prediction. New York: Springer, 2005.

[13] B. Fox et al., Wind Power Integration---Connection and System Operational Aspects. London: IET, 2007.

[14] A.C. Johnson, J.A. Hynds, D.R. Nevius, G.A. Nunan, and N. Sweetman, "Economic assessment of the utilization of lead-acid batteries in electric utility systems," U.S. Department of Energy Contract Number EX-76-C-02-2857, Nov. 1977.

[15] J.L. Delmonaco, P.A. Lewis, H.T. Roman, and J. Zemkoski, "Advanced batteries for load-leveling, the utility perspective on system integration," *IEEE Trans. Power Apparatus and Systems*, vol. PAS-101, no. 9, pp. 3315-3321, Sep. 1982.

[16] T.Y. Lee and N. Chen, "The effect of pumped storage and battery energy storage systems on hydrothermal generation coordination," *IEEE Trans. Energy Convers.*, vol. 7, no. 4, pp. 631-637, Dec. 1992.

[17] C.F. Lu, C.C. Liu, and C.J. Wu, "Dynamic modeling of battery energy storage system and application to power system stability," *IEE Proc. Gen. Trans. Dist.*, vol. 142, no. 4, pp. 429-435, Jul. 1995.

[18] T.Y. Lee and N.M. Chen, "Determination of optimal contract capacities and optimal sizes of battery energy storage systems for time-of-use rates industrial customers," *IEEE Trans. Energy Convers.*, vol. 10, no. 3, pp. 562-568, Sep. 1995.

[19] K.K. Leung and D. Sutanto, "Storage power flow controller using battery storage," *IEE Proc. Gen. Trans. Dist.*, vol. 150, no. 6, pp. 727-735, Nov. 2003.

[20] A. Oudalov, D. Chartouni, and C. Ohler, "Optimizing a battery energy storage system for primary frequency control," *IEEE Trans. Power Syst.*, vol. 22, no. 3, pp. 1259-1266, Aug. 2007.

[21] P. Mercier, R. Cherkaoui, and A. Oudalov, "Optimizing a battery energy storage system for frequency control application in an isolated power system," *IEEE Trans. Power Syst.*, vol. 24, no. 3, pp. 1469-1477, Aug. 2009.

[22] Q. Wang and S.S. Choi, "The design of battery energy storage system in a unified power-flow control scheme", *IEEE Trans. Power Delivery*, vol. 23, no. 2, pp. 1015-1024,

Apr. 2008.

[23] A. Arulampalam, M. Barnes, N. Jenkins, and J.B. Ekanayake, "Power quality and stability improvement of a wind farm using STATCOM supported with hybrid battery energy storage," *in Proc. Inst. Electr. Eng. Gen., Transmiss. Distrib.*, vol. 153, no. 6, pp. 701-710, Nov. 2006.

[24] J. Zeng, B. Zhang, C. Mao, and Y. Wang, "Use of battery energy storage system to improve the power quality and stability of wind farms," *in Proc. Int. Conf. Power Syst. Technol.*, 2006.

[25] K. Yoshimoto, T. Nanahara, and G. Koshimizu, "New control method for regulating state-of-charge of a battery in hybrid wind power/battery energy storage system," *in Proc. IEEE Power Syst. Conf. Expo.*, pp.1244-1251, Oct. 29-Nov. 1, 2006,.

[26] S. Teleke, M.E. Baran, A.Q. Huang, S. Bhattacharya, and L. Anderson, "Control strategies for battery energy storage for wind farm dispatching," *IEEE Trans. Energy Convers.*, vol. 24, no. 3, pp. 725-732, Sep. 2009.

[27] S. Teleke, M.E. Baran, S. Bhattacharya, and A.Q. Huang, "Optimal control of battery energy storage for wind farm dispatching," *IEEE Trans. Energy Convers.*, vol. 25, no. 3, pp. 787-794, Sep. 2010.

[28] D.L. Yao, S.S. Choi, K.J. Tseng, and T.T. Lie, "A statistical approach to the design of a dispatchable wind power-battery energy storage system," *IEEE Trans. Energy Convers.*, vol. 24, no. 4, pp. 916-925, Dec. 2009.

[29] D.L. Yao, S.S. Choi, K.J. Tseng, and T.T. Lie, "Determination of short-term power dispatch schedule for a wind farm incorporated with dual-battery energy storage scheme," *IEEE Trans. Sustain. Energy*, vol. 3, no. 1, pp. 74-84, Jan. 2012.

