ADVANCED HEURISTIC OPTIMIZATION ALGORITHMS FOR OPTIMAL REACTIVE POWER PLANNING AND DISPATCH IN POWER SYSTEMS

MING NIU

PhD

The Hong Kong Polytechnic University

2020

The Hong Kong Polytechnic University

Department of Electrical Engineering

Advanced Heuristic Optimization Algorithms for Optimal Reactive Power Planning and Dispatch in Power Systems

MING NIU

A thesis

submitted in partial fulfillment of the requirements for

the degree of Doctor of Philosophy

Dec 2019

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

______________ (Signed)

MING NIU ____(Name of student)

Abstract

This thesis develops three advanced heuristic optimization algorithms (HOAs) for power system reactive power planning and dispatch.

Firstly, a comprehensive overview of the state-of-the-art HOAs applied for reactive power planning (RPP) and optimal reactive power dispatch (ORPD) is presented. It covers a number of HOA variants in the research field of RPP and ORPD problems, including genetic algorithm (GA), differential evolution (DE), particle swarm optimization (PSO), and evolutionary programming (EP), etc.

A modified quantum-inspired differential evolutional algorithm (MQDE) with a novel reset strategy is developed for optimal RPP. The proposed MQDE is based on quantum mechanics combining with a competitive DE mutation scheme, i.e. DE/best/1/bin. It overcomes a major difficulty of DE techniques in ensuring the search diversity of the population when the algorithm is approaching the region of local optimum in the later stages of iteration process.

A novel HOAs-adaptive range composite differential evolution (ARCoDE) algorithm is developed for ORPD that is one of the critical components in optimal power flow (OPF) study. Due to the nature of power dispatch, the ORPD problems need to be solved in a timely manner. This imposes a limitation on number of function evaluations. The proposed ARCoDE algorithm utilizes the concept of compositing different types of trial vector generation strategies, which makes possible a decent balance between the exploration and

exploitation capabilities in the solution. In addition, a novel control parameter range adaptation mechanism is proposed to enable a highly efficient adaptive tuning of control parameters. These novelties support ARCoDE to deliver satisfactory solutions while fulfilling the stringent time requirements.

Finally, an efficiency ranking-based evolutionary algorithm (EREA) is proposed aiming at directly obtaining the most efficient DMUs. A slacks-based measure (SBM) of efficiency and its super efficiency pattern are applied to yield a full ranking of relative efficiency of DMUs in each evolving generation, based on which the most efficient DMUs can be eventually found for the multi-objective formulation of ORPD problem.

Acknowledgements

To begin, I would like to express my sincere thanks and heartfelt gratitude to my supervisor ***Prof. Zhao Xu***, Department of Electrical Engineering, the Hong Kong Polytechnic University. It has been a great honor for me to undertake my PhD degree under his supervision. Prof. Xu not only possesses a deep understanding and strong knowledge over wide range of subjects, but also never hesitates to provide invaluable guidance and share novel ideas whenever I seek advice in taking research direction and pursuing my project. Without his teaching and continuous guidance this PhD would not have been achievable.

I am also very grateful to many research assistants that I met and worked with in Department Electrical Engineering, the Hong Kong Polytechnic University over the years, for both honest friendship and valuable counsel they gave.

My deep appreciations go out to ***Dr. Jia Hou***, ***Dr. Can Wan***, ***Dr. Songjian Chai***, ***Dr. Jiayong Li***, and ***Dr. Ningzhou Xu*** who all helped me in numerous ways along the way during this adventure.

I owe and respectfully say a heartfelt thank you to my ***mum***, ***Hui Cai*** and my ***dad, Zhengbin Niu*** for their moral inspiration and constant support in whatever way they could throughout this challenging period and my life.

And finally, to my wife, ***Dr. Qin Liu*** who has been by my side throughout this PhD. It was her understanding, dedicated support and incredible patience which made completion of this work possible.

Table of Contents

List of figures

List of tables

List of abbreviations

ABC:	Artificial bee colony
ACO:	Ant colony optimization
ACS:	Ant colony system
AGA:	Adaptive genetic algorithm
ANN:	Artificial neural networks
ARCoDE:	Adaptive range composite differential evolution
AS:	Ant system
ASrank:	Rank-based ant system
BBO:	Biogeography-based optimization
CCR	Chames, Cooper and Rhodes
CLPSO:	Comprehensive learning particle swarm optimization
CoDE:	Composite differential evolution
DE:	Differential evolution
DEA:	Data envelopment analysis
DEEPSO:	Differential evolution particle swarm optimization
DMUs:	Decision-making units
DQLF:	Decoupled quadratic load flow
EA:	Evolutionary algorithm
EAS:	Elitist ant system
EGA:	Enhanced genetic algorithm
EP:	Evolutionary programming
FACTS:	Flexible ac transmission systems
FVSI:	Fast voltage stability index

GA: Genetic algorithm

GSO: Group search optimizer

HOA: Heuristic optimization algorithm

HPSO: Hybrid particle swarm optimization algorithm

ICDE: Improved $(\mu+\lambda)$-constrained differential evolution

IGA: Improved genetic algorithm

IPM: Interior point method

IPSO: Improved particle swarm optimization

LP: linear programming

MADE: Multi-agent based differential evolution

MAPSO: Multi-agent systems

MIGA: Mixed-integer genetic algorithm

MILP: Mix integer linear programming

MINLP: Mixed integer nonlinear programming

MIPSO: Mixed-integer particle swarm optimization

MMAS: Max-min ant system

MOEA: Multi-objective evolutionary algorithm

MOPSO: Multi-objective particle swarm optimization

MQDE: Modified quantum-inspired differential evolutionary Algorithm

MVMO: Mean-variance mapping optimization

NLP: Non-linear programming

NN: Neural network

OPF: Optimal power flow

OPF-SC: Optimal power flow problem with embedded security constraints

ORPD:	Optimal reactive power dispatch
POF:	Pareto-optimal front
PSO:	Particle swarm optimization
QDE:	Quantum-inspired differential evolution algorithm
QEA:	Quantum-inspired evolutionary algorithm
QP:	Quadratic programming
QPSO:	Quantum-inspired particle swarm optimization
RDEA:	Robust differential evolution algorithm
RGA:	Refined genetic algorithm
RPD:	Reactive power dispatch
RPP:	Reactive power planning
SA:	Simulated annealing
SaDE:	Self-adaptive differential evolution
SB:	Security boundary
SBM:	Slacks-based measure
SARGA:	Self-adaptive real coded genetic algorithm
SPEA:	Strength Pareto evolutionary algorithm
SQP:	Sequential quadratic programming
STATCOM:	Static compensator
SVC:	Static VAR compensator
TCPS:	Thyristor-controlled phase shifter
TCSC:	Thyristor-controlled series capacitor
TS:	Tabu search
TSCOPF:	Transient stability constrained optimal power flow

UPFC:	Unified power flow controller
VAR:	Volt-ampere reactive
VSI:	Voltage stability index
WPP:	Wind power plant
FSM:	Finite state machine

Chapter 1 Introduction

1.1 Scope of Research

1.1.1 Reactive Power Planning in Modern Power System

Generally, reactive power planning (RPP), also known as VAR planning, can be defined as the process to allocate various reactive power sources considering their locations and sizes. By solving the RPP problem, the optimal amount and location of shunt reactive power compensation devices can be determined [1], [2]. The most common objective of RPP is to minimize cost associated with VAR sources' allocation and operation. Both the investment cost and operation cost need to be minimized simultaneously. Aside from operation constraints introduced by VAR sources, RPP problems also inherit the complexity of power systems, that is mathematically nonlinear and comes with a large number of variables and uncertain parameters. Accompanied by these complicated objectives and constraints, solving a RPP problem is regarded as one of the most challenging tasks in power systems planning and operation.

Conventionally, placing new reactive power sources is done by utility through either simple estimation or direct assumption. With growth of the system and deregulation of the power market, the pursuit of the best allocation of VAR devices starts to become economically driven by the system operators [3]–[5]. Historically, various optimization algorithms have been developed for RPP, such as nonlinear [6], linear [7], or mixed integer programming [8], and decomposition methods [9]–[12]. However, these techniques are known to only converge at local optima rather than the global ones. A typical RPP problem can have many local minima.

In recent years, heuristic optimization algorithms (HOAs) [13]–[20] have attracted attention for its effectiveness in solving optimization problems. HOAs are powerful optimization techniques that mimicking natural selection processes, such as in genetics. Theoretically, HOAs are capable of converging to the global optimum solution with 100% probability [2]. They are useful especially when traditional optimization algorithms fail to deliver optimal solutions. Although often being computational expensive, HOAs can benefit from recent advances in distributed computing techniques that could scale down execution times dramatically. With HOAs, doing a large amount of computation tasks in pursuing global optimum instead of local ones is achievable.

1.1.2 Optimal Reactive Power Dispatch in Modern Power System

On the other side, the dispatch of reactive power is also critical as it ensures the security and economy of power system operation. Optimal reactive power dispatch (ORPD) problem is a complicated mixed-integer non-linear optimization problem involving many constraints and discrete/continuous decision variables [21]. Without assumptions such as convexity, differentiability and continuity, traditional techniques including linear programming, non-linear programming, and interior point method may not handle these problems well [9], [22], [23]. In addition, the performance of these methods is highly affected by the initial solution guess. In view of the above issues, a variety of HOAs have been proposed to solve OPF and ORPD problems [24], including e.g. genetic algorithm (GA) [25], [26], evolutionary programming (EP) [27], particle swarm optimization (PSO) [28], [29], DE [30], seeker optimization [31], mean-variance mapping optimization (MVMO) [32], QEA [33], etc. In practice, fast ORPD computation is needed for e.g. power flow management [34], reactive source control of wind farm [35], and so on. Therefore, it is practically valuable to develop highly efficient HOAs to fulfill practical operation needs. In addition, the OPF and ORPD formulations are becoming even more complicated due to integration of renewable energies [5], [36]–[39], which also motivates applications of HOAs.

1.2 Incentives of Thesis

The major concerns of applying HOAs to RPP and ORPD problems include convergence speed and control parameters selection. The former can be enhanced by introducing exploitive recombination strategies, but the robustness of the algorithm may be compromised accordingly. That is the method should obtain satisfactory solutions in reasonable times while not too sensitive to changes in parameters. The latter, control parameters selection, can be handled by different adaptive or self-adaptive mechanisms to shorten the tedious trial-and-error procedure for fine tuning control parameters. However, these adaptive or self-adaptive strategies can still provide unsatisfactory parameters for practical applications, where only limited numbers of function evaluations are allowed due to the critical time requirement. For example, reactive power dispatch in power systems can be conducted in every 15 minutes, asking for a fast optimization solution. In addition, when dealing with multi-objective RPP/ORPD problems, a quality Pareto-optimal front (POF) is essentially a prerequisite for a post processing use e.g.to gauge the most efficient solutions. However, the search ability of most multi-objective HOAs severely deteriorates when more than three objectives are involved, resulting in a poor POF [39]

1.3 Aims and Objectives

The main objective set for this thesis is to apply advanced HOAs into power system RPP and ORPD problems. Three HOAs are developed for solving RPP and ORPD problems. Test results demonstrate that the proposed HOAs are capable of delivering satisfactory solutions while fulfilling critical requirements.

The specific aims of the thesis are:

1. Provide a comprehensive overview of advanced HOAs applied in RPP and ORPD problem. Wide variety of HOAs are covered, such as including GA, DE, PSO, and EP, etc. (Chapter 2);

2. Propose a modified quantum-inspired differential evolutional algorithm with a novel reset strategy for optimal reactive power planning. It overcomes a major difficulty of DE techniques in ensuring the search diversity of the population when the algorithm is approaching the region of local optimum in the later stages of iteration process. (Chapter 3);

3. Develop a novel HOAs-adaptive range composite differential evolution algorithm for ORPD. It utilizes the concept of compositing different types of trial vector generation strategies, which makes possible a decent balance between the exploration and exploitation capabilities in the solution. (Chapter 4);

4. Present an efficiency ranking based evolutionary algorithm for multi-objective ORPD problems. This evolutionary algorithm features a data envelopment analysis-based fitness assignment strategy to guide the iterations. With the proposed algorithm multi-objective ORPD can be solved efficiently. (Chapter 5).

1.4 Publications and Awards from the Thesis

Journal Publications:

[1] ***Ming Niu***, Zhao Xu, "Efficiency ranking-based evolutionary algorithm for power system planning and operation", *IEEE Transactions on Power Systems*, Jul. 2014.

[2] ***Ming Niu***, Can Wan, Zhao Xu, "A review on applications of heuristic optimization algorithms for optimal power flow in modern power systems", *Journal of Modern Power Systems and Clean Energy*, pp. 289-297, Dec. 2014.

[3] Can Wan, ***Ming Niu***, Yonghua Song, Zhao Xu, "Pareto optimal prediction intervals of electricity Price", *IEEE Transactions on Power Systems*, pp. 817-819, Jan. 2016.

[4] Ning Zhou Xu, Ka Wing Chan, Chi Yung Chung, ***Ming Niu***, "Enhancing adequacy of isolated systems with electric vehicle-based emergency strategy", *IEEE Transactions on Intelligent Transportation Systems*, 2019, *Early Access*.

Award:

[5] Top Five Best Algorithm Award, "Competition on Application of Modern Heuristic Optimization algorithms for Solving Optimal Power Flow Problems", IEEE PES Working Group on Modern Heuristic Optimization Intelligent Systems Subcommittee & Power System Analysis, Computing, and Economic Committee, Aug, 2014.

Conference Publications:

[6] Songjian Chai, ***Ming Niu***, Zhao Xu, Loi Lei Lai, Kit Po Wong, "Nonparametric conditional interval forecasts for PV power generation considering the temporal dependence", in *Proceedings of IEEE PES General Meeting*, pp.1-5, Jul. 2016.

Chapter 2 Literature Review

2.1 Fundamentals of Optimization Theory

Optimization principle is of vital importance in modern engineering design and system operations across various fields. In recent years, optimization research has been widely studied in many engineering fields including the electric power systems. In mathematics, an optimization problem is also referred to as a mathematic programming problem. It studies the way to seek best solutions from sets of available alternatives.

Mathematical representation of a generic optimization problem is given in the following way. First comes the objective function, i.e. the function needs to be minimized or maximized:

$$\textit{Minimize/maximize} \qquad f_0(x). \tag{2.1}$$

A solution x_0 is to be found in $\mathbb{R}^n$ such that, $\forall x \in \mathbb{R}^n$

$$\begin{aligned} f(x_0) \leq f(x) \quad & \text{for minimization, or} \\ f(x_0) \geq f(x) \quad & \text{for maximization} \end{aligned} \tag{2.2}$$

where $f_0: \mathbb{R}^n \to \mathbb{R}$ is the objective function or cost function; $x = [x_1, x_2, \cdots, x_n]$ is the optimization or control variables of the optimization problem, because we need to select their values to achieve the optima of the objective function; and $\mathbb{R}^n$ to $\mathbb{R}$ stands for real number set of n and single dimensionality. If the inequality conditions in (2.2) only holds within a small neighborhood of the solution x_0, the objective f is said to have a local minima or maxima, and x_0 is the local optimal solution to the problem (2.1); otherwise, f has the global minima or maxima, and x_0 is the global optimal solution to the problem (2.1).

Note, often the optimization problem is accompanied with constraints in the forms of equality or inequality functions that can limit the actual solution space of x. A complete of representation of the constrained optimization problem is therefore given by,

$$\begin{array}{cc} \textit{Minimize/maximize} & f_0(x), x \in \mathbb{R}^n \\ \textit{subject to} & f_i(x) \le b_i,\ \ i = 1, 2, \cdots, p \ , \\ & g_j(x) = 0,\ \ j = 1, 2, \cdots, q \end{array} \quad (2.3)$$

where $h_i: \mathbb{R}^n \to \mathbb{R}$, $i = 1, 2, \cdots, p$ are the inequality constraint functions, and $b_1, b_2, \cdots, b_p$ are the boundaries of the constraint functions; $g_j: \mathbb{R}^n \to \mathbb{R}$, $j = 1, 2, \cdots, p$ are the inequality constraint functions.

The optimization research comprises of a wide range of different sub-research fields. Basic categorization of the subfields depends on the properties of the optimization objective function, control variables, constraints, problem formulation, etc.

Linear programming (*LP*) deals with optimization problems with linear objective functions and constraints. For example, the formulation in (2.3) is a linear programming problem, if the objective function (and the constraints must) fulfills the following relation, $\forall x, y \in \mathbb{R}^n, \alpha, \beta \in \mathbb{R}$

$$f_0(\alpha x + \beta y) = \alpha f_0(x) + \beta f_0(y). \quad (2.4)$$

Otherwise, a problem is a *non-linear programming* (*NLP*) problem.

Quadratic programming (*QP*) solves optimization problems with the objective function of quadratic forms. An *integer programming* problem is defined as optimization problem in which control variables are restricted to be integers; while in *mixed integer programming* only some control variables are integers and the others are of non-integer type.

Due to the inclusion of constraints, *constrained optimization* usually involves higher complexities in solving the problem than *unconstrained* ones. *Deterministic programming* is in opposition to *stochastic programming* where the problem formulation is of stochastic nature with random numbers embedded in the objective and constraints. *Combinatory optimization* is a method solving problems in which the set of feasible solutions is discrete or can be reduced to a discrete one. *Dynamic programming* tackles decision-making that spans several points in time. The optimization strategy is to break down the problem into smaller sub problems that are correlated by constraint equations.

Furthermore, combinations of 'basic' categories lead to more 'advanced' classification, such as *mix integer linear programming* (*MILP*) and *mixed integer nonlinear programming* (*MINLP*). Table 2.1 summarizes major subfields of mathematical optimization. Readers are referred to literatures [40], [41] for further details in optimization theory.

Table 2.1. Major subfields of mathematical optimization.

	Objective function	*Constraints*	*Control (decision) variables*	*Comments*
Linear programming	Linear	Linear	Real	
Nonlinear programming	Nonlinear/linear	Nonlinear/linear	Real	Some of objectives or constraints nonlinear
Quadratic programming	Quadratic	Linear	Real	
Integer programming	Linear	Linear	Integer /binary	
Mixed Integer Programming	Linear	Linear	Integer /binary	
Mix Integer Linear Programming	Linear	Linear	Real/integer /binary	Some of objectives or constraints nonlinear
Mix Integer Nonlinear Programming	Nonlinear/linear	Nonlinear/linear	Real/integer /binary	
Constrained optimization	-	Yes	-	
Unconstrained optimization	-	No	-	
Deterministic Programming	Deterministic formulation	Deterministic formulation	-	
Stochastic programming	with uncertainties	with uncertainties	-	
Combinatory Optimization	-	-	-	Discreet feasible solution space
Dynamic Programming	-	-	-	Decision making over time

2.2 Multi-objective Optimization

More than often, real-world problems can involve more than one objective function to be optimized simultaneously. This multi-criterion decision-making process is termed as *multi-objective optimization.* The generic formulation of a multi-objective minimization problem is given below,

$$\begin{array}{ll} \textit{Minimize} & F(x) = [f_1(x), f_2(x), \cdots, f_N(x)]^\top, \;\; x \in \mathbb{R}^n \\ \textit{subject to} & h_i(x) \le b_i, \;\; i = 1, 2, \cdots, p \\ & g_j(x) = 0, \;\; j = 1, 2, \cdots, q \end{array} , \tag{2.5}$$

where, if $N = 1$ the multi-objective problem is then reduced to a single objective problem. For single objective optimization, a global optimal solution always exists, though sometimes difficult to find. However, this is generally not the case for multi-objective problems. It is common that optimal decisions need to be taken in the presence of trade-offs between two or more conflicting objectives, for instance, minimizing cost whilst maximizing quality when buying goods and services. Rather there can be a set of candidate solutions that are *Pareto optimal,* meaning that these solutions cannot be further improved for one objective without sacrificing one or more other objectives. The concept of Pareto optimality is first introduced by Francis Ysidro. Later Vilfredo Pareto generalized it [42]. The set of the Pareto optimal solutions is called Pareto set. The corresponding objective vectors formulate a *Pareto front* in the feasible objective space [43]. Figure 2.1 illustrates the concept of Pareto front for a two-objective minimization problem.