[30] Q. Li, S.S. Choi, Y. Yuan, and D.L. Yao, "On the determination of battery energy storage capacity and short-term power dispatch of a wind farm," *IEEE Trans. Sustain. Energy*, vol. 2, no. 2, pp. 148-158, Apr. 2011.

[31] B. Hartmann and A. Dan, "Cooperation of a grid-connected wind farm and an energy storage unit---demostration of a simulation tool," *IEEE Trans. Sustain. Energy*, vol.3, no. 1, pp. 49-56, Jan. 2012.

[32] John A. Turner. "A Realizable Renewable Energy Future". *Science*, vol. 285, no. 5428, pp: 687 – 689, July 1999.

[33] K. Agbossou, R. Chahine, J. Hamelin, F. Laurencelle, A. Anouar, J. -M. St-Arnaud and T. K. Bose. "Renewable Energy Systems Based on Hydrogen for Remote Applications," *Journal of Power Sources*, vol. 96, pp: 168-172, 2001.

[34] Y. Ma, Long Zhou, Norman Tse, Abdi Osman, L L Lai. “An initial computational intelligence for smart grid,” *in Proceedings of the Eighth International Conference on Machine Learning and Cybernetics*, Baoding, pp: 3425-3429, 12-15 July 2009.

[35] J. Shu. Wei Xue, and Weimin Zheng. “A parallel transient stability simulation for power systems,” *IEEE Trans. Power Syst.*, vol. 20, no. 4, pp. 1709-1717, Nov. 2005.

[36] J.A. Hollman, and J.R. Marti. “Real time network simulation with pc-cluster,” *IEEE Trans. Power Syst.*, vol. 18, no. 2, pp. 563-569, May 2003.

[37] J.A. Hollman, and J.R. Marti. “Step-by-step eigenvalue analysis with emtp discrete-time solutions,” *IEEE Trans. Power Syst.*, vol. 25, no. 3, pp. 1220-1231, Aug. 2010.

[38] I. Foster, C. Kesselman, and S. Tuecke. “The anatomy of the Grid: enabling scalable virtual organizations,” *Int. J. High Performance Computing Applications.*, vol. 15, no. 3, pp. 200-222, Aug. 2011.

[39] Ian Foster, Carl Kesselman. “The grid: blueprint for a new computing infrastructure”. Morgan Kaufmann Publishers, 1998.

[40] M. Ali, Z. Dong and P. Zhang. “Adoptability of Grid Computing technology in power systems analysis, operations and control,” *IET Gens. Transm. Distrib.*, vol. 3, no. 10, pp. 949-959, Oct. 2009.

[41] I. Foster, Y. Zhao, and I. Raicu, etc, “Cloud computing and grid computing 360-degree compared,” Grid Computing Environments Workshop, Austin, pp. 1-10, 2008.

[42] AP Engelbrecht, “Computational Intelligence an Introduction”, John Willey & Sous, Ltd.2007.

[43] G.K Venayagamoorthy, “Adaptive critic design based neurocontrollers for turbogenerator control,” Report to the IEEE Neural Network Society. http://www.okstate.edu/elec-engr/faculty/yen/nntc/new/v3n2/kumar.pdf. Accessed 9 Dec 2013.

[44] S.P.Teeuwsen,I.Erlich,M.A.EI-Sharkawi,U.Bachmann, “Genetic Algorithm and decision tree -based Oscillatory stability Assessment,” *IEEE Trans. Power Syst.*, vol 21, no. 2, pp. 746-753, May 2006.

[45] J. Kennedy, R. Eberhart, “Particle Swarm Optimization,” *in Proceedings of IEEE International Conference on Neural Networks*, IV. pp. 1942–1948, Nov/Dec 1995.

[46] Ian Foster. “What is the Grid? A Three Point Checklist”. http://www.mcs.anl.gov/~itf/Articles/WhatIsTheGrid.pdf. Accessed 1 Dec 2013.

[47] Frédéric Magoulès, Jie Pan, Kiat-An Tan and Abhinit Kumar, “Introduction to Grid Computing”. Boca Raton : CRC Press, c2009.

[48] Ian Foster, Carl Kesselman, “The grid : blueprint for a new computing infrastructure”. Morgan Kaufmann Publishers, 1998.

[49] Malcolm Irving, Gareth Taylor, Peter Hobson, “Plug in to grid computing,” *IEEE Power and Energy Magazine*, vol. 2, no. 2, pp. 40-44, Mar.-Apr. 2004.

[50] Globus Toolkit. http://www.globus.org/. Accessed 1 Dec 2013.

[51] gLite. http://glite.web.cern.ch/glite/. Accessed 1 Dec 2013.

[52] BONIC. http://boinc.berkeley.edu/. Accessed 1 Dec 2013.

[53] M. Ali, Z. Y. Dong, X. Li and P. Zhang. “RSA-Grid: a grid computing based framework for power system reliability and security,” *IEEE Power Engineering Society General Meeting*, 2006.

[54] M Ali, Z.Y. Dong and P. Zhang, “Adoptability of grid computing in power systems analysis, operations and control,” *IET Gen Trans and Dist.*, vol. 3, no. 4, pp.949-959, Oct. 2009.

[55] Malcolm Irving and Gareth Taylor, “Prospects for grid-computing in future power networks”. A discussion paper circulated by Brunel Institute of Power Systems, 2003.

[56] G. A. Taylor, M. R. Irving, P. R. Hobson, C. Huang, P. Kyberd and R. J. Taylor, “Distributed monitoring and control of future power systems via Grid Computing,” *IEEE Power Engineering Society General Meeting*, 2006.

[57] Qi Huang, Kaiyu Qin, Wenyong Wang, “Development of a grid computing platform for electric power system applications,” *IEEE Power Engineering Society General Meeting*, 2006.

[58] GRIDCC EU ICT Project. “Grid Enabled Remote Instrumentation with Distributed Control and Computation”. www.gridcc.org. Accessed 21 Nov 2013.

[59] Djalma M. Falcão, “High performance computing in power system applications”. Vector and Parallel Processing-VECPAR'96, vol.1215, pp. 1-23, Springer, 1997.

[60] C.H. Liang, C.Y. Chung and K.P. Wong and X.Z. Duan, “Parallel optimal reactive power flow based on cooperative co-evolutionary differential evolution and power system decomposition,” *IEEE Trans. Power Syst.*, vol. 22, no. 1, pp. 249-257, Feb. 2007.

[61] Ch'i-Hsin Lin, Shin-Yeu Lin, “Distributed optimal power flow with discrete control variables of large distributed power systems,” *IEEE Trans. Power Syst.*, vol. 23, no. 3, pp. 1383-1392, Jul. 2008.

[62] Barker, P.P., De Mello, and R.W., “Determining the impact of distributed generation on power systems. I. Radial distribution systems,” *IEEE Power Engineering Society Summer Meeting*, vol. 3, pp.1645-1656, 2000.

[63] Cristaldi, L., Ferrero, A., Salicone, S., “A distributed system for electric power quality measurement,” *IEEE Transactions on Instrumentation and Measurement*, vol 51, no. 4,

pp. 776 – 781, Aug. 2002.

[64] T.O. Ting, K.P. Wong and C.Y. Chung, "Hybrid constrained genetic algorithm / particle swarm optimisation load flow algorithm," *IET Gen Trans and Dist.*, vol. 2, no. 6, pp. 800-812, Nov. 2008.

[65] C.H. Liang, C.Y. Chung, K.P. Wong, X.Z. Duan and C.T. Tse, "Study of differential evolution for optimal reactive power flow," *IET Gen Trans and Dist.*, vol. 1, no. 2, pp. 253 -260, Mar. 2007.

[66] J. Yu, W. Yan, W. Li, C.Y. Chung and K.P. Wong, "An unfixed piecewise-optimal reactive power-flow model and its algorithm for AC-DC systems," *IEEE Trans. Power Syst.*, vol. 23, no. 1, pp. 170-176, Feb. 2008.

[67] Yan Wei, Liu Fang, C.Y. Chung and K.P. Wong, "A hybrid strategy based on GA and IPM for optimal reactive power flow," *IEEE Trans. Power Syst.*, vol. 21, no. 3, pp. 1163-1169, Aug. 2006.

[68] C.H. Liang, C.Y. Chung, K.P. Wong and X.Z. Duan, "Comparison and improvement of evolutionary programming techniques for optimal reactive power flow," *IET Gen Trans and Dist.*, vol. 153, no. 2, pp. 228 – 236, Mar. 2006.