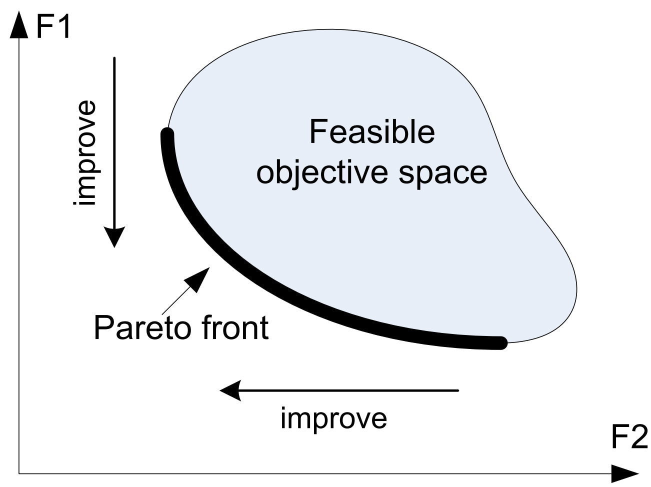

Figure 2.1. Illustration of Pareto front for a two-objective minimization problem.

Pareto optimal solutions represent different trade-offs between different objectives, reflecting the decision makers' preferences or attitudes towards different objectives in consideration. The approach of solving the multi-objective problem is usually through aggregating all involved objectives into one single formulation by using methods such as,

- *Sum weighted method*: This method simply assigns different weights and then sums up all objectives into one expression. The resultant 'single' objective is designed by the decision maker to reflect its preference. A popular formulation is a linear combination of all objectives, as in (2.6),

$$\textit{Minimize} \quad M(x) = \sum_{k=1}^{N} w_k f_k(x)\,, \text{with } w_k \geq 0 \text{ and } \sum_{k=1}^{N} w_k = 1 \qquad (2.6)$$

where the weights (w_k) represent degree of importance for each objective, specified by the decision maker a priori. Here constraints in (2.5) also apply.

- *Goal programming method*: It minimizes deviation from pre-set goals. Typical formulation is given by (2.7)

$$\text{Minimize} \quad M(x) = \sum_{k=1}^{N} w_k |f_k(x) - T_k|, \text{ with } w_k \geq 0 \text{ and } \sum_{k=1}^{N} w_k = 1 \qquad (2.7)$$

where w_k are weights assigned and T_k is objective k's goal, both specified by the decision maker a priori. Constraints in (2.5) also apply. This method needs a priori information of priorities and goals in the original optimization problem. To find out the goals, separated optimization procedures for individual objectives need to be performed. Alternatively, estimation of these objectives is acceptable.

- *Epsilon constrained method*: This method seeks Pareto optimal solutions via optimization of one objective, subject to the rest objectives as constraints bounded by allowable range ε_k. The entire Pareto set can be generated by repetitively solving the problem for different values of ε_k. The formulation of the epsilon constrained method usually takes,

$$\begin{array}{ll} \text{Minimize} & f_i(x) \\ \text{Subject to} & f_k(x) \leq \varepsilon_k, k = 1, 2, \cdots, N; i \neq k \end{array} \qquad (2.8)$$

and other constraints in (2.5) also apply. The Pareto set is achievable by repeatedly solving above equations for different epsilons. This method is relatively simple, but computationally intensive.

2.3 Modern Heuristic Optimization Algorithms

The modern HOAs represent a group of intelligent algorithms that either make analog of the natural evolution process based on Darwinian principles or mimic a certain natural phenomenon in searching for an optimal solution. They have been successfully applied to a wide range of power system optimization problems where non-differentiable regions exist, and the global solution are extremely difficult to be gauged. The most popularly used HOAs in solving RPP and ORPD problem are compactly introduced as the follows:

2.3.1 Genetic Algorithm

GA is one of the most popular and famous approaches in evolutionary computation. Founded on the mechanism of natural genetics and Darwinian principles of evolution and natural selection, this novel algorithm is developed by J.H. Holland in 1965, and showed strong capabilities and advantages for solving a wide range of problems introduced in his pioneering book [44]. GA can be considered as a population-based approach, the search process of which is conducted by means of transforming a set of points (individuals) to another set of points in the search space. In original GA, each individual is represented via a fixed-length binary string. This method maps the points in the search space into the instances of artificial chromosome. Desired precision can be simply approximated through tuning the length of binary string. The strong preference to the binary representation of GA probably derives from Schema Theorem [13] which tries to investigate the mathematical foundation of GA.

2.3.2 Particle Swarm Optimization

PSO, which is introduced by Kennedy and Eberhart in 1995 [45], [46], is one of the most important swarm intelligence paradigms. PSO uses a simple mechanism that mimics swarm behavior in birds flocking and fish schooling to guide the particles to search for globally optimal solution. As PSO is easy to implement, it has rapidly progressed in recent years and with many successful applications in solving real-world optimization problems.

2.3.3 Differential Evolution

The Differential evolution (DE) approach is firstly proposed in a technical report by Storn and Price [47]. It is a population-based method and is generally considered a parallel stochastic direct search optimizer that is simple yet powerful. DE is a stochastic population-based optimization algorithm with real parameters and real-valued functions. The core idea behind DE is a scheme for generating trial parameter vectors. DE generates new parameter vectors by weighing the difference vector between two population members and then adding that to a third member. If the resulting vector yields a lower objective function value than a previously determined population member, the newly generated vector replaces the vector to which it was compared. In comparisons to most other HOAs, DE algorithm is much simpler and more straightforward to implement. The main body of the algorithm takes four or five lines of code in any programming language. Despite its simplicity, the gross performance of DE in terms of accuracy, convergence rate and robustness makes it attractive for applications to various real-world optimization problems [48]–[50], where finding an approximate solution in a reasonable amount of computational time is of considerable importance. The spatial complexity of DE is lower than that of some highly competitive real parameter optimizers. This feature helps in extending DE to handle expensive and large-scale optimization problems.

2.3.4 Evolutionary Programming

EP is first introduced by L. J. Fogel and Burgin in the research of artificial intelligence [51]. In order to achieve intelligent behavior, the authors come up with the idea of defining the environment as a sequence of symbols (in a finite alphabet) and evolving an algorithm to predict the next symbol to appear based on the former observed sequence of symbols. Finite state machine (FSM) is chosen to be the form of individuals, as it provides a meaningful representation for the required behaviors in the environment. While the original form of EP was applied in discrete problems due to the FSM representation, Back, Fogel, and Michalewicz extend EP into the real-valued continuous optimization problem [52]. Both the mutation mode and the number of mutations per offspring FSM are with respect to a probability distribution, which means some individual may mutates more than once in one generation.

2.4 A Review of HOAs for Power System Reactive Power Planning and Dispatch

2.4.1 GA-based Approaches

An improved genetic algorithm (IGA) with the dynamical hierarchy of the coding system is developed to solve the ORPD problem [53]. IGA demonstrates ability to code a large number of control variables in a practical system. It is tested on IEEE 30-bus system with both normal and contingent operation states. ORPD problem for a multi-node auction market is studied by means of GA in [54] to maximize the total participants' benefit at all nodes in the power system. In [26], a self-adaptive real coded genetic algorithm (SARGA) is developed to solve OPF problem, where the self-adaptation in real coded genetic algorithm is reached through simulated binary crossover operator. A novel evolutionary algorithm (EA) is developed combing a new decoupled quadratic load flow (DQLF) solution with enhanced genetic algorithm (EGA) to solve the multi-objective OPF problem [55]. A strength Pareto evolutionary algorithm (SPEA)-based approach is employed to obtain the Pareto-optimal set. The proposed multi-objective evolutionary algorithm demonstrates superiority in comparisons to PSO–Fuzzy approach. An adaptive genetic algorithm (AGA) is developed to solve ORPD problems and voltage control [56], where the probabilities of crossover and mutation are adjusted in terms of the fitness values of the solutions and the normalized fitness distances between the solutions in the evolution process. In [57], a refined genetic algorithm (RGA) is developed for solving OPF problem. This GA can code a large number of control variables and has less sensitivity to starting points. GA is also used to deal with power system security enhancement based ORPD in [58] considering the actions to possible overloads in the network due to contingencies. An EGA with advanced and problem-specific operators is introduced for solving OPF with both continuous and discrete control variables [25]. An efficient real-coded mixed-integer genetic algorithm (MIGA) is presented in [59] to solve non-convex ORPD problems with security constraints. According to the numerical studies on IEEE 26-bus and 57-bus systems, MIGA performs better than EP. A novel hybrid method integrating a GA with a nonlinear interior point method (IPM) is proposed for OPF problem [60]. In this hybrid approach, GA is responsible for solving the discrete optimization with the continuous variables, and the IPM is responsible for solving the continuous optimization with the discrete variables. Numerical simulations are implemented on IEEE 30-bus, 118-bus and realistic Chongqing 161-bus test systems, respectively. Wankhade and Vaidya discuss the effects of various combination of control variables on the convergence of simple genetic algorithm [61]. Statistical parameter-based study is conducted to visualize the effects of the selection of control variables on ORPD convergence in terms of the computation time and the accuracy improvement. The experiment results prove that the set of control variables with the voltage of slack bus, the active/reactive power outputs of generators, and the reactive power outputs of controllable buses can be the most effective in obtaining the global solution under normal and contingent conditions. In [62], it is claimed that the main disadvantages of GAs is the high CPU execution time and the qualities of the solution deteriorate with practical large-scale OPF problems. An efficient parallel GA is developed for the solution of large-scale OPF problem with the consideration of practical generators constraints. The length of the original chromosome is reduced on basis of the decomposition level and adapted with the topology of the new partition. Partial decomposed active power demand is added as a new variable and searched within the active power generation variables of the new decomposed chromosome. The strategy of the ORPD problem is decomposed into two sub-problems, of which the first sub-problem is related to active power planning to minimize the fuel cost function and the second sub-problem is designed to make corrections to the voltage deviation and reactive power violation in an efficient reactive power planning of multi Static VAR Compensator (SVC). Numerical results on three test systems—IEEE 30-bus, 118-bus and 15 generation units—with prohibited zones are presented and compared extensively with results obtained using stochastic search algorithms, enhanced GA, ant colony optimization (ACO), and GA-fuzzy system approaches. GA is successfully used for optimal reactive power planning in [20] to search for a global optimal solution. It has been verified on practical 51-bus and 224-bus systems to indicate its feasibility and capability. GA based optimization technique is applied for the proper allocation of VAR sources and model analysis method shows better solution than L-index approach of detection not only in the aspect of RPP (active power loss) but also in terms of voltage stability [63]. In [64], a hybrid approach of GA, SA (simulated annealing) and TS (tabu search), which improve the GA by adopting the acceptance probability of SA avoiding being trapped by a local optimal solution, has been tested (37-bus practical area power system with 79 control variables) to show the effectiveness of the planning method in Shandong province of China.

2.4.2 PSO-based Techniques

As an efficient and reliable evolutionary-based approach, PSO algorithm is applied for optimal settings of ORPD problem control variables [65]. A novel particle swarm optimization approach based on multi-agent systems (MAPSO) is presented [29] to solve OPF problems. Each agent, representing a particle to PSO, in MAPSO competes and cooperates with its neighbors. Experiment results prove that the proposed MAPSO approach can reach better solutions much faster than the mature approaches. A multi-objective PSO technique is developed to deal with the highly nonlinear and non-convex multi-objective OPF problems [66]. In addition to conventional objective generation cost, another conflicting objective environmental pollution is formulated and minimized simultaneously. A fuzzy based hybrid PSO approach for solving ORPD problem considering the forecasting uncertainties of wind speed and load demand in power systems is proposed in [67]. A comprehensive learning PSO (CLPSO) is developed for reactive power dispatch to reduce grid congestions [68]. A fuzzy decision-based mechanism is employed to determine the best compromise solution from the derived Pareto set. A new multi-objective PSO (MOPSO) technique for solving OPF problem is proposed in [69]. The MOPSO methodology is formulated via the redefinition of global best and local best individuals in multi-objective optimization domain. Reference [28] presents a hybrid particle swarm optimization algorithm (HPSO) to solve the discrete ORPD problem. Newton-Raphson algorithm for the minimization of the mismatch of the power flow equations is integrated to the proposed HPSO algorithm. PSO technique is applied for a transient-stability constrained OPF (TSCOPF) problem modeled as an extended OPF with additional rotor angle inequality constraints [70]. Onate Yumbla et al. proposed a PSO algorithm with reconstruction operators to solve the ORPD problem with embedded security constraints (OPF-SC), represented by a mixture of continuous and discrete control variables [71]. The major objective is to minimize the total operating cost, taking into account both operating security constraints and system capacity requirements. The reconstruction operators guarantee searching the optimal solution within the feasible space, reducing the computation time and improving the quality of the solution. An improved PSO algorithm is developed in [72] for multi-objective OPF problem. The improved PSO that profits from chaos queues and self-adaptive concepts is used to improve the quality of the solution, particularly to avoid being trapped in local optima. In addition, a new mutation strategy combining different mutant rules is proposed to increase the search ability of the proposed algorithm. The proposed multi-objective OPF considers the fuel cost, loss, voltage stability and emission impacts involved in the objective functions. A fuzzy decision-based mechanism is used to select the best compromise solution of Pareto set obtained by the proposed PSO. In [73], PSO and group search optimizer (GSO) are used to solve the OPF problem with distributed generator failures in power networks. An OPF problem considering controllable and uncontrollable distributed generators is formulated, and cases with single and multiple generator failures are addressed. PSO has been proved powerful solution and outperform the other heuristic tools and successful application in RPP is reported [64], [74]. Modified PSO with dynamic inertia parameter with optimal off-nominal tap ratios of Online Tap Changers (OLTC), reactive power generations of alternators, and Shunt Capacitor (ShC) admittances has been applied for RPP and results in improved and lower transmission losses as well as reduction of annual operational cost [75].

2.4.3 DE-based Techniques

A multi-agent based differential evolution (MADE) based on multi-agent systems is developed for dealing with OPF problem with non-smooth and non-convex generator fuel cost curves in [76]. A novel robust differential evolution algorithm (RDEA) with new recombination operator is introduced to solve multi-objective OPF problem including two objective functions of generation cost and voltage stability margin [77]. Similarly, DE is used to solve ORPD problem with multiple and competing objectives [30]. The OPF problem is divided into two sub-problems, i.e., active power dispatch and reactive power dispatch are considered. A DE-based approach to solve OPF is developed by Abou El Ela, Abido, and Spea [65]. In their formulation, different objective functions that reflect fuel cost minimization, voltage profile improvement, and voltage stability enhancement are examined are examined. Non-smooth pricewise quadratic cost function is also been considered. Sayah and Zehar propose a similar formulation of OPF with non-smooth and non-convex generator fuel cost curves [78]. They employ a modified DE with a more exploitative mutation strategy and a random mutant factor. For testing purpose, the authors adopt a six-bus and the IEEE 30 bus test systems with three different types of generator cost curves. Comparisons are made among EP, PSO, typical DE. Results are in favor of the proposed modified DE. In [79], DE is comprehensively studied in terms of concept, mechanism, and parameter setting for solving OPF problems. The effectiveness of parallel computing technology for speeding up the computation is also analyzed. It has been concluded that DE requires relatively large populations to avoid premature convergence for medium-size test systems. In order to overcome this disadvantage, a decomposition and coordination method is proposed by the same authors based on the cooperative co-evolutionary architecture and the voltage-var sensitivity-based power system decomposition technique and incorporated with DE in [80]. The framework is implemented with a three-level parallel computing topology. Basu has used DE to minimize the generator fuel cost in OPF with flexible ac transmission systems (FACTS) devices including thyristor-controlled series capacitor (TCSC) and thyristor-controlled phase shifter (TCPS) [81]. Comparisons among DE, EP, and GA are conducted, indicating that the DE approach can obtain better solution and less computational complexities. Considering transient stability constraints into OPF, Cai *et al.* used DE to find the optimal setting for power system operation [82]. To deal with the large-scale system and speed up the computation, DE is parallelized and implemented on a Beowulf PC-cluster. Sivasubramani and Swarup propose a hybrid algorithm combining sequential quadratic programming (SQP) and DE for solving ORPD [83]. In this hybrid method, SQP is used to generate an individual, which is a member of an initial population, for DE algorithm. This manipulation makes DE more effectively to reach the optimal solution. In [84], an application of Fast Voltage Stability Index (FVSI) to RPP is proposed using Artificial Intelligence Technique based DE tool has been applied in the IEEE 30-bus system and results show striking lower system losses and improvement of voltage stability. A new variant of the DE algorithm (DE/best/1) is employed with the Pareto concept the RPP and the simulation results show the potential and outperformance to solve multi-objective RPP formulation compared with other approaches (GA, PSO) [84].

2.4.4 EP Techniques

In [27] and [85], an efficient and reliable EP algorithm is developed to solve OPF problem using the gradient information. The proposed algorithm has been successfully tested on IEEE 30-bus system with different highly non-linear curves of generator performance. An improved EP and its hybrid version combined with the nonlinear interior point technique are proposed in [86] to solve ORPD problems, indicating the superiority of computational efficiency and optimality. The common practices in regulating reactive power are integrated in modifying the mutation direction of control variables of EP to improve its speed. The interior point method is applied to reach a fast-initial solution which assisted the initial population of the improved EP method. An improved EP with multiple subpopulations and parallel search for solving ORPD with non-smooth and non-convex generator fuel cost curves is proposed in [87]. Gaussian and Cauchy mutation operators have been included in different subpopulations to improve the search diversity and avoid the local optimum. In [88], EP is applied for solving security constrained optimal power flow (SCOPF) problem, where contingency-case security constraints are involved in the optimization of the defined objective function. EP based ORPD in deregulated electric market environment is used and validated in [89]. EP algorithm is proposed to solve the OPF problem of generator units with ramp rate limits and non-smooth fuel functions such as quadratic, piece-wise, valve point loading and combined cycle cogeneration plants [90]. In [91], an application of fast voltage stability index (FVSI) to RPP using EP is proposes and has been used in the IEEE 30-bus system. Results observe improvement of voltage stability, considerable reduction in system losses and more savings on real power with the use of FVSI for the RPP problem as well as avoiding voltage collapse in the system. The effectiveness of EP and differential evolution to solve RPP problem incorporating FACTS Controllers like SVC is compared in [92]. Results show that the losses are reduced when using unified power flow controller (UPFC) than using TCSC and SVC. By the DE method with FVSI approach, more savings on the installment costs and energy are achieved. Results demonstrate that annual cost saving is increased using UPFC than TCSC and SVC devices.

2.4.5 Other Techniques

Artificial neural networks (ANNs) have been employed to model stability and security constraints in ORPD to formulate the system security boundary (SB) [93]. The key novelties of the proposed algorithm include that a neural network (NN) is trained to derive the SB model and a differentiable mapping function obtained from the NN is used as a constraint in the formulation of ORPD problem. This approach ensures that the operating points resulting from the ORPD solution process are within a feasible and secure region, comparing with typical security constrained ORPD models. A new ANN memory model–based algorithm is proposed to online implement the unified ORPD [94]. The ANN memory model is used to store the load patterns and their related optimal schedules. The proposed algorithm maximizes voltage stability margin while minimizing two other objectives generation cost and transmission loss. The computation efficiency is dramatically improved comparing with typical approaches.

Different ACO algorithms are proposed to handle optimal reactive power dispatch problem in [95], including ant system (AS), elitist ant system (EAS), rank-based ant system (ASrank) and max-min ant system (MMAS). The problem is modeled as a combinatorial optimization problem involving nonlinear objective function with multiple local minima. The proposed ACO algorithms have been compared to conventional mathematical methods, genetic algorithm, evolutionary programming, and particle swarm optimization to demonstrate the effectiveness and efficiency. This paper presents the ant colony system (ACS) method for constrained load flow problem formulated as a network-constrained optimization problem [96]. The proposed ACS is a distributed algorithm consisting of a set of cooperating ants to collaboratively search an optimum solution of the constrained load flow problem. In addition, the ACS algorithm is also applied to the reactive power control problem with network operating constraints to minimize real power losses.

Simulated annealing (SA) technique is proposed for solving OPF in [97]. It has been demonstrated that SA is able to solve the OPF problem as well as the load flow and the economic dispatch problem simultaneously. In Reference [98], a novel HOA algorithm, called biogeography-based optimization (BBO) is employed to solve constrained OPF problems in power systems with the consideration of valve point nonlinearities of generators. The simulation results of the proposed approach have been compared with EP, GA, PSO, mixed-integer particle swarm optimization (MIPSO) and sequential quadratic programming (SQP) to indicate its effectiveness for the global optimization of multi-constraint OPF problems. A QEA-based on quantum computation is developed for bid-based active and reactive OPF problems [33]. In [99], an artificial bee colony (ABC) algorithm based on the intelligent foraging behavior of honeybee swarm is proposed for ORPF problem.