[69] H.R. Cai, C.Y. Chung and K.P. Wong, "Application of differential evolution algorithm for transient stability constrained optimal power flow," *IEEE Trans. Power Syst.*, vol. 23, no. 2, pp. 719-728, May 2008.

[70] Z. Wang, C.Y. Chung, K.P. Wong, C.T. Tse, and K.W. Wang, "Robust power system stabilizer design under multioperating conditions using differential evolution," *IET Gen Trans and Dist.*, vol. 2, no. 5, pp. 690-700, Sep. 2008.

[71] H. Yu, C.Y. Chung, K.P. Wong and J.H. Zhang, "A chance constrained transmission network expansion planning method with consideration of load and wind farm uncertainties," *IEEE Trans. Power Syst.*, vol. 24, pp. 1568-1576, Aug. 2009.

[72] T.W. Lau, C.Y. Chung, K.P. Wong, T.S. Chung and S.L. Ho, "Quantum-inspired evolutionary algorithm approach for unit commitment," *IEEE Trans. Power Syst.*, Vol. 24, pp. 1503-1512, Aug. 2009.

[73] Lu Gang, C.Y. Chung, K.P. Wong and F.S.Wen, "Unit maintenance scheduling coordination mechanism in electricity market environment," *IET Gen Trans and Dist.*, vol. 2, no. 5, pp. 646-654, Sep. 2008.

[74] S.X. Zhang, C.Y. Chung, K.P. Wong and H. Chen, "Analyzing two-settlement electricity market equilibrium by coevolutionary computation approach," IEEE Trans. on Power Systems, Vol. 24, pp. 1155-1164, Aug. 2009.

[75] Wang Yong, K. P. Wong, C.Y. Chung and F.S. Wen, “Determination of appropriate price level in installed-capacity market,” IET Generation, Transmission and Distribution, Vol. 1, Issue 2, pp. 127 – 132, Jan.2007.

[76] H. Chen, K.P. Wong, C.Y. Chung and D.H.M. Nguyen, “A coevolutionary approach to analyzing supply function equilibrium model,” *IEEE Trans. Power Syst.*, vol. 21, no. 3, pp. 1019-1028, Aug. 2006.

[77] H. Chen, K.P. Wong, D. H. M. Nguyen and C.Y. Chung, “Analyzing oligopoly electricity market using coevolutionary computation,” *IEEE Trans. Power Syst.*, vol. 21, no. 1, pp. 143-152, Feb. 2006.

[78] P. Kundur. Power System Stability and Control. McGraw-Hill, New York, 1994.

[79] R. Al-Khannak, L. Ye,“Integrating grid computing technology for developing power systems reliagility and efficiency,” *12th WSEAS International Conference on SYSTEMS*, Heraklion, Greece, pp. 491-497, Jul. 22-24, 2008.

[80] M. Ali, Z. Y. Dong, X. Li and P. Zhang, “Applications of grid computing in power systems,” *Proceedings of the Australasian Universities Power Engineering Conference*, vol. 2, pp. 396-401, 2005.

[81] R. Al-Khannak, B. Bitzer. “Load balancing for distributed and integrated power systems using Grid Computing”. ICCEP07, 2007.

[82] Costas G. Baslis, Stylianos E. Papadakis and Anastasios G. Bakirtzis, “Simulation of optimal medium-term hydro-thermal system operation by Grid Computing,” *IEEE Trans. Power Syst.*, vol.24, no.3, pp.1208-1217, 2009.

[83] SpeedTest. http://www.speedtest.net. Accessed 21 Nov 2013.

[84] Rajkuma Buyya. Personal Home Page. http://www.buyya.com/cv.html. Accessed 21 Nov 2013.

[85] K. Krauter, R. Buyya and M. Masheswaran, “A taxonomy and survey of Grid management systems for distributed computing,” *Int. J. Software: Practice and Experience*, vol. 32, no. 2, pp. 135-164, Feb. 2002.

[86] R. Buyya, “Economic-based distributed resource management and scheduling for grid computing,” Ph.D. dissertation, Dep. Information Technology, Monash Univ., 2002.

[87] R. Buyya, H. Stockinger, J. Giddy and D. Abramson. “Economic models for management of resources in peer-to-peer and Grid Computing,” *J. Concurrency and Computation: Practice and Experience.*, vol. 14, no. 13-15, pp. 1507-1542, Aug. 2002.

[88] R. Buyya and M. Murshed. “Gridsim: a toolkit for the modeling and simulation of distributed resource manamement and scheduling for Grid Computing,” *Concurrency*

and Computation: Practice and Experience., vol. 14, no. 13, pp. 1175-1220, Nov. 2002.

[89] I. Foster and C. Kesselman, “Globus: A metacomputing infrastructure toolkit,” *Int. J. Supercomputer Applications.*, vol. 11, no. 2, pp. 115-128, Jun. 1997.

[90] GridWay. http://www.gridway.org/doku.php. Accessed 21 Nov 2013.

[91] J. Novotny, S. Tuecke and V. Welch, “An online credential repository for the Grid: MyProxy,” *in Proc. 4th IEEE Int. Conf. High Performance Distributed Computing*, pp. 104-111, 2001.

[92] Torque. http://db.apache.org/torque/. Accessed 21 Nov 2013.

[93] Ganlia. http://ganglia.info/. Accessed 21 Nov 2013.

[94] Power system test case archive. http://www.ee.washington.edu/research/pstca/. Accessed 21 Nov 2013.

[95] Rajkuma Buyya. World Wide Grid. http://www.buyya.com/ecogrid/wwg/. Accessed 21 Nov 2013.

[96] I. Foster, Y. Zhao, and I. Raicu, etc, “Cloud computing and grid computing 360-degree compared,” Grid Computing Environments Workshop, Austin, pp. 1-10, 2008.

[97] R. Buyya, C. Yeo, and S. Venugopal, etc, “Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th Utility,” Future Gener. Comp. Syst., vol. 25, no. 6, pp. 599-616, Jun. 2009.

[98] R. Buyya, J. Broberg, and A. Goscinski, Cloud Comptuing: Principles and Paradigms. Wiley, Feb 2011.

[99] Google Data Centers. http://www.google.com/about/datacenters/. Accessed 21 Nov 2013.

[100] L. Barroso and U. Holzle, “The datacenter as a computer: an introduction to the design of warehouse-scale machines,” Synthesis Lectures on Computer Architecture, vol. 4, no. 1, pp. 1-108, 2009.

[101] Sato, K. “An Inside Look at Google BigQuery, White paper”. https://cloud. google.com/ files/BigQueryTechnicalWP.pdf. Accessed 21 Nov 2013.

[102] FI-WARE. http://www.fi-ware.eu/. Accessed 21 Nov 2013.

[103] Amazon EC2. http://aws.amazon.com/ec2/. Accessed 21 Nov 2013.

[104] OpenPBS. http://www.mcs.anl.gov/research/projects/openpbs/. Accessed 21 Nov 2013.

[105] Condor. http://research.cs.wisc.edu/htcondor/. Accessed 21 Nov 2013.

[106] MPICH. http://www.mpich.org/. Accessed 21 Nov 2013.

[107] J. Shu, W. Xue, and W.M. Zheng, “A parallel transient stability simulation for power systems,” *IEEE Trans. Power Syst.*, vol. 20, no. 4, pp. 1709-1717, Nov. 2005.

[108] J.A. Hollman and J.R. Marti, “Step-by-step Eigenvalue analysis with EMTP discrete-time solutions,” *IEEE Trans. Power Syst.*, vol. 25, no. 3, pp. 1220-1231, Aug 2010.

[109] K. Meng, Z.Y. Dong, K.P. Wong, Y. Xu, and F.J. Luo. “Speed-up the computing efficiency of power simulator for engineering-based power system transient stability simulations,” *IET Gen. Trans. & Dist.*, vol. 4, no. 5, pp. 652-661, May 2010.

[110] Google App Engine. http://code.google.com/intl/zh-CN/appengine/. Accessed 21 Nov. 2013.

[111] D. Sanderson, Programming Google App Engine. O’Reilly, Nov. 2009.

[112] Eclipse. http://www.eclipse.org/. Accessed 21 Nov. 2013.

[113] Y. Xu, Z.Y. Dong, L. Guan, R. Zhang, K.P. Wong and F. Luo, “Preventive dynamic security control of power systems based on pattern discovery technique,” *IEEE Trans. Power Syst.*, vol. 27, no. 3, pp. 1236-1244, Aug. 2012.

[114] M.R. Sikonja and I. Kononenko, “Theoretical and Empirical Analysis of ReliefF and RReliefF”, *Machine Learning*, vol. 53, pp. 23-69, 2003.