2.4.6 Hybrid Techniques

A hybrid tabu search and simulated annealing (TS/SA) approach is proposed in [100] to deal with ORPD control with FACTS devices including two types TCSC and TCPS. Test results on IEEE 30-bus system demonstrate that the proposed hybrid TS/SA approach can perform better than GA, SA, or TS alone. A hybrid approach integrating fuzzy systems with GA and PSO algorithm is proposed for the application for OPF problem [101]. A hybrid algorithm of DE and EP (DEEP) is proposed for solving ORPF problem [102]. The proposed DEEP algorithm reduces the required population size by using the advantages of DE and EP. In order to overcome the limits of DE and ABC, a hybrid DE and ABC technique (DE-ABC) is developed for solving the ORPD problem [103]. Numerical tests indicate the robustness of the DE-ABC approach. A hybrid evolving ant direction differential evolution (EADDE) algorithm is developed to deal with the OPF problem with non-smooth and non-convex generator fuel cost characteristics in [104].

2.5 Discussion

HOAs are typically very versatile with respect to RPP/ORPD problem format. In addition, most HOAs are able to escape local optima which is critical for solving optimization problems.

However, all the HOAs discussed tend to be computationally intensive, yielding impractically long execution times for RPP/ORPD problems involving large scale systems. To overcome this, parallel processing was executed by most of the reviewed population based HOAs therefore the computational time can be significantly reduced.

Furthermore, the reviewed HOAs possess several parameters which must be tuned to ensure good performance. Consideration of the tuning of pre-determined parameters of HOAs will make these algorithms less robust. Hence, one of the most challenging aspects for HOAs lies in how to consistently converge to a feasible solution that provides an acceptable objective value within limited function evaluations.

In the reviewed literatures, on the premise of a proper pre-defined parameter choices, almost every HOA method is claimed as being more robust or can converge to a better solution compared with other HOAs. However, comparisons between different HOAs are difficult, as the selection of pre-defined parameters for each HOAs dominates the results. Moreover, according to the no-free-lunch (NFL) theorem, there cannot exist any algorithm for solving all problems that is generally superior to its peers. Therefore, the rest of this thesis focuses on HOAs design with respect to solving a specific aspect or formulation of RPP/ORPD problem.

Chapter 3 A Modified Quantum-inspired Differential Evolutional Algorithm for Optimal Reactive Power Planning

DE algorithm emerged as a competitive form of evolutionary computing more than a decade ago. The first written account of DE appears in a technical report by Storn and Price [47]. It is a population-based method and is generally considered a parallel stochastic direct search optimizer that is simple yet powerful. In DE research community, individual trial solutions (which constitute a population) are called parameter vectors. DE algorithm operates through the same computational steps as a standard evolutionary algorithm. However, unlike traditional EAs, DE algorithm employs difference in the parameter vectors to explore the objective function. Here, we point out some of the reasons that researchers have been attracted to DE as an optimization tool:

1. Compared with most other EAs, DE algorithm is much simpler and more straightforward to implement. The main body of the algorithm takes four or five lines of code in any programming language.

2. Despite its simplicity, the gross performance of DE in terms of accuracy, convergence rate, and robustness makes it attractive for applications to various real-world optimization problems [48]–[50], where finding an approximate solution in a reasonable amount of computational time is of considerable importance.

3. Number of control parameters in DE are limited (C_r, F, and NP in classical DE). The effects of these parameters on algorithm performance have been well studied [105]–[107].

4. The spatial complexity of DE is lower than that of some highly competitive real parameter optimizers. This feature helps in extending DE to handle expensive and large-scale optimization problems.

Although DE has been used to solve various optimization problems, it has a major difficulty in ensuring the search diversity of the population when the algorithm is approaching the region of local and global optimum [108]. Search efficiency will be impaired by the inefficient mutation operator during a searching process with rapidly descending population diversity.

Another research field called quantum computing has appeared and has generated intensive research over the last decade. Quantum computing is a novel model that utilizes the quantum states and quantum entanglement from quantum mechanics to achieve more effective computing than classical computer technology affords [109]. Narayanan [110] proposed the quantum-inspired genetic algorithm, which utilizes the interference as a crossover operator. Han and Kim [111] proposed a quantum-inspired evolutionary (QEA) algorithm based on the concepts of qubits and state superposition. QEA adapts qubit chromosomes to represent linear solutions and uses a quantum gate as a variation operator to update the qubit chromosomes. By investigating the various QEAs existing at that moment, qubit individuals are updated by applying a rotation gate. However, the magnitude and direction of rotation angle of quantum rotation gate must be determined in optimization process. At present, direction of the rotation angle is usually determined by a query lookup table that inefficiently deals with numerous conditional judgments.

For the above problem, Meng *et al*. [112] propose a quantum-inspired particle swarm optimization (QPSO) with a novel updating mechanism based on the idea of classical PSO, which involves the exchange of information among the velocity, global-best, local-best, and current particles. Moreover, QPSO is extended with a combination of self-adaptive probability selection and chaotic sequences mutation to solve an economic load dispatch problem after undergoing some standard benchmark function tests. Recently, a quantum-inspired DE algorithm (QDE) with a novel updating strategy for qubit individuals is proposed on the basis of the DE algorithm [113]. The mutation operator and crossover operator of DE are implemented in order to generate new qubit representations. This strategy successfully avoids the problems involving the rotation angle and improves the diversity of the qubit individuals. QDEs have been successfully implemented to solve permutative scheduling problems [114], N queens problem [115], and 0–1 knapsack problem [116] as well as for classification rule discovery [117]. However, previous work neither explored the ability of QDEs to solve functional optimization in a continuous real-valued search space nor emphasized the advantage of applying qubit representation to prevent a deficiency of population diversity in the later stages of the DE iteration process. More importantly, the qubit individuals cannot obtain efficient guidance about the updating direction from traditional parent-child selection operator of DE. It is because the binary encoding process realized by measuring state of qubits on a chromosome is a probabilistic operation with great randomicity.

To surmount all of the above problems, this chapter proposes a newly modified QDE (MQDE) algorithm. The improvement is mainly achieved through the concepts of quantum computing (such as a qubit and a random superposition of quantum states) and through a novel update/reset strategy that incorporates the DE mutation scheme DE/best/1/bin.

3.1 Basics of DE Algorithm

The DE algorithm is an evolutionary theory based multi-objective optimization algorithm to find the global optimum solution of multidimensional space. The key of DE is an approach to generate trial solution vectors by weighing the difference vector between two population members and adding it to a third population member. If the generated vector produces a smaller objective value than its target vector, it will be prioritized in the evolutionary optimization process. The DE algorithm consists of four steps: *initialization*, *mutation*, *crossover*, and *selection*.

Initialization: DE starts with a stochastic population composed of *NP* *D*-dimensional real-valued vectors. X_i represents the ith population vector of generation *G*, shown as

$$X_i = [x_{i,1}, x_{i,2}, \cdots, x_{i,D}] \tag{3.1}$$

Since each variable has a certain range, the jth element of the ith vector can be generated as

$$x_{i,j} = x_j^{\min} + rand_{i,j}[0,1] \cdot (x_j^{\max} - x_j^{\min}) \tag{3.2}$$

Mutation: The mutation of DE maintains the population diversity and provides necessary information to steer the optimization. One of the simplest DE mutation operators generates a mutated vector for every target vector X_i of generation *G,* given as

$$V_i = X_{r_1} + F \cdot (X_{r_2} - X_{r_3}), \tag{3.3}$$

where r_1, r_2, and r_3 which differ from i are integers randomly selected from $[1, NP]$, F is the mutation constant controlling the amplification of the differential variation.

Crossover: To enhance population diversity, the mutated vector is mixed with a predetermined target vector to form the trial vector, which is also known as crossover. Specifically, the trial vector can be expressed as

$$U_i = [u_{i,1}, u_{i,2}, \cdots, u_{i,D}], \tag{3.4}$$

where,

$$u_{i,j} = \begin{cases} v_{i,j} & \text{if } rand_{i,j}[0,1] \le C_R \text{ or } j = j_{rand} \\ x_{i,j} & \text{otherwise} \end{cases} \tag{3.5}$$

and j_{rand} is an integer randomly selected from $[1, 2, \cdots, D]$ to ensure that $u_{i,j}$ obtains at least one element from $v_{i,j}$. In addition, the crossover constant C_R

controls the population diversity.

Selection: This step is adopted to determine the survival of the target or trial vector:

$$X_i^{G+1} = \begin{cases} U_i^G & \text{if } f(U_i^G) \le f(X_i^G) \\ X_i^G & \text{otherwise} \end{cases}, \tag{3.6}$$

where $f(.)$ is the objective to be minimized. Equation (3.6) ensures the fitness of the population does not deteriorate.

Based on the fact that the performance of DE depends on the acquisition strategies of trial vector and control parameter settings, many DE variants are developed by academia. With respect to the acquisition strategies of trial vector, a trigonometric mutation operator is proposed by Fan and Lampinen [118] to improve the convergence performance of DE. As the trial vector moves to the best of three individuals chosen for mutation, the local search is enhanced by their mutation operator. Mezura-Montes *et al.* [119] proposes a novel mutation operator named "current-to-best/1", which generate a novel trial vector by integrating the optimal solution in the current population to that of the current parent. Feoktistov and Janaqi [120] classify mutation operators into four types based on the method they utilize the values of objective function. Obviously, the performance of "current-to-best/1" strategy with respect to explore the search space is poor when applied to solve the multimodal problems. To meet this challenge, many efforts have been implemented to ameliorate the application of this strategy. With the aid of establishing a local neighborhood model, the "current-to-best/1" strategy is ameliorated by Das *et al*. [121] The proposed model utilizes the optimal individual solution obtained in its small neighborhood to mutate each vector. A "current-to-pbest/1" strategy is presented by Zhang and Sanderson [107]. Their strategy not only uses the optimal individuals from "current-to-best /1" strategy, but also utilizes information from other good solutions. However, the inferior solutions obtained lately are also combined in this strategy. Yong *et al*. [35] study the performance of DE in terms of integrating several available generation strategies of trial vector to several superior control parameter settings. Meanwhile, a composite DE (CoDE) is also proposed to randomly integrate three selected recombination strategies with three control parameters settings to acquire trial vectors.

The convergence performance of DE and the robustness of solutions are improved by tuning the control parameters including NP, F, and C_r. Storn and Price [47] argue that the settings of three parameters are not difficult and suggest that $NP \in [5D, 10D]$, F can be chosen as 0.5, C_r can be treated as 0.1 or 0.9. On the contrary , it is proved that the performance of DE is very sensitive to control parameters [122]. To prevent premature convergence, it is demonstrated that the value of F should be larger than a problem-dependent threshold. If $F > 1.0$, the convergence performance can be improved. Although there are many different suggestions for control parameters, consensus has been reached that $F \in [0.4, 1.0]$, and C_r should be close to 1.0 or 0.0.

Some smart adaptive strategies are developed to tune the control parameters during DE evolution process. Two schemes are introduced to adapt F. One scheme randomly changes F and the other scheme linearly reduces F from a predefined maximum value [121]. A self-adaptive DE (jDE) is proposed in [105], in which both F and C_r are randomly adjusted on the basis of certain probabilities. An adaptive differential evolution with optional external archive (JADE) is developed by Zhang and Sanderson [107], which utilizes normal and Cauchy distributions to produce F and C_r for each target vector. In addition, recent F and C_r are adopted by JADE to generate new values. Different from the above methods, self-adaptive differential evolution (SaDE) is proposed in [106], which the trial vector generation strategies and control parameters are adjusted simultaneously by learning from the previous search.

3.2 Reactive Power Planning Formulation

Optimal allocation of volt-ampere reactive (VAR) sources including capacitor banks, static VAR compensators (SVCs), and static compensators (STATCOMs) are critical tasks in RPP or VAR planning [2]. Generally, locations for new VAR sources are either directly assumed or estimated. Recent research has presented some rigorous optimization-based methods for RPP. However, RPP problem is non-linear, non-convex, or even discrete in nature, traditional Newton-type optimization methods encounter difficulties in obtaining the global optimum. Indeed, because of the complicated objective functions, constraints and solution algorithms, RPP is treated as one of the most challenging problems for power systems. On the contrary, evolutionary computation methods have been developed for this purpose [1], [15], [123]. In this work, the proposed MQDE is adopted as the optimization tool to solve the planning problem with its rapid convergence and global optimization capability.

The reactive power planning problem requires the optimal determination of the locations, types, and sizes of equipment to be installed in a transmission system. The majority of RPP objectives are intended to provide new reactive power supplies at the least cost. Many variants of this objective include the cost of real power losses or the fuel cost. In addition, some technical indices such as the deviation from a given voltage schedule or the voltage stability margin can be used as objectives for optimization [124], [125]. The objective function can be formulated as

$$Minimize \quad F = C_1 P_{\text{loss}} + C_2 Q_{\text{c}} + C_3 V_{\text{SI}} + C_4 f(V, Q), \tag{3.7}$$

where P_{loss} denotes the real power losses, Q_{c} denotes the newly installed VAR source, V_{SI} is the voltage stability index (VSI) of the system [126], $f(V, Q)$ is a composite of inequality constraints, and each C can be treated as constant coefficient.

The VSI is given by

$$V_{\text{SI}} = \frac{a}{b + c \cdot \lambda_{\text{RQV}}}, \tag{3.8}$$

where a, b, and c are predetermined constants and λ_{RQV} is the smallest eigenvalue of the reduced Jacobian matrix $J_{dQdV}(V, \Theta)$.

The function $f(V, Q)$ is proposed to penalize infringement of voltage limits and reactive power generation. These constraints cannot be treated as equality constraints in normal power flow equations and are formulated as

$$f(V, Q) = \left(V - \frac{V_{\text{limit}}}{V_{\text{max}}} - V_{\text{min}}\right)^2 + \left(Q - \frac{Q_{\text{limit}}}{Q_{\text{max}}} - Q_{\text{min}}\right)^2, \tag{3.9}$$

where V and Q denote the violation values of bus voltage magnitudes and reactive power outputs, respectively. The value of $f(V, Q)$ equals to zero if V and Q are both within their range.

The coefficients C_1, C_2, and C_3 are adjusted by operators according to realistic requirements. The constant C_4 is a very large value adopted to increase the penalty for operation violation.

3.3 Proposed MQDE Algorithms

The classical DE algorithm is capable of solving various challenging optimization problems. However, it suffers from reduced population diversity at later stage of iterations leading to local optimum solution due to the greedy selection involved in DE. A countermeasure to effectively ensure the population diversity is to introduce the qubit concept, i.e. QDE algorithm [113]. Nevertheless, if traditional parent-child selection operator of classical DE remains for qubit individual update, the global searching capability of QDE algorithm will be impaired. Details of this will be discussed later on.

In view of deficiencies of classical and existing QDE algorithms, a newly modified QDE method, termed as MQDE, is proposed. MQDE has greater searching ability because it not only implements a combination of qubits, DE mutation and crossover operator but also introduces a novel reset strategy based on a more competitive DE mutation scheme DE/best/1/bin. The representation and the MQDE algorithm are presented below.

Representation: The smallest unit of information stored in a two-state quantum computer is called a qubit, which can be the state of 0, the state of 1, or in any superposition of the two states. The qubit is defined as follows:

$$\begin{bmatrix} \alpha_{j,i} \\ \beta_{j,i} \end{bmatrix}, \quad \begin{cases} j = 1, 2, \cdots, m \\ i = 1, 2, \cdots, n \end{cases} \tag{3.10}$$

and the state of a qubit can be expressed as

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{3.11}$$

where α and β are complicated numbers that specify the probability amplitudes of the corresponding states. The probability of the qubit will be found in the 0 state and is given by $|\alpha|^2$, and the probability of the qubit will be found in the 1 state and is given by $|\beta|^2$. j and i represent the dimension of qubit individuals and the population size, respectively. Normalization of the states to unity guarantees that

$$|\alpha|^2 + |\beta|^2 = 1 \tag{3.12}$$

QEA uses $[\alpha, \beta]^\top$ as the representation of qubits. However, the MQDE algorithm only uses a single variable θ in the application of the DE mutation and the crossover operator. Since (3.12) is the equation of a unit circle, each point can be represented by a single variable θ corresponding to the Cartesian co-ordinates given by $\cos\theta$ and $\sin\theta$, where θ is defined in $[0, \pi/2]$. It is noticed that the range of θ is confined within one quarter of a unit circle in our approach. This will help to reduce the computational effort while maintaining full coverage of all possible states. Hence, a qubit individual is a string of m qubits which is defined as

$$\overrightarrow{\theta_i} = [\theta_{1,i}, \theta_{2,i}, \cdots, \theta_{j,i}, \cdots, \theta_{i,m}] \tag{3.13}$$

Due to its probabilistic nature, a qubit is able to represent a linear superposition of all possible solutions. As a result, a total of 2^m individuals can be represented by combinations of different qubit states. This qubit representation has better characteristics than other representations for generating diversity in the population. For instance, if there exists a three-qubit individual with two pairs of amplitudes, such as

$$[\frac{\pi}{4}, \frac{\pi}{3}, \frac{\pi}{6}] \tag{3.14}$$

then the states of the individual can be represented as

$$\begin{aligned} \frac{\sqrt{6}}{8}|000\rangle + \frac{\sqrt{2}}{8}|001\rangle + \frac{3\sqrt{2}}{8}|010\rangle + \frac{\sqrt{6}}{8}|011\rangle \\ \frac{\sqrt{6}}{8}|100\rangle + \frac{\sqrt{2}}{8}|101\rangle + \frac{3\sqrt{2}}{8}|110\rangle + \frac{\sqrt{6}}{8}|111\rangle \end{aligned} \tag{3.15}$$

The above result means that the probabilities of the states $|000\rangle$, $|001\rangle$, $|010\rangle$, $|011\rangle$, $|100\rangle$, $|101\rangle$, $|110\rangle$, and $|111\rangle$ are $3/32$, $1/32$, $9/32$, $3/32$, $3/32$, $1/32$, $9/32$, and $3/32$, respectively. Hence, the three-qubit system contains the information of eight states.

Initialization: In MQDE algorithm, all the qubits (θ) of all the individuals in the population are initialized uniformly at random in $[0, \pi/2]$. The binary solution is derived by observing the states of $\overrightarrow{\theta_i}$ as follows:

$$P_{j,i} = \begin{cases} 1 & \text{if } rand_{i,j}[0,1] \leq \sin^2(\theta_{j,i}) \\ 0 & \text{otherwise} \end{cases} \tag{3.16}$$

Mutation: In QEA, the following rotation gate is utilized as a mutation operator to update a qubit of an individual [111]:

$$\begin{bmatrix} \alpha'_{j,i} \\ \beta'_{j,i} \end{bmatrix} = \begin{bmatrix} \cos(\Delta\theta_{j,i}) & -\sin(\Delta\theta_{j,i}) \\ \sin(\Delta\theta_{j,i}) & \cos(\Delta\theta_{j,i}) \end{bmatrix} \cdot \begin{bmatrix} \alpha_{j,i} \\ \beta_{j,i} \end{bmatrix} \tag{3.17}$$

where $\Delta\theta_{j,i}, j = 1, 2, \cdots, m, i = 1, 2, \cdots, n$, denotes rotation angle of each qubit toward either the 0 state or the 1 state. However, the magnitude and direction of rotation angle, which have an effect on the rate of convergence, must be determined in the process of optimization. At present, the direction of the rotation angle is usually determined by a query lookup table that inefficiently deals with numerous conditional judgments. For the above problem, a typical QDE uses one of the simplest forms of DE mutation operator, *DE/rand/1/bin*, to update qubit individuals. In this work, MQDE uses a different but more competitive DE mutation scheme, DE/best/1/bin, to cooperate with a novel update/reset strategy. The mutation operator in MQDE is similar to that of a classical DE, but it is applied to the qubit (θ) defined by (3.13) instead of operating directly on the individual. The mutant qubits of the ith individual in generation t are determined as follows:

$$\vec{\theta}_i^{mt} = \vec{\theta}_{\text{best}}^t + F \cdot (\vec{\theta}_{r_1}^t - \vec{\theta}_{r_2}^t) \tag{3.18}$$

where r_1, r_2, and i are distinct, F is the mutation constant that controls the amplification of the differential variation, and $\vec{\theta}_{\text{best}}^t$ is the qubit individual corresponding to the binary solution with the best fitness value of the current generation.

Crossover: The crossover operation is performed on the original qubits and the respective mutant qubits shown as:

$$\theta_{j,i}^{ct} = \begin{cases} \theta_{j,i}^{ct} & \text{if } rand_{i,j}[0,1] \leq C_R \text{ or } j = j_{rand} \\ \theta_{j,i}^{t} & \text{otherwise} \end{cases} \tag{3.19}$$

where the index j_{rand} is randomly chosen from $[1, 2, \cdots, m]$ which ensures that at least one qubit in each individual is different from the original. The crossover constant C_r controls the diversity of the population.

Selection: The diversity of the population decreases towards the later stages of the iteration process in the DE algorithm. The effectiveness of the mutation operator will be reduced by this kind of situation. If the iteration process is attracted by a local optimum and no continuous transitional local optimum exists between this local optimum and the global optimum, then the DE algorithm can hardly escape from the current local optimum. Once the concept of quantum computing has been introduced and the differential operator has been applied to the qubits, a typical QDE algorithm has the ability to obtain a better fitness value than the local optimum when the iteration process becomes trapped. This is due to the characteristic of the superposition of states and the randomness in measuring the state of qubits on a chromosome. A typical QDE uses the traditional greedy updating principle. By using (3.16) to observe the state of the newly obtained qubits (θ), a trial binary solution is obtained that replaces the corresponding binary solution in the population if the fitness values are higher. If so, the corresponding trial qubit individual replaces the current one. The replacement is made using the following equations:

$$P_i^{t+1} = \begin{cases} P_i^{ct} & \text{if } f(P_i^{ct}) \leq f(P_i^t) \\ P_i^t & \text{otherwise} \end{cases} \tag{3.20a}$$

and

$$\vec{\theta}_i^{t+1} = \begin{cases} \vec{\theta}_i^{ct} & \text{if } f(P_i^{ct}) \leq f(P_i^t) \\ \vec{\theta}_i^t & \text{otherwise} \end{cases} \tag{3.20b}$$

where P_i^{ct} is the ith binary solution found by observing the ith qubit individual after the crossover operation and $f(P_i)$ is the fitness value of the corresponding individual.

This updating mechanism for qubit individuals in a typical QDE has the following two disadvantages:

1. The process of measuring the state of qubits with (3.16) is based on a probabilistic comparison mechanism. There is no absolute one-to-one correspondence between the observed binary solution and the qubit individual in terms of quality. Hence, there is a lack of evolutionary guidance for a qubit individual in that the updating of a qubit individual is judged only according to the size of the fitness value by evaluating the two binary solutions as in (3.20).

2. As mentioned above, the purpose of introducing the qubit representation is to surmount the difficulty of escaping from a local optimum in the later stages of the DE iteration process. A typical QDE has the ability to obtain a better fitness value than the current local optimum. However, the selection operator of QDE interrupts the continuity of escaping from a local optimum. Due to the intrinsic differential property, when a typical QDE is trapped by a local optimum in the later stages of the iteration process, θ is far more likely to approach a high value (or low value) in the range $[0, \pi/2]$. At such a moment, if a binary solution with a better fitness value appeared under a low-probability case, this binary solution would have no effect on guiding the update direction of a qubit individual, owing to the selection strategy of QDE in (3.20). The winner of (3.20b) would remain in a high-value region (or low-value region), which would lead the typical QDE to lose the continuity in escaping from the local optimum.

To overcome the above problems, MQDE exploits a novel reset index. After the greedy selection for the current generation as in (3.20), MQDE selects the best binary solution

$P_{j,\text{best}}$ and the corresponding qubit individual $\theta_{j,\text{best}}$ to perform the following detection and reset:

$$\begin{aligned}
&\text{if} \quad && \theta_{j,\text{best}} \geq \tfrac{\pi}{4} \ \& \ P_{j,\text{best}} = 0, \\
& && \theta_{j,\text{best}} = \tfrac{\pi}{8}; \\
&\text{else if} \quad && \theta_{j,\text{best}} < \tfrac{\pi}{4} \ \& \ P_{j,\text{best}} = 1, \\
& && \theta_{j,\text{best}} = \tfrac{3\pi}{8}; \\
&\text{end} &&
\end{aligned} \tag{3.21}$$

This new best qubit individual for the current iteration obtained by the reset index is then incorporate by the mutation operator (DE/best/1/bin) as in (3.18). This combination provides an explicit guidance about qubits update direction. Consequently, the two disadvantages mentioned above will be successfully overcome.

It is worth noting that the reset values for $\theta_{j,\text{best}}$ are selected as $\pi/8$ and $3\pi/8$. These values are experimentally found to give a better solution. A reasonable explanation is the following: if the reset values are selected as 0 and $\pi/2$, the diversity of a qubit individual would be destroyed; conversely, if the reset value are uniformly selected as $\pi/4$, the guidance for the update direction of a qubit individual would be impaired. The pseudo-code of MQDE is provided in Table 3.1.

Table 3.1. Pseudo code of MQDE

```
Procedure MQDE
Begin
    t ← 0
    initialize Q(t)
    make P(t) from Q(t) by (3.17)
    evaluate P(t) by function evaluation
    P_best ← best solution among P(t)
    θ_best ← qubit chromosome corresponding to P_best
    while  t < T  do
        t ← t + 1
        generate mutant qubits by (3.18)
        do crossover by (3.19)
        make P(t) from Q(t) by (3.17)
        evaluate P(t)
        update Q(t + 1), P(t + 1) by (3.20)
        update P_best accordingly
        update θ_best by (3.21)
    end while
end
```

3.4 Case Studies

3.4.1 Validation with Benchmark Functions

To verify the performance of MQDE, benchmarks from IEEE CEC 2006 [127] are introduced. These test functions involve various kinds (linear, nonlinear, polynomial, quadratic, and cubic) of objective functions with different numbers of decision variables and different kinds of constraints. For the purpose of comparison , DE, real-coded genetic algorithm (RGA) [45] and PSO [128] are also adopted in the case study. The parameters in MQDE are chosen as $F = 0.01$, $C_r = 0.8$. Gray coding is used to convert from binary

Table 3.2. Performance comparisons of MQDE, DE, RGA and PSO

Prob.	Method	Best	Mean	Worst	Std Dev
g01	MQDE	-15.000	-15.000	-15.000	2.142E-13
	DE	-15.000	-15.164	-15.371	1.263E-01
	RGA	-15.000	-15.000	-15.000	1.571E-07
	PSO	-15.000	-15.000	-15.000	2.379E-08
g05	MQDE	5130.659	5130.659	5130.659	5.826E-17
	DE	5172.852	5403.971	6474.813	8.936E+01
	RGA	5287.270	5301.125	5446.711	1.852E+01
	PSO	5129.834	5130.589	5134.856	2.398E-01
g09	MQDE	694.526	695.147	697.882	3.515E+00
	DE	680.630	680.630	680.633	2.721E-03
	RGA	706.531	760.381	857.545	5.614E+01
	PSO	680.630	680.630	680.630	6.957E-11
g10	MQDE	7049.378	7049.378	7049.378	4.437E-12
	DE	7131.163	7221.182	7507.863	5.321E+01
	RGA	7371.652	7742.331	8213.455	8.261E+01
	PSO	7049.248	7091.336	7156.653	1.121E+01
g13	MQDE	0.054	0.054	0.054	1.383E-11
	DE	0.054	0.054	0.054	6.524E-07
	RGA	1.715	2.389	2.713	9.365E-01
	PSO	0.054	2.136	3.861	1.143E+00

string to real value. For DE, RGA and PSO, the parameters are set as recommended in the corresponding references and can be found in [45], [47], [128]. The constraint-handling technique based on preference of feasible solutions over infeasible solutions [129] is implemented to all the methods. For each test function, all methods run 25 times independently and are stopped when a maximum of 5×10^3 fitness evaluations is reached. All the experiments are executed on a 2.50 GHz, Intel Core i5, with 4G RAM PC. Table 3.2 shows the best, mean, worse and standard deviation of the objective function value of the resulting solutions over 25 independent runs.

In the cases of g05, g09 and g10, results indicate that the performance of PSO is better than MQDE in terms of the accuracy of best optimum. A reasonable explanation is: After introducing the qubit concept to DE, the dimension of the population increases dramatically. In the later stages of the iteration process while the current fitness value lays in the adjacent region of the global optimum, those bits of the current best binary solution whose values mainly determine the accuracy of the fitness value may have the opportunity to evolve to the best corresponding to the global optimal binary solution. However, at such a moment this scenario occurs, the algorithm cannot guarantee that other bits still remain at the best value. Further, the smaller the difference between the current fitness value and the global optimum, the harder it is to achieve the simultaneously occurrence of the above two cases. Hence, the selection operator as well as the proposed reset index can hardly keep the best value for those bits which mainly determine the accuracy of the fitness value. Consequently, in some cases, MQDE cannot beat PSO in terms of accuracy of the best optimum.

Summarizing the performance of all the four algorithms, the promising results of MQDE illustrate that the algorithm not only successfully surmounts the disadvantages of DE as mentioned above, but also can be used as a powerful and robust tool for mathematical optimizations.

3.4.2 MQDE for Power System RPP

The advantages of the proposed MQDE algorithm in solving mathematical optimization problems have been demonstrated. Then, it is employed to effectively solve the planning problem of power system. Subsequently, a capacitor configuration task on IEEE 30 bus system is utilized to verify the proposed MQDE algorithm. The system consists of 5 generators, 41 branches and 21 load buses and the detailed system data can be obtained in [9]. Bus#19, 23, 26 and 29 are selected as the candidate compensation buses based on the sensitivity analysis [130]. Newton-Raphson method is employed to calculate the power flow and is realize by Matpower [131]. The population size is fixed as 20 and the maximum iteration is set as 1000. Based on the data obtained from 100 trials, the comparisons between the two methods are illustrated in Table 3.3 and Figure 3.1, which shows that the MQDE also succeeded in finding satisfactory solution for the tested problem and MQDE has shown the superiority to the typical DE in terms of robustness.

Table 3.3. Comparison with different approaches before and after compensation for RPP

	Shunt Compensation (MVar)					P_{loss} (MW)	V_{SI}	$f(V,Q)$
	Bus#19	Bus#23	Bus#26	Bus#29	Bus#19			
Original	0	0	0	0	0	49.457	1.000	0.170
DE optimal	4.371	5.691	5.877	9.131	4.371	46.363	0.892	0
MQDE optimal	4.360	5.660	5.870	9.120	4.360	46.371	0.903	0

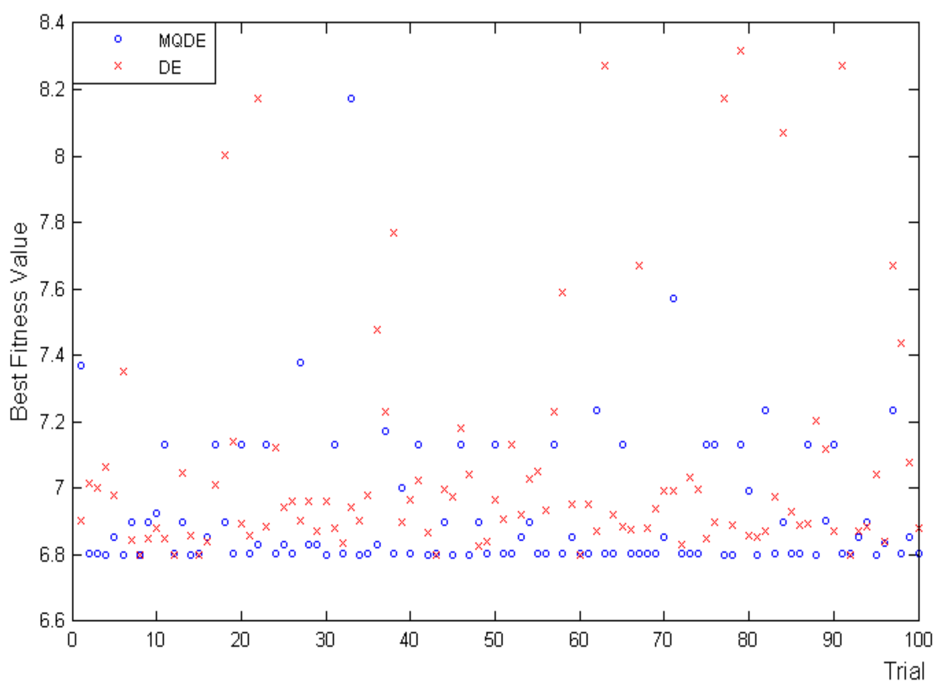

Figure 3.1. Distribution of fitness values of the two approaches

Chapter 4 Adaptive Range Composite Differential Evolution for Fast Optimal Reactive Power Dispatch

This chapter proposes a novel adaptive range composite differential evolution (ARCoDE) algorithm targeting at practical applications such as the fast ORPD. The proposed ARCoDE utilizes the concept of compositing different types of trial vector generation strategies [35], which is able to provide a decent balance for the algorithm between the exploration and exploitation capabilities. In addition, a novel control parameter range adaptation mechanism is proposed to enable a highly efficient adaptive tuning of control parameters. These novel properties effectively support ARCoDE to conquer the difficulties introduced by the limited numbers of function evaluations due to the critical time requirements in many practical applications including ORPD problem.

4.1 ORPD Problem Formulation

The classical ORPD problem can be formulated as a complicated mixed integer nonlinear optimization model, which consists of many nonlinear constraints and discrete/continuous decision variables. In general, the application of HOAs to solve the ORPD problems may suffer from the relatively long computational time. In addition, the fine-tuning of control parameters needs numerous amounts of trial and error tests. Moreover, due to the uncertainties introduced by load and the increased penetration of renewable energy generation such as wind and solar power [20], [39], [132], [133], the solutions provided by HOAs are less likely robust.

Typically, the ORPD problem aims at minimizing the total power losses involved in every transmission line, expressed as,

$$\textit{Minimize}:\ f_Q = \sum_{k\in(i,j)} P_{\mathrm{loss},k} = \sum_{i=1}^{L}\sum_{j=1}^{L}\left[g_{ij}\left(|V_i|^2 + |V_j|^2 - 2|V_i|\,|V_j|\cos\left(\delta_i - \delta_j\right)\right)\right] \quad (4.1)$$

where f_Q is the sum of power losses to be minimized; $P_{\mathrm{loss},k}$ is the active power loss for the transmission line k; L is the number of bus nodes; $|V_i|$ and δ_i are the voltage magnitude and phase angle at the i-th bus node; g_{ij} is the conductance between the node i and j.

The decision variables can be categorized as the control variables x and dependent variables u. Specifically, the control variables x include voltages at generation buses V_{G}, reactive power compensation of the shunt capacitors and inductors Q_{C}, and transformer tap settings T,

$$x^{\top} = [V_{\mathrm{G}}, Q_{\mathrm{C}}, T], \quad (4.2)$$

The dependent variables u consist of the voltage of load buses V_{PQ}, reactive power outputs of generators Q_{G} and power flow of transmission lines S_{L}, defined as,

$$u^{\top} = [V_{\mathrm{L}}, Q_{\mathrm{G}}, S_{\mathrm{L}}]. \quad (4.3)$$

Both equality and inequality constraints are involved in the ORPD problem. The equality constraints are mainly related to the alternating current power balance equations, defined as,

$$P_{\mathrm{G}_i} - P_{\mathrm{D}_i} - P_{i,\mathrm{loss}}^{\mathrm{B}} = 0, \quad Q_{\mathrm{G}_i} - Q_{\mathrm{D}_i} - Q_{i,\mathrm{loss}}^{\mathrm{B}} = 0, \quad i \in N_{\mathrm{B}} \tag{4.4}$$

where P_{G_i} and Q_{G_i} are the active and reactive power injection to the bus node i respectively, P_{D_i} and Q_{D_i} represent the active and reactive load at bus i respectively, $P_{i,\mathrm{loss}}^{\mathrm{B}}$ and $Q_{i,\mathrm{loss}}^{\mathrm{B}}$ denote the active and reactive power losses at bus i respectively, and N_{B} is the number of bus nodes. The boxed inequality constraints include,

$$V_i^{\min} \leq V_i \leq V_i^{\max}, \ i \in N_{\mathrm{B}}, \tag{4.5}$$

$$Q_{\mathrm{G}_i}^{\min} \leq Q_{\mathrm{G}_i} \leq Q_{\mathrm{G}_i}^{\max}, \ i \in N_{\mathrm{G}}, \tag{4.6}$$

$$Q_{\mathrm{C}_i}^{\min} \leq Q_{\mathrm{C}_i} \leq Q_{\mathrm{C}_i}^{\max}, \ i \in N_{\mathrm{C}}, \tag{4.7}$$

$$T_i^{\min} \leq T_i \leq T_i^{\max}, \ i \in N_{\mathrm{T}}, \tag{4.8}$$

$$S_{\mathrm{L}_i} \leq S_{\mathrm{L}_i}^{\max}, \ i \in N_{\mathrm{B}}, \tag{4.9}$$

where constraints (4.5)—(4.9) define the limits for bus voltage V_i, reactive power generation Q_{G_i}, reactive power compensation Q_{C_i}, transformer tap position T_i and branch power flow S_{L_i}, respectively; N_{G}, N_{C}, and N_{T} denote the total number of generator buses, buses with shunt elements, and transformer tap positions respectively.

Traditionally, the ORPD problem can be solved by deterministic optimization algorithms such as interior point method or approximately transformed into a convex optimization problem. However, the global optimality of the converged solution can hardly be guaranteed due to the inherent nonconvexity involved in the original ORPD problem. Besides, convex approximations and inaccurate simplifications of the ORPD might result in infeasible solutions for practical application. In contrast, HOAs can be directly applied to solve the original formulations without simplifications. In addition, HOAs are the one of the most prevalent methods to cope with the global optimality search for the non-differentiable and non-convex mixed integer optimization problems. However, the concerns about solution robustness and computational time hinder their practical applicability, which are particularly attended in the proposed ARCoDE algorithm.

4.2 Adaptive Range Composite DE for ORPD

In Chapter 3, different generation strategies for trial vector and the control parameter tuning have been extensively investigated. However, there are significant restrictions of those methods when dealing with the practical ORPD problem that demands fast computation speed as well as solution robustness. It is mainly because that the ORPD problems in practice are solved in short time intervals. Only a moderate number of function evaluations for those strategies are allowed to well adapt the candidate vector and control parameters. Regarding the aforementioned practical requirement and bottleneck, this work proposes a novel ARCoDE method aiming at efficiently solving the ORPD problem. One of the most critical procedures lies in the incorporation of two candidate vector generation mechanisms with two adaptive ranges of control parameters in each iteration for creating offspring candidate vectors.

Different from the original CoDE, the new algorithm is characterized by the faster convergence while preserves consistent population diversity during evolutions. To achieve fast convergence with limited function evaluations, two trial vector generation strategies including DE/best/2/bin and DE/rand/2/bin [47] are employed in the proposed ARCoDE algorithm. Instead of setting specific values for control parameters F and C_r, explorative and exploitative parameter ranges are innovatively introduced in the proposed algorithm. During the evolution process, the algorithm gradually adjusts the chosen probabilities and the sizes of these ranges. At each generation, each candidate vector generation mechanism from the pool is utilized to create an offspring candidate. According to the feedback probability, the algorithm control parameters are determined within their ranges F and C_r. Accordingly, two candidate vectors of their corresponding targets are generated. Then the superior candidate would be employed for the next iteration on condition that the candidate surpasses its target. Table 4.1 presents the pseudo-code of ARCoDE, details of which are illustrated below.

4.2.1 Candidate Vector Generation Strategies

To allow fast convergence, the greedy combination strategies by eliciting the winner information during the evolution process are adopted. Consequently, the DE/best/2/bin is selected as one of the candidate recombination strategies in the pool. However, such strategy may easily lead the evolutionary of the population to local optimums. To prevent the premature convergence resulted from such a greedy strategy, the DE/rand/2/bin strategy is also included to further enhance the explorative capability of the proposed algorithm. It employs the addition of two difference vectors to the base vector, giving rise to more diverse perturbation in comparison with the DE/rand/1/bin using merely a single difference vector.

4.2.2 Control Parameter Adaptation

As discussed above, solving ORPD problems in practice may need limited function evaluations for fast convergence. The existing adaptive strategies including jDE, JADE, and SaDE can hardly guarantee to gauge suitable values for F and C_r in such a small number of function evaluations. Instead, it is promising to evolve suitable ranges for these control parameters during the iteration process. Generally, larger values of parameter F result in more dispersive distribution of the mutant vectors over the searching region and helps to improve individual diversity. On the contrary, small F values make the explored candidates concentrating on the current ones, thus accelerating the convergence. Moreover, a large C_r makes the candidate vector severely deviated from the target, and the offspring could inherit little information from their ancestors. Consequently, sufficient diversity of the offspring candidates is maintained. Small C_r values are promising to fit the separable problems fairly well, in this case, the difference between the candidate and the target might reside in one component.