[115] D. Sanderson, Programming Google App Engine. O’Reilly, Nov. 2009.

[116] J.A. Momoh, M.E. El-Hawary, and R. Adapa, “A review of selected optimal power flow literature to 1993 I Nonlinear and quadratic programming approaches,” *IEEE T.PWRS.*, vol.14, no.1, pp.96–104, Feb. 1999.

[117] J.A. Momoh, M.E. El-Hawary, and R. Adapa, “A review of selected optimal power flow literature to 1993. II. Newton, linear programming and interior point methods,” *IEEE Trans. Power Syst.*, vol. 14, no. 1, pp. 105–111, Feb. 1999.

[118] D. Corne, M. Dorigo, and F. Glover, New Ideas in Optimization. London: McGraw-Hill Education, 1999.

[119] K. Price, R. Storn and J.A. Lampinen, Differential Evolution: A Practical Approach to Global Optimization. Springer, 2005.

[120] J. Vesterstrom and R. Thomsen, “A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems,” *in Proc. IEEE Congr. Evolut. Comput.*, Portland, OR, pp. 1980–1987, Jun. 20–23, 2004.

[121] D.K. Tasoulis, N.G. Pavlidis, V.P. Plagianakos, and M.N. Vrahatis, “Parallel differential evolution,” *in Proc. IEEE Congr. Evolut. Comput*, pp. 2023-2029, Portland, Jun. 2004.

[122] G.Y Yang, Z.Y. Dong, and K.P. Wong, “A modified differential evolution algorithm with fitness sharing for power system planning,” *IEEE Trans. Power Syst.*, vol. 23, no. 2, pp. 514-522, May 2008.

[123] J.H. Van Sickel, K.Y. Lee and J.S. Heo, “Differential evolution and its applications to power plant control,” *in Proc. Intelligent Systems Applications to Power Systems 2007*, pp. 1-6, Nov. 2007.

[124] A.K. Qin, V.L. Huang and P.N. Suganthan, “Differential evolution algorithm with strategy adaptation for global numerical optimization,” *IEEE Trans. Evolut. Comput.*, vol. 13, no. 2, pp. 398-417, Apr. 2009.

[125] K. Meng, H.G. Wang, Z.Y. Dong, and K.P. Wong, “Quantum-inspired particle swarm optimization for valve-point economic load dispatch,” *IEEE Trans. Power Syst.*, vol. 25, no. 1, pp. 215-222, Feb. 2010.

[126] P. W. Sauer and M. A. Pai, Power System Dynamics and Stability. Englewood Cliffs, NJ: Prentice-Hall, 1998.

[127] A.A. Fouad and V. Vittal, Power System Transient Stability Analysis using the Transient Energy Function Method. Englewood Cliffs, NJ: Prentice Hall, 1986.

[128] J.A. Momoh, R.J. Koessler, M.S. Bond, B. Stott, D. Sun, A. Papalexopoulos, and P. Ristanovic, “Challenges to optimal power flow,” *IEEE Trans. Power Syst.*, vol. 12, pp. 444-455, Feb. 1997.

[129] D. Gan, R.J. Thomas, and R.D. Zimmerman, “Stability-constrained optimal power flow,” *IEEE Trans. PWRS.*, vol.15, pp.535-540, May 2000.

[130] Y. Yuan, J. Kubokawa, and H. Sasaki, “A solution of optimal power flow with multicontingency transient stability constraints,” *IEEE Trans. Power Syst.*, vol. 18, no. 3, pp. 1094-1102, Aug. 2003.

[131] D. Layden and B. Jeyasurya, “Integrating security constraints in optimal power flow studies,” *in Proc. IEEE PES GM’04*, pp.125-129, Jun.2004.

[132] L. Chen, Y. Taka, H. Okamoto, R. Tanabe, and A. Ono, “Optimal operation solutions of power systems with transient stability constraints,” *IEEE Trans. Circuits Syst. I*, vol. 48, no. 3, pp. 327-339, Mar. 2001.

[133] G.A. Maria, C. Tang, and J. Kim, “Hybrid transient stability analysis,” *IEEE Trans. Power Syst.*, vol. 5, no. 2, pp. 384-393, May 1990.