Motivated by the observations above, explorative and exploitative ranges for F and C_r, respectively are defined based on the characteristic of these control parameters, defined as follows,

$$\text{Explorative range of } F\text{:}\ \ F^{er} = [0.7, 0.9], \tag{4.10}$$

$$\text{Exploitative range of } F\text{:}\ \ F^{et} = [0.5, 0.7], \tag{4.11}$$

$$\text{Explorative range of } C_r\text{:}\ \ C_r^{er} = [0.8, 0.9], \text{ and} \tag{4.12}$$

$$\text{Exploitative range of } C_r\text{:}\ \ C_r^{et} = [0, 0.2]. \tag{4.13}$$

The initial probabilities of applying different range to each individual are set to 0.5, i.e. $P_{F/C_r}^{er/et} = 0.5$. Consequently, each range owns the same probability to be applied by the candidates in the beginning of iterations. Roulette Wheel selection based on the probability is utilized to determine the range for each candidate in the current population. Thereafter, a F/C_r value will be randomly selected within this range and assigned to the corresponding individual. After evaluating all the latest candidates, the number of candidates pertaining to different ranges and entering the next generation is recorded by $NS_{F/C_r}^{er/et}$, and the number of the discarded candidates is recorded by $NF_{F/C_r}^{er/et}$. The counters $NS_{F/C_r}^{er/et}$ and $NF_{F/C_r}^{er/et}$ are cumulated after a certain number of generations, which is called *learning period.* Subsequently, the probability can be calculated according to:

$$P_{F/C_r}^{er/et} = \frac{NS_{F/C_r}^{er/et}}{NS_{F/C_r}^{er/et} + NF_{F/C_r}^{er/et}}, \tag{4.14}$$

which is actually the ratio of the evolved anew winners among the all the newly generated candidates by each range during the *learning period*. Consequently, the probabilities of applying each range should be refreshed every generation after the *learning period*. When the evolutionary process reaches a pre-defined point, the range with lower percentage of success rate is discarded， and the range with higher percentage of success rate will be split in half, and then the aforementioned procedures are repeated with this new range. This adaptation procedure is capable of gradually evolving suitable ranges for parameters F and C_r, which requires a relatively small number of function evaluations.

4.2.3 Constraint Handling Strategy

This work adopts the constraint handling method proposed by Deb and based on the superiority of feasible solutions [128]. There is no parameter to be tuned for Deb's selection criterion, consistent with one of the main motivations of this study. The constraint handling rules are incorporated with the optimum searching below:

In the procedure of candidate selection, the candidate A is compared to the candidate B taking both the objective value and constraint violations into account. The candidate A will take the place of B and be added in the subsequent evolution once any of the following conditions is satisfied.

1. Candidate A is feasible and candidate B is infeasible.
2. Both candidates are feasible, but candidate A reaches better objective value.

3. Both candidates are infeasible, but candidate A has a smaller overall constraint violation.

Table 4.1. Pseudo-code of ARCoDE

Input: *NP*: the number of candidates for each generation.
Max_FES: maximum number of function evaluations.
The strategy candidate pool: "DE/best/2/bin" and "DE/rand/2/bin.
Initial ranges of the control parameters: $F^{er} = [0.7, 0.9]$; $F^{et} = [0.5, 0.7]$; $C_r^{er} = [0.8, 1]$; $C_r^{et} = [0, 0.2]$.
Initial probabilities of each range: $P_{F/Cr}^{er} = P_{F/Cr}^{et} = 0.5$.
(1) $G = 0$; Initialize the candidate population $P_0 = \{\vec{X}_{1,0}, \cdots, \vec{X}_{NP,0}\}$ by uniformly sampling within the feasible region;
(2) Calculate the objective values for all the candidates $f(\vec{X}_{1,0}), \cdots, f(\vec{X}_{NP,0})$;
Evaluate the constraint violation $g(\vec{X}_{1,0}), \cdots, g(\vec{X}_{NP,0})$;
(3) $FES = NP$;
(4) **while** $FES < Max\_FES$ do
(5) $P_{G+1} = \phi$;
(6) **for** $i = 1: NP$ do
(7) Calculate parameter range probabilities $P_{F/Cr}^{er/et}$ using equation (4.14) and update the success and fail memory. Then apply Roulette Wheel selection to select the ranges;
(8) Generate two candidate vectors $\vec{u}_{i\_1,G}$ and $\vec{u}_{i\_2,G}$ for the target $\vec{X}_{1,G}$ based on the two candidate vector generation mechanisms with control parameters determined according to the ranges obtained by Step (7);
(9) Calculate the objective values and the constraint violation values of the two candidates $\vec{u}_{i\_1,G}$ and $\vec{u}_{i\_2,G}$;
(10) Choose the best trial vector from the two trial vectors $\vec{u}_{i\_1,G}$ and $\vec{u}_{i\_2,G}$, and the target vector$\vec{X}_{1,G}$ according to Deb's selection criterion;
(11) $FES = FES + 2$;
(12) **end for**
(13) $G = G + 1$;
(14) **end while**
Output: the candidate with the best objective function value or the smallest constraint violation value in the population

4.3 Case Studies

A 41-bus offshore wind power plant (WPP) ORPD test case in the 2014 IEEE Competition on "Application of Modern Heuristic Optimization Algorithms for Solving Optimal Power

Flow Problem" is used to validate the effectiveness of the proposed ARCoDE. It is known to all that the inclusion of wind power introduces much uncertainty to the system operation [39], [134]–[136]. This test case consists of 18 continuous variables including the reactive power set-points of wind power generators and one additional continuous variable defining the adjustment of reactor. The tap positions of the adjustable on-load transformers are modelled by 2 integer variables. The switchable shunt capacitor is described by 1 discrete variable. In the fast ORPD problem, the reactive power requirements with respect to the actual operating condition are determined by the stepwise variations of reactive power requirements (q_{ref}) throughout 24 hours. Considering 15-min intervals, the case results in 96 scenarios, some of which turn out to be hard-to-solve optimization tasks. According to (4.1)—(4.9), the target of the problem is to minimize the active power losses subject to the power system operational constraints, given limited number of function evaluations. The WPP ORPD problem contains 96 scenarios, among which the 13 most challenging scenarios are selected for the test purpose. These 13 scenarios are particularly selected since their feasible solutions can hardly be found by those award-winning algorithms in the competition. The topology of the offshore WPP system is shown in Figure. 4.1, and a detailed description of test cases and the competition results can be found in [137].

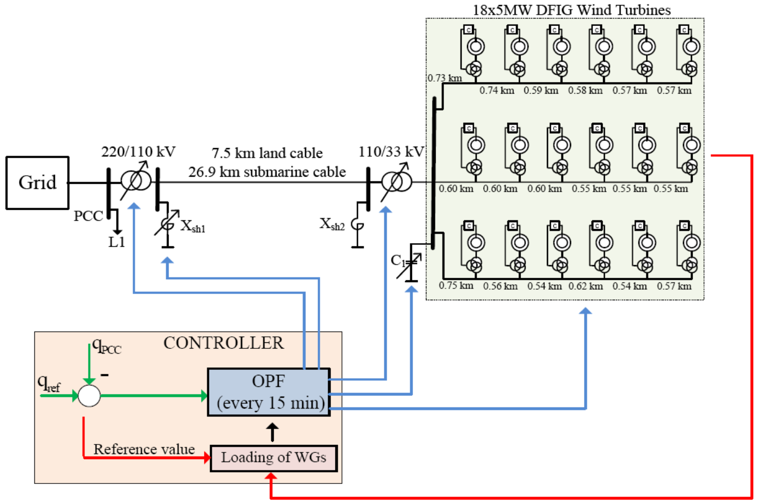

Figure 4.1. 41 bus offshore WPP ORPD test case [32], [137].

4.3.1 Numerical Results of ARCoDE on the Test Cases

The test environment is a DELL Desktop Workstation with Intel (R) Xeon (R) CPU E5-2650 v2 @2.60GHz RAM 64GB. Table 4.2 shows the numerical results (including the best solutions, objective values, the sum of constraint violations, and the average computation time through 31 trials) obtained by the proposed ARCoDE algorithm, where WG_i_Q with i=1, 2,..,18 represent the reactive power set-points of wind generators; OLTC_T_j with j=1, 2 represent the tap position of stepwise adjustable on-load transformers; C_1 represents the stepwise adjustment of capacitor; X_{sh1} represents the adjustment of reactor; obj_best represents the best objective value (power loss) according to (4.1) obtained by ARCoDE, and gvar_best means the smallest sum of different constraint violations. It can be observed that the computation time of each scenario is about 60s, which reflects the satisfactory efficiency for practical application.

Table 4.2. Best results obtained by ARCoDE under 13 scenarios

Scenario	50	51	52	53	75	76	77	78	79	80
WG1_Q	1.644	1.633	1.650	1.650	-1.546	-1.650	-1.650	-1.650	-1.650	-1.650
WG2_Q	1.647	1.647	1.650	1.649	-1.604	-1.650	-1.650	-1.650	-1.650	-1.650
WG3_Q	1.519	1.648	1.650	1.650	-0.891	-1.650	-1.650	-1.650	-1.650	-1.650
WG4_Q	1.616	1.601	1.650	1.650	-0.816	-1.650	-1.650	-1.650	-1.650	-1.650
WG5_Q	1.594	1.643	1.650	1.650	-0.601	-1.650	-1.650	-1.650	-1.650	-1.650
WG6_Q	1.643	1.571	1.650	1.650	-0.394	-1.650	-1.650	-1.650	-1.650	-1.650
WG7_Q	1.616	1.650	1.650	1.650	-1.636	-1.650	-1.650	-1.650	-1.650	-1.650
WG8_Q	1.645	1.650	1.650	1.650	-1.222	-1.650	-1.650	-1.650	-1.650	-1.650
WG9_Q	1.630	1.618	1.650	1.650	-1.543	-1.650	-1.650	-1.650	-1.650	-1.650
WG10_Q	1.626	1.649	1.650	1.650	-1.353	-1.650	-1.650	-1.650	-1.650	-1.650
WG11_Q	1.612	1.639	1.650	1.650	-1.141	-1.650	-1.650	-1.650	-1.650	-1.650
WG12_Q	1.628	1.633	1.650	1.650	-1.100	-1.650	-1.650	-1.650	-1.650	-1.650
WG13_Q	1.615	1.645	1.650	1.650	-0.907	-1.650	-1.650	-1.650	-1.650	-1.650
WG14_Q	1.604	1.646	1.650	1.650	-1.541	-1.650	-1.650	-1.650	-1.650	-1.650
WG15_Q	1.594	1.638	1.650	1.650	-1.507	-1.650	-1.650	-1.650	-1.650	-1.650
WG16_Q	1.575	1.610	1.650	1.650	-1.378	-1.650	-1.650	-1.650	-1.650	-1.650
WG17_Q	1.625	1.627	1.650	1.650	0.759	-1.650	-1.650	-1.650	-1.650	-1.650
WG18_Q	1.606	1.643	1.650	1.650	-0.854	-1.650	-1.650	-1.650	-1.650	-1.650
OLTC_T1	1.0993	1.0993	1.0993	1.0993	0.9669	0.9669	0.9669	0.9669	0.9669	0.9669
OLTC_T2	0.8700	0.8700	0.8700	0.8700	1.1083	1.1300	1.1300	1.1300	1.1300	1.1300
C_1	-12.100	-12.100	-12.100	-12.100	-4.033	-4.033	-4.033	-4.033	-4.033	-4.033
X_{sh1}	0	0	0	0	9.8965	9.8965	9.8965	9.8965	9.8965	9.8965
obj_best (MW)	1.433	1.379	1.280	1.216	2.011	2.637	2.637	2.637	2.637	2.637
gvar_best	0	0	0	0	0	0	0	0	0	0
Computation time* (s)	64.128	63.910	64.522	64.085	55.990	66.266	65.469	63.703	63.221	66.484

*: The average value through 31 running trials
Units are MVar if not specified.

4.3.2 Comparison with the Award-Winning Algorithms in the Competition

The mean and standard deviation of fitness values (evaluated by the benchmark program provided by [137]) from ARCoDE are compared with those from the top 3 ranking algorithms in the IEEE Competition on "Application of Modern Heuristic Optimization Algorithms for Solving Optimal Power Flow Problem", i.e. improved (μ+λ)-constrained differential evolution (ICDE) [138], differential evolution particle swarm optimization (DEEPSO) [139] and MVMO. The function evaluation numbers in all these methods are assigned to be 10000 according to the competition rules. Such a short function evaluations makes it extremely difficult to find feasible solutions of the complicated ORPD problem for conventional HOAs.

Table 4.3. Experimental results of ICDE, DEEPSO, MVMO and ARCoDE over 31independent trials

Scenario	ICDE Mean (Std)	DEEPSO Mean (Std)	MVMO Mean (Std)	ARCoDE Mean (Std)
50	2.428E+04(4.567E+04)	1.422E+00(0.004E+00)	1.430E+00(0.016E+00)	1.437E+00(0.003E+00)
51	9.079E+04(1.096E+05)	1.373E+00(6.384E-04)	1.380E+00(0.010E+00)	1.382E+00(0.003E+00)
52	3.760E+05(2.803E+05)	1.280E+00(3.920E-05)	1.280E+00(5.043E-05)	1.280E+00(7.112E-06)
53	4.482E+05(2.999E+05)	1.216E+00(5.257E-05)	1.216E+00(7.427E-05)	1.216E+00(8.466E-06)
54	3.770E+05(2.412E+05)	1.258E+00(2.251E-05)	1.258E+00(2.719E-05)	1.258E+00(3.225E-06)
55	2.944E+05(1.089E+05)	1.261E+00(4.576E-05)	1.261E+00(4.534E-05)	1.261E+00(5.289E-06)
56	9.737E+03(2.293E+04)	1.428E+00(0.003E+00)	1.435E+00(0.016E+00)	1.445E+00(0.002E+00)
75	2.968E+01(1.471E+02)	7.550E+01(4.092E+02)	2.022E+00(0.003E+00)	2.011E+00(5.296E-04)
76	3.895E+05(2.278E+05)	1.270E+06(5.761E+06)	2.637E+00(3.740E-08)	4.444E+05(2.328E+06)
77	3.281E+05(1.986E+05)	9.233E+05(5.118E+06)	2.637E+00(2.497E-08)	2.637E+00(1.796E-08)
78	3.186E+05(1.941E+05)	6.464E+05(3.600E+06)	7.086E+03(3.944E+04)	7.016E+04(3.841E+05)
79	2.937E+05(1.696E+05)	2.008E+04(1.118E+05)	2.637E+00(3.005E-08)	5.776E+05(3.002E+06)
80	3.341E+05(1.841E+05)	2.637E+00(7.478E-08)	2.637E+00(3.005E-08)	2.637E+00(1.356E-05)
-	0	3	3	N.A.
+	13	2	1	N.A.
~	0	8	9	N.A.

"Mean" and "Std" indicate the average and standard deviation of the function fitness values obtained in 31 runs, respectively.
"-", "+", and "~" denote that the performance of the corresponding algorithm is better than, worsen than, and similar to that of ARCoDE, respectively.
Units are MVar if not specified.

Table 4.3 illustrates the means and standard deviations achieved by different algorithms with 31 independent trials. Their performance under different scenarios is outlined in the last three rows of Table 4.3. On these 13 test scenarios, the proposed ARCoDE performs significantly better than the ICDE. The performances of ARCoDE and DEEPSO are quite similar. For test scenarios 76 and 79, MVMO shows a better performance in terms of optimality and robustness than the other 3 competitors. However, it should be noted that the proposed ARCoDE does not need to fine-tune the pre-defined control parameter setting, which indicates a significant advantage over the other three algorithms. In Table 4.3, symbols for minus, plus, and tilde represent that the performance of the corresponding algorithm is better than, worse than, and similar to that of ARCoDE. For example, the MVMO wins the ARCoDE in 3 scenarios, but loses in 1 scenario. They perform similarly in the rest 9 scenarios out of the total 13. Notably, the ARCoDE needs no local search operator that increases computational complexities and exists in the DEEPSO and MVMO. In terms of the constraint handling, ARCoDE is much simpler than ICDE, which utilizes the concept of multi-objective optimization. The preference of feasible solutions makes the constraint handling part of ARCoDE less computationally complex.

In summary, the proposed ARCoDE algorithm outperforms the three benchmarks. The mean fitness value evolution process of the ICDE, DEEPSO, MVMO, and ARCoDE with respect to the number of function evaluations under four typical scenarios are displayed in Figure. 4.2—Figure. 4.5.

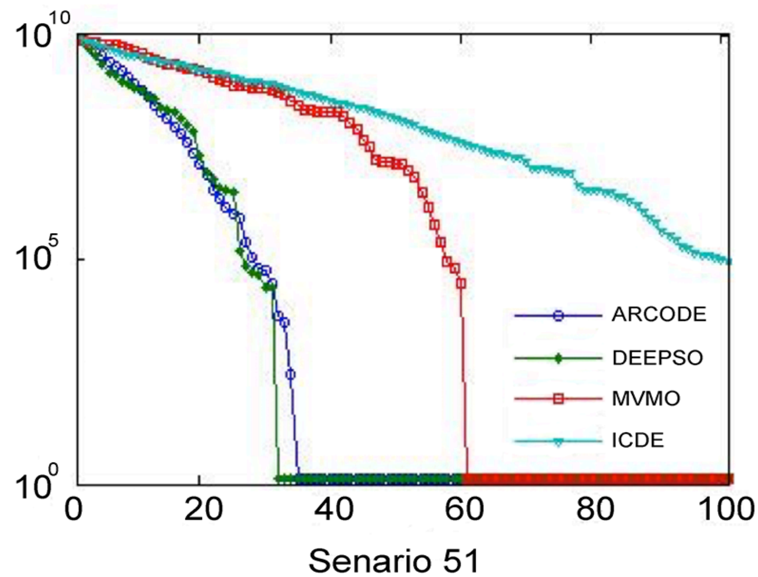

Figure 4.2. Mean fitness value evolution process of the modified ICDE, DEEPSO, MVMO, and ARCoDE with respect to the number of function evaluations on Scenario 51

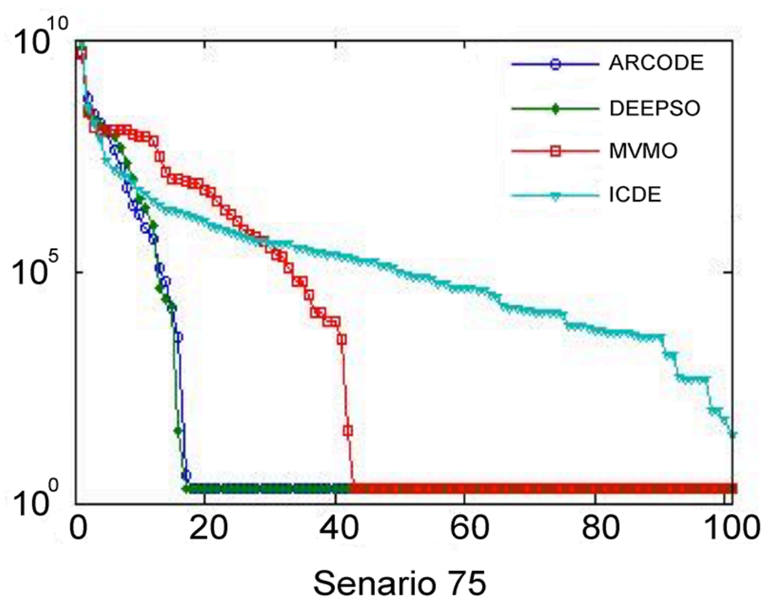

Figure 4.3. Mean fitness value evolution process of the modified ICDE, DEEPSO, MVMO, and ARCoDE with respect to the number of function evaluations on Scenario 75

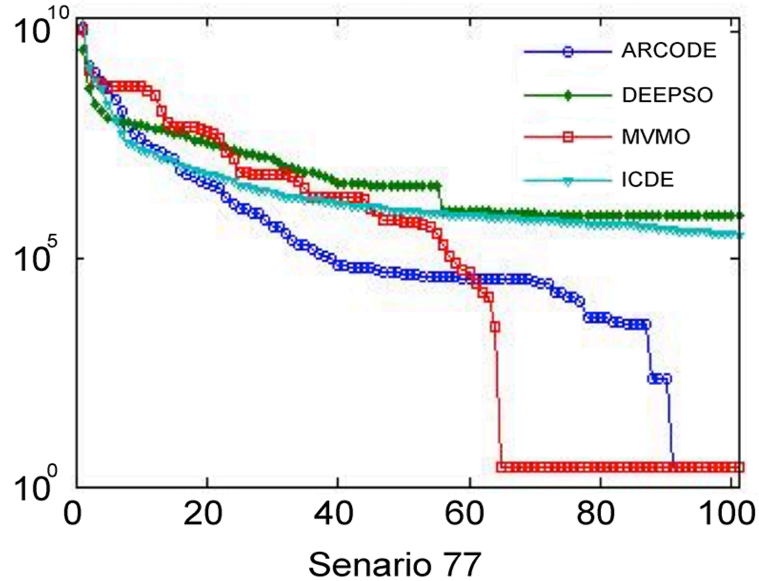

Figure 4.4. Mean fitness value evolution process of the modified ICDE, DEEPSO, MVMO, and ARCoDE with respect to the number of function evaluations on Scenario 77

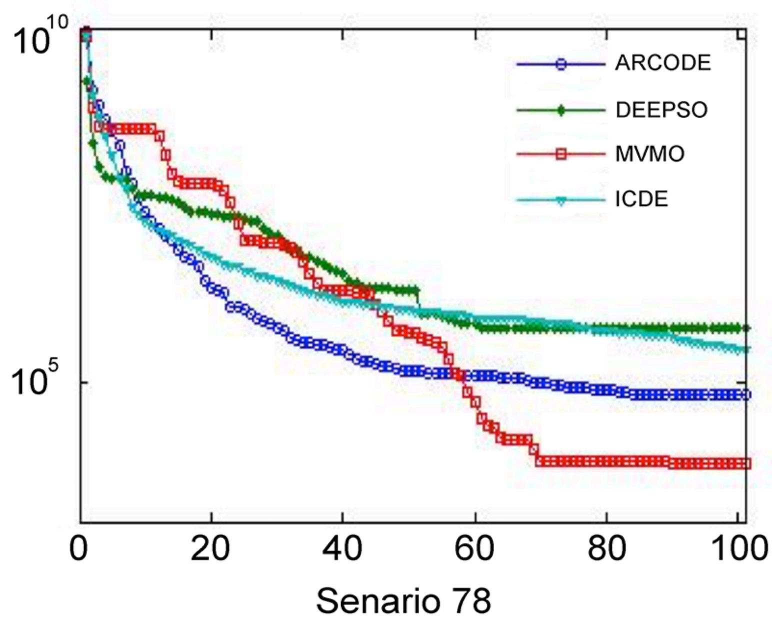

Figure 4.5. Mean fitness value evolution process of the modified ICDE, DEEPSO, MVMO, and ARCoDE with respect to the number of function evaluations on Scenario 78

4.3.3 Comparison among Various DE Algorithms

To demonstrate the advantages of the proposed adaptive range of control parameter settings, ARCoDE is also compared with three adaptive DEs including jDE, JADE, and SaDE. In jDE, JADE, and SaDE. In this experiment, the control parameters F and C_r are self-adapted for the ARCoDE algorithm, while the parameter setting for these three benchmarks is referred to the original papers. For all these four DEs, the function evaluation number is set to 10000, and each method is executed 31 times on each test cases. Table 4.4 summarizes the experimental results.