[134] E.M. Montes, J.V. Reyes, and C.A. Coello, “A comparative study of differential evolution variants for global optimization”, i*n Proc. Genetic Evol. Comput. Conf.* (GECCO 2006), pp. 485-492, Jul. 2006.

[135] H.R. Tizhoosh, “Opposition-based learning: a new scheme for machine intelligence,” *in Int. Conf. on Computational Intelligence for Modelling Control and Automation* (CIMCA’2005), Vienna, Austria, vol.1, pp. 695-701, 2005.

[136] S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, “Opposition-based differential Evolution,” *IEEE Trans. Evol. Comput.*, vol. 12, no. 1,pp. 64–79, Feb. 2008.

[137] H. Wang, Y. Liu, S. Y. Zeng, H. Li and C. H. Li, “Opposition-based particle swarm algorithm with Cauchy mutation,” *in Proc. Congr. Evol. Comput.*, pp. 4750–4756, 2007.

[138] R. D. Zimmerman, C. E. Murillo-Sánchez, and D. Gan, “MATPOWER, a MATLAB power system simulation package,” Power Systems Engineering Research Center, Cornell University, 1997. http://www.pserc.cornell.edu/matpower/. Accessed 25 Nov. 2013.

[139] A.G. Bakirtzis, P.N. Biskas, C.E. Zoumas, and V. Petridis, “Optimal power flow by enhanced genetic algorithm,” *IEEE Trans. Power Syst.*, vol. 17, no.2, pp. 229-236, May 2002.

[140] J. Yuryevich and K.P. Wong, “Evolutionary programming based optimal power flow algorithm,” *IEEE Trans. Power Syst.*, vol. 14, no. 4, pp. 1245-1250, Nov. 1999.

[141] N. Mo, Z.Y. Zou, K.W. Chan, and T.Y.G. Pong, “Transient stability constrained optimal power flow using particle swarm optimisation,” *IET Gen., Transm., Distrib.*, vol. 1, no. 3, pp. 476–483, May 2007.

[142] M.A. Pai, Energy Function Analysis for Power System Stability. Norwell, MA: Kluwer, 1989.

[143] Power system test case archive. http://www.ee.wash-ington.edu/research/pstca/. Accessed 25 Nov. 2013.

[144] T. Kwok, K. Smith, S. Lozano, and D. Taniar, “Parallel fuzzy c-means clustering for large data sets,” *In B. Monien and R. Feldmann*, (Eds.), EURO-PAR, LNCS vol. 2400, pp. 365-374, 2002.

[145] Y. Ma, T. Runolfsson, and J.N. Jiang, “Cluster analysis of wind turbines of large wind farm with diffusion distance method,” *IET Renew. Power Gen.*, vol. 5, no.2, pp. 109-116, Mar. 2011.

[146] K. Meng, Z.Y. Dong, and K.P. Wong, “Self-adaptive radial basis function neural network for short-term electricity price forecasting,” *IET Gen., Transm., Distrib.*, vol. 3, no. 4, pp. 325-335, Apr. 2009.

[147] H. Salazar, R. Gallego, and R. Romero, “Artificial neural networks and clustering techniques applied in the reconfiguration of distribution systems,” *IEEE Trans. Power Deliver.*, vol. 21, no. 3, pp. 1735-1742, Jul. 2006.

[148] M. Ramezani, C. Singh, and M.R. Haghifam, “Role of clustering in the probabilistic evaluation of TTC in power systems including wind power generation,” *IEEE Trans.*

Power Syst., vol. 24, no. 2, pp. 849-858, May 2009.

[149] R.O. Duda and P.E. Hart, Pattern Classification and Scene Analysis. Wiley, New York, 1974.

[150] J. Bezdek, R. Ehrlich, and W. Full, "FCM: The fuzzy c-means clustering algorithm," *Comput. Geosci.*, vol. 10, no. 2-3, pp. 191-203, 1984.

[151] J.M. Lu, X. Yuan, and T. Yahagi, "A method of face recognition based on fuzzy c-means clustering and associated sub-NNs," *IEEE Trans. Neural Networks*, vol. 18, no. 1, pp. 150-160, Jan. 2007.

[152] C. Hwang and F.C.H. Rhee, "Uncertain fuzzy clustering: interval type-2 fuzzy approach to c-means," *IEEE Trans. Fuzzy Systems*, vol. 15, no. 1, pp. 107-120, Feb. 2007.