Table 4.4. Experimental results of jDE, JADE, SaDE and ARCoDE over 31independent trials

Scenario	jDE Mean (Std)	JADE Mean (Std)	SaDE Mean (Std)	ARCoDE Mean (Std)
50	3.579E+09(9.754E+08)	1.450E+03(1.987E+03)	9.291E+02(1.747E+03)	1.437E+00(0.003E+00)
51	4.299E+09(1.083E+09)	6.951E+05(1.200E+06)	2.044E+06(3.206E+06)	1.382E+00(0.003E+00)
52	4.777E+09(1.056E+09)	4.755E+06(90.96E+06)	9.533E+06(9.080E+06)	1.280E+00(7.112E-06)
53	4.761E+09(1.222E+09)	3.338E+06(7.740E+06)	6.235E+06(7.514E+06)	1.216E+00(8.466E-06)
54	5.103E+09(9.391E+08)	6.839E+06(1.012E+07)	1.011E+07(1.111E+07)	1.258E+00(3.225E-06)
55	4.665E+09(9.477E+08)	5.051E+06(9.511E+06)	5.490E+06(5.254E+06)	1.261E+00(5.289E-06)
56	3.725E+09(1.258E+09)	1.273E+03(1.743E+03)	1.039E+05(5.726E+05)	1.445E+00(0.002E+00)
75	1.174E+07(7.458E+06)	2.012E+00(3.834E-04)	2.012E+00(2.719E-04)	2.0116E+00(5.296E+04)
76	4.436E+07(5.195E+07)	2.402E+07(2.580E+07)	1.772E+07(1.498E+07)	4.444E+05(2.328E+06)
77	3.846E+07(1.541E+07)	1.595E+07(2.488E+07)	2.150E+07(2.570E+07)	2.637E+00(1.796E-08)
78	4.109E+07(2.084E+07)	1.981E+07(2.163E+07)	1.975E+07(2.334E+07)	7.016E+04(3.841E+05)
79	3.891E+07(2.575E+07)	2.005E+07(2.579E+07)	2.262E+07(3.546E+07)	5.776E+05(3.002E+06)
80	3.533E+07(1.624E+07)	1.809E+07(2.496E+07)	2.185E+07(2.537E+07)	2.637E+00(1.356E-05)
-	0	0	0	N.A.
+	13	12	12	N.A.
~	0	1	1	N.A.

"Mean" and "Std" indicate the average and standard deviation of the function fitness values obtained in 31 runs, respectively.
"-","+", and "~" denote that the performance of the corresponding algorithm is better than, worsen than, and similar to that of ARCoDE, respectively.
Units are MVar if not specified.

Overall, the ARCoDE significantly outperforms the jDE, JADE, and SaDE in terms of the optimality and robustness of solutions. As the assumption described in Chapter 3, when dealing with practical ORPD kind of problems, these adaptive strategies may not guarantee to a satisfied control parameter setting within limited numbers of function evaluations. However, by relaxing the search criterion of control parameters from a specific value to a range, the ARCoDE presents a promising application for fast ORPD problems. The evolution of the mean fitness values achieved by the modified jDE, JADE, SaDE, and ARCoDE with respect to the number of function evaluations under four typical scenarios are depicted in Figure. 4.6-Figure. 4.9.

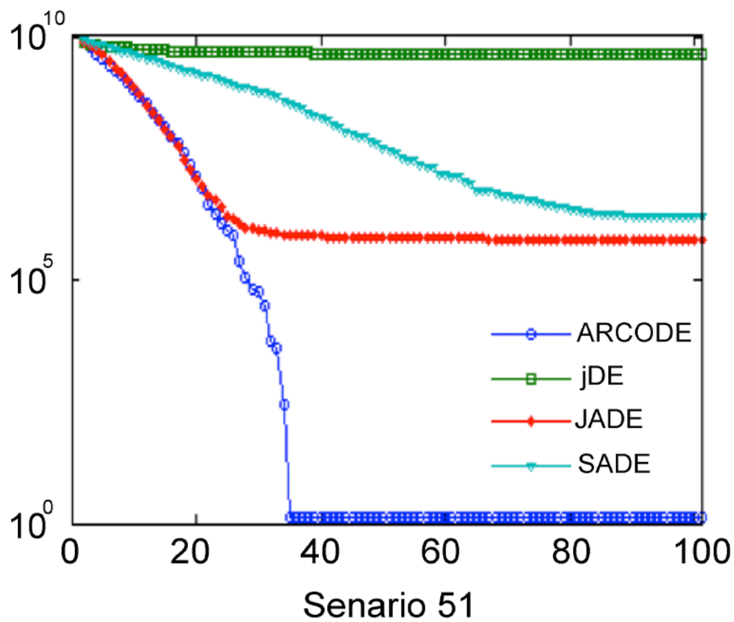

Figure 4.6. Mean fitness value evolution process of the jDE, JADE, SADE, and ARCoDE with respect to the number of function evaluations on Scenario 51.

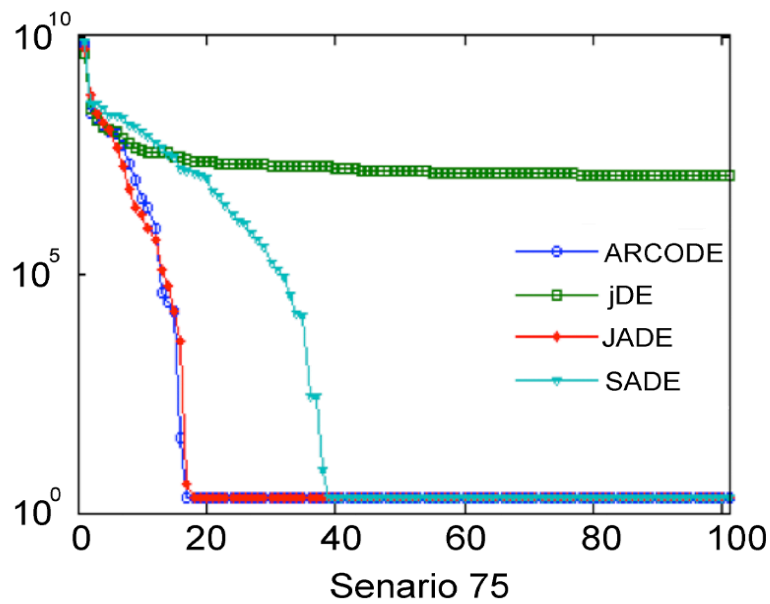

Figure 4.7. Mean fitness value evolution process of the jDE, JADE, SADE, and ARCoDE with respect to the number of function evaluations on Scenario 75.

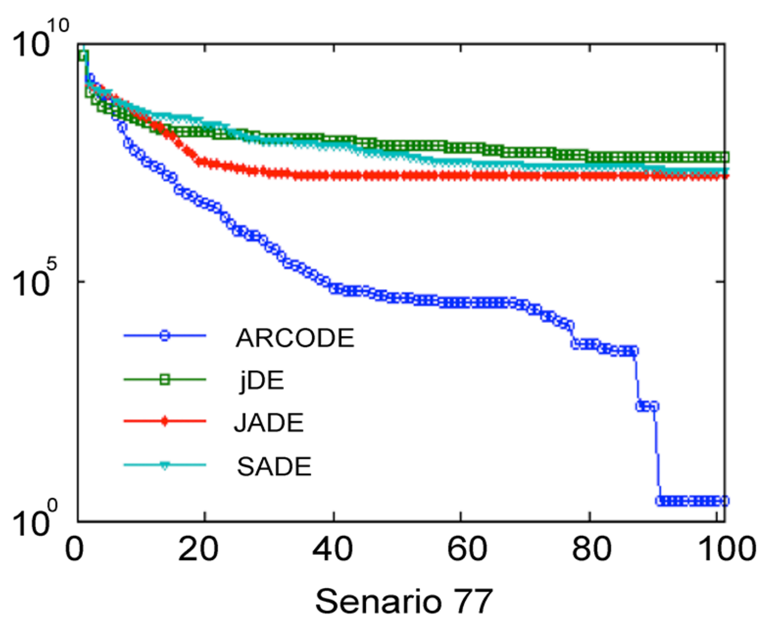

Figure 4.8. Mean fitness value evolution process of the jDE, JADE, SADE, and ARCoDE with respect to the number of function evaluations on Scenario 77.

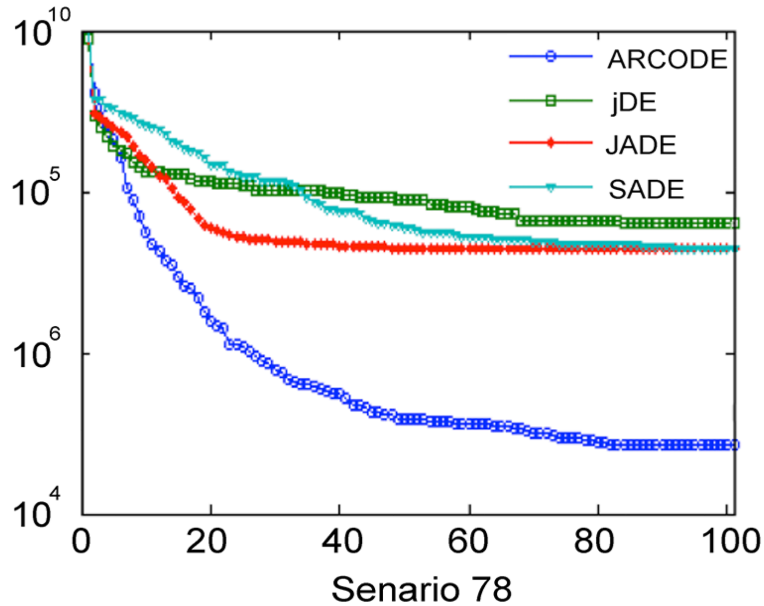

Figure 4.9. Mean fitness value evolution process of the jDE, JADE, SADE, and ARCoDE with respect to the number of function evaluations on Scenario 78.

4.3.4 Solution Feasibility Comparison in Terms of Feasible Rate

The ORPD problem contains a lot of equality and inequality constraints. Therefore, it is crucial for HOAs to provide stable feasible solutions, i.e., candidate solutions satisfying all the constraints of the ORPD problem. In terms of rate of feasible solutions, the ARCoDE is compared against 6 methods as shown in Table 4.5. For all the test scenarios, the feasible rates are obtained through 31 trials. It can be seen from Table 4.5 that the ARCoDE achieves 100% feasible rate for 10 of the 13 test scenarios. The feasible rate performance of the ARCoDE is remarkably higher than the other methods for most of the scenarios. It should be pointed out that the ARCoDE does not need a fine-tuned control parameter setting for trial vector generations. In addition, the constraint handling part of the ARCoDE is free of pre-defined parameter settings, which indeed enable the ease of its application in many practical problems.

In summary, the overall performance of the ARCoDE is highly competitive with the six rival methods. It is therefore convinced that the proposed ARCoDE can provide superior optimization performance for the ORPD problems.

Table 4.5. Comparison of ARCoDE with respect to ICDE, DEEPSO, MVMO,jDE, JADE, and SaDE in terms of feasible rate

Scenario	Feasible rate (%)						
	ICDE	DEEPSO	MVMO	jDE	JADE	SADE	ARCODE
50	25.8	100	100	0	64.5	77.4	100
51	6.5	100	100	0	74.2	58.1	100
52	0	100	100	0	77.4	0	100
53	0	100	100	0	83.9	0	100
54	0	100	100	0	67.7	0	100
55	0	100	100	0	77.4	0	100
56	32.3	100	100	0	64.5	67.7	100
75	93.5	96.8	100	100	100	100	100
76	0	87.1	100	0	29.0	0	93.5
77	0	93.5	100	0	54.8	0	100
78	0	96.8	96.8	0	38.7	0	90.3
79	0	96.8	100	0	35.5	0	87.1
80	0	100	100	0	35.5	0	100

Chapter 5 Efficiency Ranking-Based Evolutionary Algorithm for Multi-objective Reactive Power Dispatch

To properly trade-off between multiple inputs (e.g. operation cost of adjustments of generator voltage, transformer tap ratio and shunt capacitor) and multiple outputs (e.g. power loss and voltage deviation) in multi-objective ORPD, recently, multi-objective evolutionary algorithm (MOEA) was applied to produce a set of Pareto-optimal solutions [140]. In order to gauge the most efficient solutions (less in inputs and more in outputs) among the Pareto-optimal set, a post-processing using e.g. DEA is needed to select the optimal result in terms of efficiency, of which a quality Pareto-optimal front is essentially a prerequisite. However, the search ability of most MOEAs severely deteriorates when more than three objectives are involved, resulting in a poor POF [141].

In this chapter, an efficiency ranking-based evolutionary algorithm is proposed aiming at directly obtaining the most efficient decision-making units (DMUs). A DMU generically is regarded as the entity responsible for converting inputs into outputs and whose performances are to be evaluated. A slacks-based measure (SBM) of efficiency and its super efficiency pattern [142] are firstly applied to yield a full ranking of relative efficiency of DMUs in each EA generation, based on which the most efficient DMUs can be eventually found.

5.1 Basic Concept of Data Envelopment Analysis

Data envelopment analysis (DEA) is a linear programming procedure to measure the efficiency of DMUs by a scalar measure. Specifically, the Chames, Cooper and Rhodes (CCR) model handles the ratio of multiple inputs and outputs in an attempt to gauge the relative efficiency of the DMU concerned among all the observed ones. Let the DMU_j under evaluation be designated as DMU_o, the envelopment form of CCR is expressed with a real variable θ and a nonnegative vector $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_j, \dots, \lambda_n)$ of variables as follows:

$$\begin{aligned} &\textit{Minimize} && \theta \\ &\textit{subject to} && \theta \boldsymbol{x_o} \geq \boldsymbol{X\lambda},\ \ \boldsymbol{y_o} \leq \boldsymbol{Y\lambda}\ , \\ & && \lambda \geq 0 \end{aligned} \quad (5.1)$$

where the matrices $\boldsymbol{X} = (\boldsymbol{x_j})$ and $\boldsymbol{Y} = (\boldsymbol{y_j})$ are the input and output vectors, respectively. The input excesses $\boldsymbol{s^-} \in \mathbb{R}^m$ and the output shortfalls $\boldsymbol{s^+} \in \mathbb{R}^s$, identified as slack vectors, are defined as:

$$s^- = \theta \boldsymbol{x_o} - \boldsymbol{X\lambda},\ s^+ = \boldsymbol{Y\lambda} - \boldsymbol{y_o} \quad (5.2)$$

A DMU with the full ratio efficiency $\theta^* = 1$ and with no slacks ($\boldsymbol{s}^{-*} = 0, \boldsymbol{s}^{+*} = 0$) is called CCR-efficient; otherwise, the DMU is CCR-inefficient. For $\theta^* < 1$, the score actually reflects the depth of inefficiency of that DMU.

5.2 DEA Model Selection

The following properties of CCR model indicate its inadequacies to evaluate DMUs in power system planning and operation:

1) The CCR mode is built on the assumption of constant returns to scale. In other words, if (x, y) is a feasible point, then (tx, ty) for any positive t is also feasible.

2) In discussing total efficiency, it is integrant to observe both the ratio efficiency and the slacks.

3) The CCR model assumes that all the inputs are subject to change proportionally.

To surmount the above issues, a SBM of efficiency considering variable returns to scale of activities is firstly introduced as a fitness evaluation tool in EA to assign efficiency scores for power system DMUs, formulated as:

$$\begin{array}{ll} \textit{Minimize} & \rho = \left(1 - \frac{1}{m}\sum_{i=1}^{m} s_i^- / x_{io}\right) / \left(1 + \frac{1}{s}\sum_{r=1}^{s} s_r^- / y_{ro}\right) \\ \textit{subject to} & \boldsymbol{x_o} = \boldsymbol{X\lambda} + \boldsymbol{s}^-, \;\; \boldsymbol{y_o} = \boldsymbol{Y\lambda} - \boldsymbol{s}^+, \;\; \sum_{j=1}^{n} \lambda_j = 1 \;\;, \\ & \lambda \geq 0, \;\; \boldsymbol{s}^- \geq 0, \;\; \boldsymbol{s}^+ \geq 0 \end{array} \tag{5.3}$$

SBM is a non-radial type of efficiency measure, it is also invariant with respect to the unit of each input and output item. Furthermore, it gauges one unique indicator as the total efficiency for the observed DMUs, which would be convenient to guide the EAs' search process. The access of variable returns to scale assumption is provided by the adjunction of the condition $\sum_{j=1}^{n} \lambda_j = 1$.

The standard SBM does not differentiate the efficient DMUs and thus, cannot create a full ranking of efficiency. As such, we additionally introduce a slacks-based measure of super efficiency (SESBM) as follows:

$$\begin{array}{ll} \textit{Minimize} & \rho = \left(1 + \frac{1}{m}\sum_{i=1}^{m} \phi_i\right) / \left(1 - \frac{1}{s}\sum_{r=1}^{s} s_r^- / \psi_r\right) \\ \textit{subject to} & \sum_{j=1,\neq o}^{n} x_{ij}\lambda_j \leq x_{io}(1 + \phi_i), \ \ (i = 1,2,\cdots,m) \\ & \sum_{j=1,\neq o}^{n} y_{rj}\lambda_j \leq y_{ro}(1 - \psi_r), \quad (r = 1,2,\cdots,s) \\ & \sum_{j=1}^{n} \lambda_j = 1, \ \ \phi_i \geq 0, \ \ \psi_r \geq 0, \ \ \lambda_j \geq 0 \end{array} . \tag{5.4}$$

The basic idea is to compare the unit under evaluation with a linear combination of all other units in the sample. It is conceivable that an efficient DMU may increase its input vector while remaining efficient; in that case, the unit obtains an efficiency score above one [142].