[153] D. Halperin, M.H. Overmars, and M. Sharir, "Efficient motion planning for an L-shaped object," *SIAM J. Comput.*, vol. 21, no. 1, pp. 1-23, Feb. 1992.

[154] R. Sedgewick and P. Flajolet, An Introduction to the Analysis of Algorithms. Addison-Wesley Professional, 1995.

[155] X.C. Wang, X.L. Wang, and D.M. Wilkes, "A divide-and-conquer approach for minimum spanning tree-based clustering," *IEEE Trans. Knowl. Data En.*, vol. 21, no. 7, pp. 945-958, Jul. 2009.

[156] A.M. Geoffrion, "Generalized benders decomposition," *J. Optim. Theory Appl.*, vol. 10, no. 4, pp. 237-260, Oct. 1972.

[157] J.F. Benders, "Partitioning procedures for solving mixed-variables programming problems," *Numer. Math.*, vol. 4, pp. 238-252, 1962.

[158] CPLEX. IBM. http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/index.html. Accessed 25 Nov. 2013.

[159] MINOS. http://www.stanford.edu/group/SOL/ guides/minos55.pdf. Accessed 25 Nov. 2013.

[160] X. Han, J. Guo, and P. Wang, "Adequacy study of a wind farm considering terrain and wake effect," *IET Gen., Transm., Distrib.*, vol. 6, no. 10, pp. 1001–1008, Oct. 2012.

[161] Vestas, "Life cycle assessment of electricity production from a Vestas V112 turbine wind plant," http://www.vestas.com/en/about-vestas/sustainability/wind-turbines-and-the-environment/life-cycle-assessment-(lca).aspx. Accessed 25 Nov. 2013.

[162] Australian Bureau of Meteorology Website. http://www.bom.gov.au. Accessed 25 Nov. 2013.

[163] Mercfados. "AL-MOKHA 60 MW wind farm project-final report," http://www.shebacss.com/rd/docs/Al-Mokha_60_MW_Wind_Farm_Project.pdf,

http://www.shebacss.com/rd/docs/Final_ Report_v.1_Part_2.pdf. Accessed 25 Nov. 2013.

[164] AMPL. http://www.ampl.com. Accessed 25 Nov. 2013.

[165] A. Khaligh and Z.H. Li, "Battery, ultracapacitor, fuel cell, and hybrid energy storage systems for electric, hybrid electric, fuel Cell, and pug-In hybrid electric vehicles: state of the art," *IEEE Trans. Veh. Technol.*, vol. 59, no. 6, pp. 2806–2814, Jul. 2010.

[166] G. Sideratos and N.D. Hatziargyriou, "An advanced statistical method for wind power forecasting," *IEEE Trans. Power Syst.*, vol. 22, no. 1, pp. 258-265, Feb. 2007.

[167] K. Bhaskar and S.N. Singh "AWNN-assisted wind power forecasting using feed-forward neural network," *IEEE Trans. Sustainable Energy*, vol. 3, no. 2, pp. 306-315, Apr. 2012.

[168] The OptiSeries V1.0 User Manual, PolyU, Sep. 2009.

[169] Mukund R.Patel. Wind and Solar Power Systems. CRC Press LLC, 1999.

[170] "A guide to lead-acid batteries", www.solar2renewableenergy.com. Accessed 25 Nov. 2013.

[171] E. Gargari and C. Lucas, "Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition," *in Proc. IEEE Congr. Evolutionary Computation*, 2007.

[172] M. Hadji and B. Vahidi, "A solution to the unit commitment problem using imperialistic competition algorithm," *IEEE Trans. Power Syst.*, vol. 27, no. 1, pp. 117-124, Jun. 2011.

[173] B.Mohammadi-ivatloo, A. Rabiee, A. Soroudi, and M. Ehsan, "Imperialist competitive algorithm for solving non-convex dynamic economic power dispatch," *European Symposium on Computer-Aided Process Engineering*, 2011.

[174] E. Gargari and F. Hashemzadeh, "Colonial competitive algorithm: a novel approach for PID controller design in MIMO distillation column process," *International Journal of Intelligent Computing and Cybernetics*, vol.1, no. 3, pp. 337-355, 2008.

[175] M. Yousefi, A. Darus, and H. Mohammadi, "An imperialist competitive algorithm for optimal design of plate-fin heat exchangers," *Journal of Heat and Mass Transfer*, vol. 55, no. 11, pp. 3178-3185, May. 2012.