5.3 Efficiency Ranking-Based Evolutionary Algorithm

DE algorithm is selected as the EA method in our study. Its step size and orientation of difference vectors (mutation operator) are automatically adaptive to the objective function landscape. It means that DE starts with a global search and changes automatically into a local search, which leads to a good balance between exploration and exploitation [140]. The proposed efficiency ranking-based evolutionary algorithm (EREA) is the integration of the efficiency-oriented competition mechanism and DE's reproduction and selection operators. In each generation the parent population and its corresponding offspring are combined together to construct a DMUs' input pool (size $2N$). The output part would be simulated planning or operation consequence accordingly. As the $2N$ DMUs are obtained, standard SBM is run first to classify efficient and inefficient DMUs and SESBM is then used to rank efficient DMUs. Higher efficiency scores indicate higher possibility of survival to the next generation. It should be highlighted that our approach directly evaluates the relative efficiency of DMUs through a two-stage linear programming (SBM/SESBM) instead of the two integrant but time-consuming procedures of most MOEAs (nondominated sorting and distance estimation) [141]. Therefore, the total computational complexity of EREA can be significantly reduced.

5.4 Case Studies

The proposed EREA algorithm is preliminarily tested in the IEEE 30-bus system to solve a three-objective ORPD problem. The detailed system topology, data, ORPD mathematical formulation and its constraints are given in [140]. As the proposed framework directly deals with the efficiency index, our objective is to minimize power loss and voltage deviation as well as control adjustments given current system operating point (Q_{C10} = 0, Q_{C24} = 0). In this case, a specific DMU is formulated as:

$$\boldsymbol{X} = [\Delta V_{\mathrm{G}}, \Delta T, \Delta Q_{\mathrm{C}}],\ \boldsymbol{Y} = [\Delta P_{\mathrm{L}}, \Delta V_{\mathrm{D}}], \tag{5.5}$$

where ΔV_{G}, ΔT and ΔQ_{C} represent the adjustments of generator voltages, transformer tap ratios and shunt capacitors, respectively, and ΔP_{L} and ΔV_{D} represent reduced power loss and voltage deviation due to control actions in $\boldsymbol{X}$, respectively. For demonstration, super-efficient DMUs consist of one input (ΔQ_{C}) and two outputs (ΔP_{L}, ΔV_{D}) are generated by applying the proposed EREA directly (black squares in Figure. 5.1), and by conducting post-assessment (SESBM) of the POF (red circles in Figure. 5.1) respectively. The mentioned POF (the blue diamond line in Figure. 5.1) is built as close to the genuine one as possible by firstly obtaining Pareto optimal solutions using SPEA2 and NSGA-II [141], respectively, and then performing nondominated sorting and crowding distance estimation to the combined solution set. Notably, none of the DMUs with the input (ΔQ_{C}) higher than 5MW is found to be super-efficient by both approaches employed, since the voltage deviation objective deteriorates with the increasing reactive power input as seen in Figure. 5.1. Those DMUs would never be super-efficient as we treat two output objectives without prejudices. The results also indicate that most Pareto calculations ($\Delta Q_{\mathrm{C}} > 5\ \mathrm{MW}$) are computationally heavy but unnecessary from practical decision-making perspective. The top 2 super-efficient decision makings of ORPD considering all inputs and outputs obtained by the EREA are also listed in Table 5.1.

Table 5.1. Test Results of The First and Second Decision Makings for ORPD Gauged by EREA

	V_{G1}	V_{G2}	V_{G5}	V_{G8}	V_{G11}	V_{G13}
Rank 1st	1.050	1.041	1.027	1.023	1.061	1.059
Rank 2nd	1.050	1.021	1.020	1.003	1.046	1.044
	T_{6-9}	T_{6-10}	T_{4-12}	T_{27-28}	$Q_{\mathrm{C}10}$ (MW)	$Q_{\mathrm{C}24}$ (MW)
Rank 1st	1.074	0.922	1.014	0.965	0.117	4.276
Rank 2nd	1.071	0.931	0.986	0.965	1.433	4.300

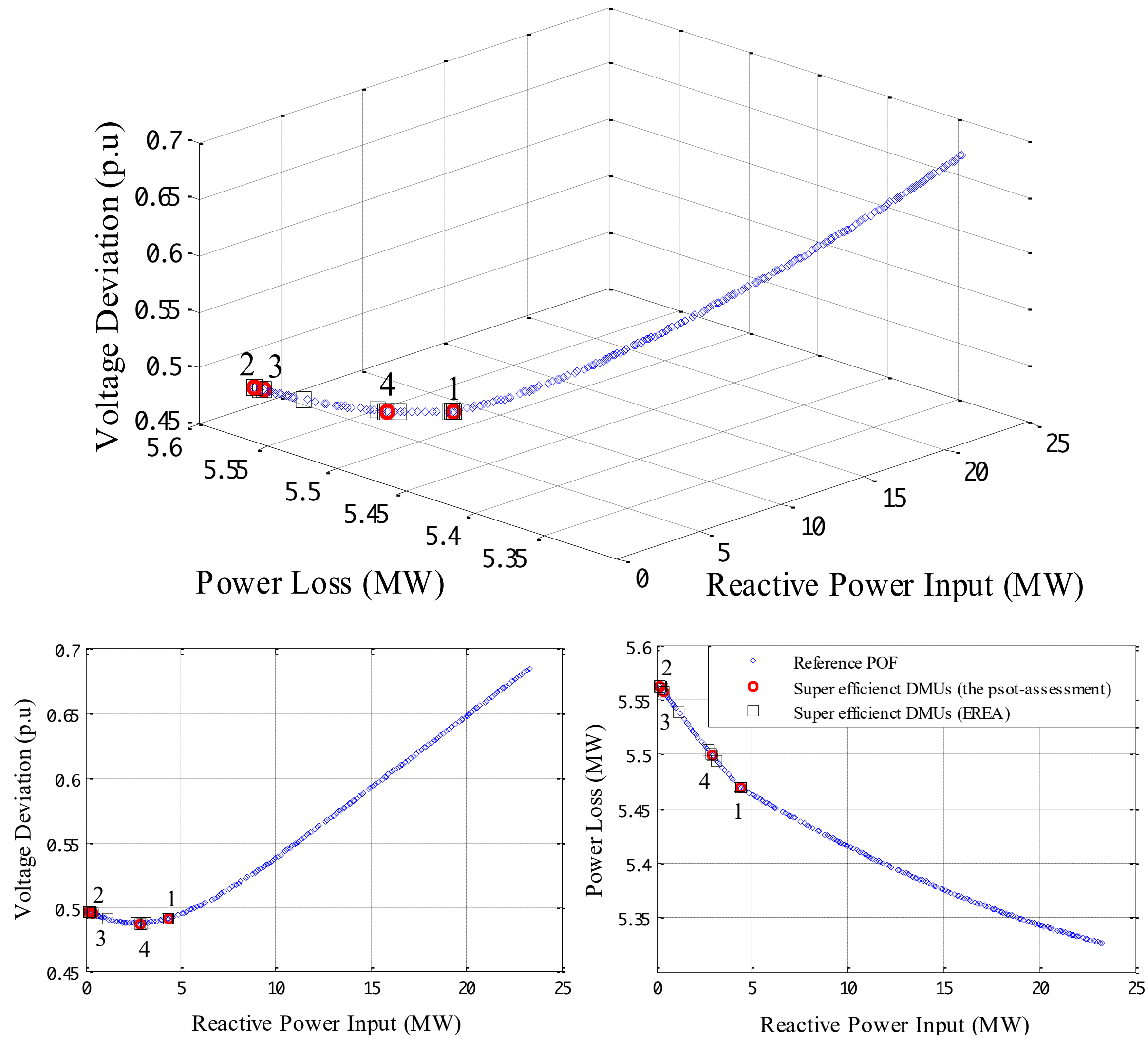

Figure 5.1. Comparison of the super-efficient DMUs for ORPD gauged by the post-assessment approach and EREA

(The number denotes the efficiency ranking generated by the post-assessment approach. The black squares are the super-efficient DMUs obtained by EREA through 100 independent trials, among which 47 of them find the super-efficient DMU in the adjacent domain of Rank 1st DMU while 36 trials converge to the adjacent domain of Rank 2nd DMU.)

Chapter 6 Conclusion

6.1 Summary of Thesis

Due the nonlinear and nonconvex nature involved in power system reactive power planning and dispatch problems, HOAs have been widely utilized to solve the related problems. In this thesis, an overview relevant research work applying HOAs for solving RPP and ORPD problem is systematically presented. It highlights the difficulties of deterministic methods facing i.e. non-continuity and non-differentiability of the objective functions. The reviewed articles are organized according to different categories of HOAs. A brief discussion of the limitations and trends of HOA applications with respect to RPP and ORPD problem is presented in addition to the corresponding literature reviews.

A novel MQDE algorithm is proposed for optimal reactive power planning. It explores the ability of MQDE to solve the constraint optimization in a continuous real-valued search space and emphasizes the advantages of applying the qubit representation to prevent a deficiency of population diversity in the later stages of the DE iteration process. In addition, the proposed reset index provides a more effective guidance about the qubits updating directions than the traditional parent-child selection operator of QDE. The power and usefulness of the proposed approach have been demonstrated by the successful optimization performance on various constraint optimization benchmark functions. In order to further demonstrate the performance of MQDE, it is implemented on a power system reactive power planning problem. Results also show an excellent performance of MQDE over typical DE in terms of satisfactory accuracy and significantly enhanced robustness.

A novel ARCoDE algorithm is proposed for optimal reactive power dispatch. It employs two trail vector generation strategies and a novel control parameter range adaptation strategy. The structure of ARCoDE is simpler and easier to implement. The experimental studies are carried out on the benchmark test cases of IEEE Competition on “Application of Modern Heuristic Optimization Algorithms for Solving Optimal Power Flow Problems”. The proposed ARCoDE is compared with the three award winning algorithms and three state-of-the-art adaptive DE algorithms. The experimental results demonstrate the overall performance of ARCoDE is better, or at least no worse than the other award-winning algorithms. In addition, the effectiveness of the combination of the selected trial vector generation strategies and the novel proposed adaptive range of control parameters are experimentally studied. The experimental results show that the fast convergence rate and the robustness of ARCoDE, which makes the proposed algorithm promising for solving fast ORPD problems. The developed algorithm also contributes to winning “Top Five Best Algorithm Award” in “Competition on Application of Modern Heuristic Optimization algorithms for Solving Optimal Power Flow Problems” organized by IEEE PES Working Group on Modern Heuristic Optimization Intelligent Systems Subcommittee & Power System Analysis, Computing, and Economic Committee in 2014.

Finally, EREA is proposed aiming at directly obtaining the most efficient DMUs for the multi-objective formulation of ORPD problem. A slacks-based measure (SBM) of efficiency and its super efficiency pattern are applied to yield a full ranking of relative efficiency of DMUs in each EA generation, based on which the most efficient DMUs can be eventually found.

6.2 Future Research Direction

The thesis proposes three advanced HOAs (i.e. MQDE, ARCoDE, and EREA) for solving optimal RPP and ORPD problems. Several aspects are provided worthy of further research in the future, including:

1) Even though MQDE has been successfully applied on cases such as the IEEE 30 bus system, comparison experiments need to be conducted with other system reactive power planning problems in future work.
2) In aspects of convergence speed and feasible rate, ARCoDE still can be enhanced. We will investigate the impacts of applying different candidate vector generation strategies.
3) In the future, we plan to combine the ARCoDE search engine and the efficiency ranking operator to solve more challenge power system planning and operation problems, i.e. co-planning of EV system and distribution system, micro-grid operation integrating PV power and energy storage resources.

Reference

[1] K. Y. Lee and F. F. Yang, "Optimal reactive power planning using evolutionary algorithms: A comparative study for evolutionary programming, evolutionary strategy, genetic algorithm, and linear programming," *IEEE Trans. Power Syst.*, vol. 13, no. 1, pp. 101–108, 1998.

[2] W. Zhang, F. Li, and L. M. Tolbert, "Review of reactive power planning: Objectives, constraints, and algorithms," *IEEE Trans. Power Syst.*, vol. 22, no. 4, pp. 2177–2186, 2007.

[3] C. Wan, Z. Xu, Y. Wang, Z. Y. Dong, and K. P. Wong, "A hybrid approach for probabilistic forecasting of electricity price," *IEEE Trans. Smart Grid*, vol. 5, no. 1, pp. 463–470, 2014.

[4] Y. Jiang, C. Wan, A. Botterud, Y. Song, and S. Xia, "Exploiting flexibility of district heating networks in combined heat and power dispatch," *IEEE Trans. Sustain. Energy*, pp. 1–1, 2019.

[5] Y. Jiang, C. Wan, J. Wang, Y. Song, and Z. Y. Dong, "Stochastic receding horizon control of active distribution networks with distributed renewables," *IEEE Trans. Power Syst.*, vol. 34, no. 2, pp. 1325–1341, 2019.

[6] S. S. Sachdeva and R. Billinton, "Optimum network Var planning by nonlinear programming," *IEEE Trans. Power Appar. Syst.*, vol. PAS-92, no. 4, pp. 1217–1225, 1973.

[7] G. T. Heydt and W. M. Grady, "Optimal Var siting using linear load flow formulation," *IEEE Trans. Power Appar. Syst.*, vol. 92, no. 5, pp. 1214–1222, 1983.

[8] K. Aoki, M. Fan, and A. Nishikori, "Optimal Var planning by approximation method for recursive mixed-integer linear programming," *IEEE Trans. Power Syst.*, vol. 3, no. 4, pp. 1741–1747, 1988.

[9] K. Y. Lee, Y. M. Park, and J. L. Ortiz, "A united approach to optimal real and reactive power dispatch," *IEEE Trans. Power Appar. Syst.*, vol. PAS-104, no. 5, pp. 1147–1153, 1985.

[10] K. Y. Lee, J. L. Ortiz, Y. M. Park, and L. G. Pond, "An optimization technique for reactive power planning of subtransmission network under normal operation,"

IEEE Trans. Power Syst., vol. 1, no. 2, pp. 153–159, 1986.

[11] M. K. Mangoli, K. Y. Lee, and Y. Moon Park, "Optimal real and reactive power control using linear programming," *Electr. Power Syst. Res.*, vol. 26, no. 1, pp. 1–10, 1993.

[12] M. K. Mangoli, K. Y. Lee, and Y. M. Park, "Optimal long-term reactive power planning using decomposition techniques," *Electr. Power Syst. Res.*, vol. 26, no. 1, pp. 41–52, 1993.

[13] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st ed. Boston, MA: Addison-Wesley Longman Publishing Co., Inc., 1989.

[14] D. B. Fogel, "An introduction to simulated evolutionary optimization," *IEEE Trans. Neural Networks*, vol. 5, no. 1, pp. 3–14, 1994.

[15] K. Y. Lee, X. Bai, and Y.-M. Park, "Optimization method for reactive power planning by using a modified simple genetic algorithm," *IEEE Trans. Power Syst.*, vol. 10, no. 4, pp. 1843–1850, 1995.

[16] L. L. Lai and J. T. Ma, "Application of evolutionary programming to reactive power planning-comparison with nonlinear programming approach," *IEEE Trans. Power Syst.*, vol. 12, no. 1, pp. 198–206, 1997.

[17] J. H. Park, S.-O. Yang, H.-S. Lee, and Y. M. Park, "Economic load dispatch using evolutionary algorithms," in *Proceedings of International Conference on Intelligent System Application to Power Systems*, 1996, pp. 441–445.

[18] R. Dimeo and K. Y. Lee, "Boiler-turbine control system design using a genetic algorithm," *IEEE Trans. Energy Convers.*, vol. 10, no. 4, pp. 752–759, 1995.

[19] V. Miranda, J. V Ranito, and L. M. Proenca, "Genetic algorithms in optimal multistage distribution network planning," *IEEE Trans. Power Syst.*, vol. 9, no. 4. pp. 1927–1933, 1994.

[20] K. Iba, "Reactive power optimization by genetic algorithm," *IEEE Trans. Power Syst.*, vol. 9, no. 2, pp. 685–692, 1994.

[21] C.-H. Wu, J. J.-F. Tsai, S. S.-M. Lo, T. T.-Y. Tsai, C.-C. Lin, Y.-H. Hung, and P.-Y. Chen, "System dynamics analysis and evaluation of state of-charge for lithium batteries," *Environ. Sci. Inf. Appl. Technol. (ESIAT), 2010 Int. Conf.*, vol. 4, pp. 391–394, 2010.

[22] N. Deeb and S. M. Shahidehpour, "Linear reactive power optimization in a large power network using the decomposition approach," *IEEE Trans. Power Syst.*, vol.

5, no. 2, pp. 428–438, 1990.

[23] K. C. Almeida and F. S. Senna, “Optimal active-reactive power dispatch under competition via bilevel programming,” *IEEE Trans. Power Syst.*, vol. 26, no. 4, pp. 2345–2354, 2011.

[24] M. Niu, C. Wan, and Z. Xu, “A review on applications of heuristic optimization algorithms for optimal power flow in modern power systems,” *J. Mod. Power Syst. Clean Energy*, vol. 2, no. 4, pp. 289–297, Dec. 2014.

[25] 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, 2002.

[26] P. Subbaraj and P. N. Rajnarayanan, “Optimal reactive power dispatch using self-adaptive real coded genetic algorithm,” *Electr. Power Syst. Res.*, vol. 79, no. 2, pp. 374–381, 2009.

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

[28] M. R. AlRashidi and M. E. El-Hawary, “Hybrid particle swarm optimization approach for solving the discrete OPF problem considering the valve loading effects,” *IEEE Trans. Power Syst.*, vol. 22, no. 4, pp. 2030–2038, 2007.

[29] B. Zhao, C. X. Guo, and Y. J. Cao, “A multiagent-based particle swarm optimization approach for optimal reactive power dispatch,” *IEEE Trans. Power Syst.*, vol. 20, no. 2, pp. 1070–1078, 2005.

[30] M. Varadarajan and K. S. Swarup, “Solving multi-objective optimal power flow using differential evolution,” *IET Gener. Transm. Distrib.*, vol. 2, no. 5, pp. 720–730, 2008.

[31] C. Dai, W. Chen, Y. Zhu, and X. Zhang, “Seeker optimization algorithm for optimal reactive power dispatch,” *IEEE Trans. Power Syst.*, vol. 24, no. 3, pp. 1218–1231, 2009.

[32] H. V Pham, J. L. Rueda, and I. Erlich, “Online optimal control of reactive sources in wind power plants,” *IEEE Trans. Sustain. Energy*, vol. 5, no. 2, pp. 608–616, 2014.

[33] J. G. Vlachogiannis and K. Y. Lee, “Quantum-inspired evolutionary algorithm for real and reactive power dispatch,” *IEEE Trans. Power Syst.*, vol. 23, no. 4, pp. 1627–1636, 2008.

[34] M. J. Dolan, E. M. Davidson, I. Kockar, G. W. Ault, and S. D. J. McArthur,

"Distribution power flow management utilizing an online optimal power flow technique," *IEEE Trans. Power Syst.*, vol. 27, no. 2, pp. 790–799, 2012.

[35] Y. Wang, Z. Cai, and Q. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 55–66, 2011.

[36] C. Wan, J. Lin, W. Guo, and Y. Song, "Maximum uncertainty boundary of volatile distributed generation in active distribution network," *IEEE Trans. Smart Grid*, vol. 9, no. 4, pp. 2930–2942, 2018.

[37] C. Wan, Z. Xu, and P. Pinson, "Direct interval forecasting of wind power," *IEEE Trans. Power Syst.*, vol. 28, no. 4, pp. 4877–4878, 2013.

[38] C. Wan, Z. Xu, P. Pinson, Z. Y. Dong, and K. P. Wong, "Probabilistic forecasting of wind power generation using extreme learning machine," *IEEE Trans. Power Syst.*, vol. 29, no. 3, pp. 1033–1044, 2014.

[39] Z. Cao, C. Wan, Z. Zhang, F. Li, and Y. Song, "Hybrid ensemble deep learning for deterministic and probabilistic low-voltage load forecasting," *IEEE Trans. Power Syst.*, vol. 35, no. 3, pp. 1881–1897, 2020.

[40] S. S. Rao, *Engineering Optimization: Theory and Practice: Fourth Edition*. 2009.

[41] P. Hudson, "Optimization: Theory and applications," *J. Oper. Res. Soc.*, vol. 30, no. 5, pp. 498–499, May 1979.

[42] A. W. Flux and V. Pareto, "Cours d'economie politique," *Econ. J.*, vol. 7, no. 25, p. 91, Mar. 1897.

[43] C. Wan, M. Niu, Y. Song, and Z. Xu, "Pareto optimal prediction intervals of electricity price," *IEEE Trans. Power Syst.*, vol. 32, no. 1, pp. 817–819, 2017.

[44] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. Cambridge, MA: MIT Press, 1992.

[45] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, 1995, pp. 1942–1948.

[46] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the International Symposium on Micro Machine and Human Science*, 1995.

[47] R. Storn and K. Price, "Differential evolution - A simple and efficient adaptive scheme for global optimization over continuous spaces," 1995.

[48] L. Lakshminarasimman and S. Subramanian, “Short-term scheduling of hydrothermal power system with cascaded reservoirs by using modified differential evolution,” *IEE Proc. - Gener. Transm. Distrib.*, vol. 153, no. 6, pp. 693–700, 2006.

[49] Z. Wang, C. Y. Chung, K. P. Wong, and C. T. Tse, “Robust power system stabiliser design under multi-operating conditions using differential evolution,” *IET Gener. Transm. Distrib.*, vol. 2, no. 5, pp. 690–700, 2008.

[50] V. S. Vakula and K. R. Sudha, “Design of differential evolution algorithm-based robust fuzzy logic power system stabiliser using minimum rule base,” *IET Gener. Transm. Distrib.*, vol. 6, no. 2, pp. 121–132, 2012.

[51] L. J. Fogel and G. H. Burgin, “Competitive goal-seeking through evolutionary programming,” Dayton, OH, 1969.

[52] T. Back, D. B. Fogel, and Z. Michalewicz, Eds., *Handbook of Evolutionary Computation*, 1st ed. Bristol, UK, UK: IOP Publishing Ltd., 1997.

[53] L. L. Lai, J. T. Ma, R. Yokoyama, and M. Zhao, “Improved genetic algorithms for optimal power flow under both normal and contingent operation states,” *Int. J. Electr. Power Energy Syst.*, vol. 19, no. 5, pp. 287–292, 1997.

[54] T. Numnonda and U. D. Annakkage, “Optimal power dispatch in multinode electricity market using genetic algorithm,” *Electr. Power Syst. Res.*, vol. 49, no. 3, pp. 211–220, 1999.

[55] M. S. Kumari and S. Maheswarapu, “Enhanced genetic algorithm based computation technique for multi-objective optimal power flow solution,” *Int. J. Electr. Power Energy Syst.*, vol. 32, no. 6, pp. 736–742, 2010.

[56] Q. H. Wu, Y. J. Cao, and J. Y. Wen, “Optimal reactive power dispatch using an adaptive genetic algorithm,” *Int. J. Electr. Power Energy Syst.*, vol. 20, no. 8, pp. 563–569, 1998.

[57] S. R. Paranjothi and K. Anburaja, “Optimal power flow using refined genetic algorithm,” *Electr. Power Components Syst.*, vol. 30, no. 10, pp. 1055–1063, Oct. 2002.

[58] D. Devaraj and B. Yegnanarayana, “Genetic-algorithm-based optimal power flow for security enhancement,” *IEE Proc. - Gener. Transm. Distrib.*, vol. 152, no. 6, pp. 899–905, 2005.

[59] Z.-L. Gaing and R.-F. Chang, “Security-constrained optimal power flow by mixed-integer genetic algorithm with arithmetic operators,” in *2006 IEEE Power Engineering Society General Meeting*, 2006, p. 8 pp.

[60] W. Yan, F. Liu, C. Y. Chung, and K. P. Wong, "A hybrid genetic algorithm-interior point method for optimal reactive power flow," *IEEE Trans. Power Syst.*, vol. 21, no. 3, pp. 1163–1169, 2006.

[61] C. M. Wankhade and A. P. Vaidya, "Optimal power flow using genetic algorithm: Parametric studies for selection of control and state variables," *Br. J. Appl. Sci. Technol.*, vol. 4, no. 2, pp. 279–301, 2014.

[62] B. Mahdad, K. Srairi, and T. Bouktir, "Optimal power flow for large-scale power system with shunt FACTS using efficient parallel GA," *Int. J. Electr. Power Energy Syst.*, vol. 32, no. 5, pp. 507–517, 2010.

[63] B. Bhattacharyya and S. Raj, "A novel approach for the voltage stability assessment and reactive power planning," in *2015 IEEE 10th Conference on Industrial Electronics and Applications (ICIEA)*, 2015, pp. 1534–1538.

[64] Y. Liu, M. Liu, and G. Gao, "Optimal reactive power planning using GA/SA/TS hybrid approach and decomposition and coordination theory," in *2008 IEEE International Conference on Industrial Technology*, 2008, pp. 1–4.

[65] A. A. Abou El Ela, M. A. Abido, and S. R. Spea, "Optimal power flow using differential evolution algorithm," *Electr. Power Syst. Res.*, vol. 80, no. 7, pp. 878–885, 2010.

[66] J. Hazra and A. K. Sinha, "A multi-objective optimal power flow using particle swarm optimization," *Eur. Trans. Electr. Power*, vol. 21, no. 1, pp. 1028–1045, Jan. 2011.

[67] R.-H. Liang, S.-R. Tsai, Y.-T. Chen, and W.-T. Tseng, "Optimal power flow by a fuzzy based hybrid particle swarm optimization approach," *Electr. Power Syst. Res.*, vol. 81, no. 7, pp. 1466–1474, 2011.

[68] K. Mahadevan and P. S. Kannan, "Comprehensive learning particle swarm optimization for reactive power dispatch," *Appl. Soft Comput.*, vol. 10, no. 2, pp. 641–652, 2010.

[69] M. A. Abido, "Multiobjective particle swarm optimization for optimal power flow problem," in *2008 12th International Middle-East Power System Conference*, 2008, pp. 392–396.

[70] N. Mo, Z. Y. Zou, K. W. Chan, and T. Y. G. Pong, "Transient stability constrained optimal power flow using particle swarm optimisation," *IET Gener. Transm. Distrib.*, vol. 1, no. 3, p. 476, May 2007.

[71] P. E. O. Yumbla, J. M. Ramirez, and C. A. C. Coello, "Optimal power flow subject to security constraints solved with a particle swarm optimizer," *IEEE Trans.*

Power Syst., vol. 23, no. 1, pp. 33–40, 2008.

[72] T. Niknam, M. R. Narimani, J. Aghaei, and R. Azizipanah-Abarghooee, “Improved particle swarm optimisation for multi-objective optimal power flow considering the cost, loss, emission and voltage stability index,” *IET Gener. Transm. Distrib.*, vol. 6, no. 6, pp. 515–527, 2012.

[73] Q. Kang, M. Zhou, J. An, and Q. Wu, “Swarm intelligence approaches to optimal power flow problem with distributed generator failures in power networks,” *IEEE Trans. Autom. Sci. Eng.*, vol. 10, no. 2, pp. 343–353, 2013.

[74] W. Zhang and Y. Liu, “Reactive power optimization based on PSO in a practical power system,” in *IEEE Power Engineering Society General Meeting, 2004.*, 2004, pp. 239-243 Vol.1.

[75] A. Nayan, “Reactive power planning using PSO with modified dynamic inertia parameter,” in *2014 Power and Energy Systems: Towards Sustainable Energy*, 2014, pp. 1–6.

[76] S. Sivasubramani and K. S. Swarup, “Multiagent based differential evolution approach to optimal power flow,” *Appl. Soft Comput.*, vol. 12, no. 2, pp. 735–740, 2012.

[77] N. Amjady and H. Sharifzadeh, “Security constrained optimal power flow considering detailed generator model by a new robust differential evolution algorithm,” *Electr. Power Syst. Res.*, vol. 81, no. 2, pp. 740–749, 2011.

[78] S. Sayah and K. Zehar, “Modified differential evolution algorithm for optimal power flow with non-smooth cost functions,” *Energy Convers. Manag.*, vol. 49, no. 11, pp. 3036–3042, 2008.

[79] 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 Gener. Transm. Distrib.*, vol. 1, no. 2, pp. 253–260, 2007.

[80] C. H. Liang, C. Y. Chung, 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, 2007.

[81] M. Basu, “Optimal power flow with FACTS devices using differential evolution,” *Int. J. Electr. Power Energy Syst.*, vol. 30, no. 2, pp. 150–156, 2008.

[82] 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, 2008.

[83] S. Sivasubramani and K. S. Swarup, “Sequential quadratic programming based differential evolution algorithm for optimal power flow problem,” *IET Gener. Transm. Distrib.*, vol. 5, no. 11, pp. 1149–1154, 2011.

[84] K. R. Vadivelu and G. V Marutheswar, “Artificial intelligence technique based reactive power planning using FVSI,” in *2013 International Conference on Advanced Computing and Communication Systems*, 2013, pp. 1–6.

[85] K. P. Wong and J. Yuryevich, “Optimal power flow method using evolutionary programming,” in *Simulated Evolution and Learning*, Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 405–412.

[86] W. Yan, S. Lu, and D. C. Yu, “A novel optimal reactive power dispatch method based on an improved hybrid evolutionary programming technique,” *IEEE Trans. Power Syst.*, vol. 19, no. 2, pp. 913–918, 2004.

[87] W. Ongsakul and T. Tantimaporn, “Optimal power flow by improved evolutionary programming,” *Electr. Power Components Syst.*, vol. 34, no. 1, pp. 79–95, Jan. 2006.

[88] P. Somasundaram, K. Kuppusamy, and R. P. Kumudini Devi, “Evolutionary programming based security constrained optimal power flow,” *Electr. Power Syst. Res.*, vol. 72, no. 2, pp. 137–145, 2004.

[89] Y. R. Sood, “Evolutionary programming based optimal power flow and its validation for deregulated power system analysis,” *Int. J. Electr. Power Energy Syst.*, vol. 29, no. 1, pp. 65–75, 2007.

[90] R. Gnanadass, P. Venkatesh, and N. P. Padhy, “Evolutionary Programming Based Optimal Power Flow for Units with Non-Smooth Fuel Cost Functions,” *Electr. Power Components Syst.*, vol. 33, no. 3, pp. 349–361, Dec. 2004.

[91] S. K. N. Kumar and P. Renuga, “FVSI based Reactive Power Planning using Evolutionary Programming,” in *2010 International Conference on Communication Control and Computing Technologies*, 2010, pp. 265–269.

[92] K. R. Vadivelu and G. V Marutheswar, “Artificial intelligence technique based Reactive Power Planning incorporating FACTS Controllers in Real Time Power Transmission System,” in *2014 IEEE 2nd International Conference on Electrical Energy Systems (ICEES)*, 2014, pp. 26–31.

[93] V. J. Gutierrez-Martinez, C. A. Canizares, C. R. Fuerte-Esquivel, A. Pizano-Martinez, and X. Gu, “Neural-network security-boundary constrained optimal power flow,” *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 63–72, 2011.

[94] B. Venkatesh, “Online ANN memory model–based method for unified OPF and

voltage stability margin maximization," *Electr. Power Components Syst.*, vol. 31, no. 5, pp. 453–465, May 2003.

[95] A. Abbasy and S. H. Hosseini, "Ant colony optimization-based approach to optimal reactive power dispatch: A comparison of various ant systems," in *2007 IEEE Power Engineering Society Conference and Exposition in Africa - PowerAfrica*, 2007, pp. 1–8.

[96] J. G. Vlachogiannis, N. D. Hatziargyriou, and K. Y. Lee, "Ant colony system-based algorithm for constrained load flow problem," *IEEE Trans. Power Syst.*, vol. 20, no. 3, pp. 1241–1249, 2005.

[97] C. A. Roa-Sepulveda and B. J. Pavez-Lazo, "A solution to the optimal power flow using simulated annealing," in *2001 IEEE Porto Power Tech Proceedings (Cat. No.01EX502)*, 2001, p. 5 pp. vol.2.

[98] P. K. Roy, S. P. Ghoshal, and S. S. Thakur, "Biogeography based optimization for multi-constraint optimal power flow with emission and non-smooth cost function," *Expert Syst. Appl.*, vol. 37, no. 12, pp. 8221–8228, 2010.

[99] K. Ayan and U. Kılıç, "Artificial bee colony algorithm solution for optimal reactive power flow," *Appl. Soft Comput.*, vol. 12, no. 5, pp. 1477–1482, 2012.

[100] W. Ongsakul and P. Bhasaputra, "Optimal power flow with FACTS devices by hybrid TS/SA approach," *Int. J. Electr. Power Energy Syst.*, vol. 24, no. 10, pp. 851–857, 2002.

[101] S. Kumar and D. K. Chaturvedi, "Optimal power flow solution using fuzzy evolutionary and swarm optimization," *Int. J. Electr. Power Energy Syst.*, vol. 47, pp. 416–423, 2013.

[102] C. Y. Chung, C. H. Liang, K. P. Wong, and X. Z. Duan, "Hybrid algorithm of differential evolution and evolutionary programming for optimal reactive power flow," *IET Gener. Transm. Distrib.*, vol. 4, no. 1, pp. 84–93, 2010.

[103] Y. Li, Y. Wang, and B. Li, "A hybrid artificial bee colony assisted differential evolution algorithm for optimal reactive power flow," *Int. J. Electr. Power Energy Syst.*, vol. 52, pp. 25–33, 2013.

[104] K. Vaisakh and L. R. Srinivas, "Evolving ant direction differential evolution for OPF with non-smooth cost functions," *Eng. Appl. Artif. Intell.*, vol. 24, no. 3, pp. 426–436, 2011.

[105] J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer, "Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems," *IEEE Trans. Evol. Comput.*, vol. 10, no. 6, pp. 646–657,

2006.

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

[107] J. Zhang and A. C. Sanderson, “JADE: Adaptive differential evolution with optional external archive,” *IEEE Trans. Evol. Comput.*, vol. 13, no. 5, pp. 945–958, 2009.

[108] P. Kaelo and M. M. Ali, “A numerical study of some modified differential evolution algorithms,” *Eur. J. Oper. Res.*, vol. 169, no. 3, pp. 1176–1184, 2006.

[109] P. W. Shor, “Quantum computing,” *Doc. Math.*, vol. Extra Vol., pp. 467–486, 1998.

[110] A. Narayanan and M. Moore, “Quantum-inspired genetic algorithms,” in *Proceedings of IEEE International Conference on Evolutionary Computation*, 1996, pp. 61–66.

[111] K.-H. Han and J.-H. Kim, “Quantum-inspired evolutionary algorithm for a class of combinatorial optimization,” *IEEE Trans. Evol. Comput.*, vol. 6, no. 6, pp. 580–593, 2002.

[112] K. Meng, H. G. Wang, Z. 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, 2010.

[113] H. Su and Y. Yang, “Quantum-inspired differential evolution for binary optimization,” in *2008 Fourth International Conference on Natural Computation*, 2008, pp. 341–346.

[114] T. Zheng, “Quantum-inspired Differential Evolutionary Algorithm for Permutative Scheduling Problems,” in *Evolutionary Algorithms*, M. Y. E.-E. Kita, Ed. Rijeka: IntechOpen, 2011, p. Ch. 7.

[115] A. Draa, S. Meshoul, H. Talbi, and M. Batouche, “A quantum inspired differential evolution algorithm for rigid image registration,” in *in Proceedings of the International Conference on Computational Intelligence, Istanbul*, 2004, pp. 408–411.

[116] A. R. Hota and A. Pat, “An adaptive quantum-inspired differential evolution algorithm for 0–1 knapsack problem,” in *2010 Second World Congress on Nature and Biologically Inspired Computing (NaBIC)*, 2010, pp. 703–708.

[117] H. Su, Y. Yang, and L. Zhao, “Classification rule discovery with DE/QDE

algorithm," *Expert Syst. Appl.*, vol. 37, no. 2, pp. 1216–1222, 2010.

[118] H.-Y. Fan and J. Lampinen, "A trigonometric mutation operation to differential evolution," *J. Glob. Optim.*, vol. 27, no. 1, pp. 105–129, Sep. 2003.

[119] E. Mezura-Montes, J. Velazquez-Reyes, and C. A. C. Coello, "Modified differential evolution for constrained optimization," in *2006 IEEE International Conference on Evolutionary Computation*, 2006, pp. 25–32.

[120] V. Feoktistov and S. Janaqi, "Generalization of the strategies in differential evolution," in *18th International Parallel and Distributed Processing Symposium, 2004. Proceedings.*, 2004, p. 165.

[121] S. Das, A. Konar, and U. K. Chakraborty, "Two improved differential evolution schemes for faster global search," in *Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation*, New York, NY, USA: ACM, 2005, pp. 991–998.

[122] R. Gämperle, S. D. Müller, and P. Koumoutsakos, "A parameter study for differential evolution," in *WSEAS Int. Conf. on Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation*, Press, 2002, pp. 293–298.

[123] A. A. Cuello-reyna and J. R. Cedeno-maldonado, "A differential evolution approach to optimal reactive power planning," in *2006 IEEE/PES Transmission & Distribution Conference and Exposition: Latin America*, 2006, pp. 1–7.

[124] J. Z. Zhu, C. S. Chang, W. Yan, and G. Y. Xu, "Reactive power optimisation using an analytic hierarchical process and a nonlinear optimisation neural network approach," *IEE Proc. - Gener. Transm. Distrib.*, vol. 145, no. 1, pp. 89–97, 1998.

[125] B. Kermanshahi, K. Takahashi, and Y. Zhou, "Optimal operation and allocation of reactive power resource considering static voltage stability," in *POWERCON '98. 1998 International Conference on Power System Technology. Proceedings (Cat. No.98EX151)*, 1998, pp. 1473–1477 vol.2.

[126] 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, 2008.

[127] J. J. Liang, T. P. Runarsson, E. Mezura-Montes, M. Clerc, P. N. Suganthan, C. A. C. Coello, and K. Deb, "Problem definitions and evaluation criteria for the CEC 2006 special session on constrained real-parameter optimization," *J. Appl. Mech.*, vol. 41, no. 8, pp. 8–31, 2006.

[128] K. Deb, "An efficient constraint handling method for genetic algorithms," *Comput. Methods Appl. Mech. Eng.*, vol. 186, no. 2, pp. 311–338, 2000.

[129] F. Herrera, M. Lozano, and J. L. Verdegay, “Tackling real-coded genetic algorithms: operators and tools for behavioural analysis,” *Artif. Intell. Rev.*, vol. 12, no. 4, pp. 265–319, Aug. 1998.

[130] K. N. Miu, H.-. Chiang, and G. Darling, “Capacitor placement, replacement and control in large-scale distribution systems by a GA-based two-stage algorithm,” *IEEE Trans. Power Syst.*, vol. 12, no. 3, pp. 1160–1166, 1997.

[131] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, “MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education,” *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 12–19, 2011.

[132] C. Wan, Z. Xu, P. Pinson, Z. Y. Dong, and K. P. Wong, “Optimal prediction intervals of wind power generation,” *IEEE Trans. Power Syst.*, vol. 29, no. 3, pp. 1166–1174, 2014.

[133] C. Wan, J. Lin, J. Wang, Y. Song, and Z. Y. Dong, “Direct quantile regression for nonparametric probabilistic forecasting of wind power generation,” *IEEE Trans. Power Syst.*, vol. 32, no. 4, pp. 2767–2778, 2017.

[134] C. Wan, J. Wang, J. Lin, Y. Song, and Z. Y. Dong, “Nonparametric prediction intervals of wind power via linear programming,” *IEEE Trans. Power Syst.*, vol. 33, no. 1, pp. 1074–1076, 2018.

[135] C. Wan, C. Zhao, and Y. Song, “Chance constrained extreme learning machine for nonparametric prediction intervals of wind power generation,” *IEEE Trans. Power Syst.*, pp. 1–1, 2020.

[136] C. Zhao, C. Wan, and Y. Song, “An adaptive bilevel programming model for nonparametric prediction intervals of wind power generation,” *IEEE Trans. Power Syst.*, vol. 35, no. 1, pp. 424–439, 2020.

[137] “IEEE PES working group on modern heuristic optimization.” .

[138] G. Jia, Y. Wang, Z. Cai, and Y. Jin, “An improved (μ+λ)-constrained differential evolution for constrained optimization,” *Inf. Sci. (Ny).*, vol. 222, pp. 302–322, 2013.

[139] R. P. Q. Alves, “Stochastic location of FACTS devices in electric power transmission networks,” Faculdade de Engenharia da Universidade do Porto, 2013.

[140] M. A. Abido and J. M. Bakhashwain, “Optimal VAR dispatch using a multiobjective evolutionary algorithm,” *Int. J. Electr. Power Energy Syst.*, vol. 27, no. 1, pp. 13–20, 2005.

[141] D. K. Saxena, J. A. Duro, A. Tiwari, K. Deb, and Q. Zhang, “Objective reduction

in many-objective optimization: Linear and nonlinear algorithms," *IEEE Trans. Evol. Comput.*, vol. 17, no. 1, pp. 77–99, 2013.

[142] W. W. Cooper, K. Tone, and L. M. Seiford, *Data Envelopment Analysis: A Comprehensive Text with Models, Applications References, and DEA-Solver Software with Cdrom*. Norwell, MA, USA: Kluwer Academic Publishers, 1999.