Heterogeneous particle swarm optimization with an application of unit commitment in power system

Nandar Lynn

2016

Nandar Lynn. (2016). Heterogeneous particle swarm optimization with an application of unit commitment in power system. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/69219>

<https://doi.org/10.32657/10356/69219>

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

**HETEROGENEOUS PARTICLE SWARM
OPTIMIZATION WITH AN APPLICATION OF
UNIT COMMITMENT IN POWER SYSTEM**

NANDAR LYNN

SCHOOL OF ELECTRICAL AND ELECTRONIC ENGINEERING

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

2016

**HETEROGENEOUS PARTICLE SWARM
OPTIMIZATION WITH AN APPLICATION OF
UNIT COMMITMENT IN POWER SYSTEM**

NANDAR LYNN

School of Electrical and Electronic Engineering

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

2016

Acknowledgements

Foremost, I would like to express my sincere gratitude to my supervisor, Associate Professor Ponnuthurai Nagaratnam Suganthan for his motivation, patience, guidance, advice, encouragement and immense knowledge. He is extraordinary person and it is a great honor that I have the privilege to work with him. Without his constant advice and encouragement, I may not accomplish my postgraduate study smoothly.

I also want to express my thanks to Dr. Narasimalu Srikanth from ERI@N and Dr. Rammohan Mallipeddi from Kyungpook National University, for their valuable advices and suggestions for my research work.

I would like to thank all my good friends and lab-mates including for their support during the past four years.

Finally, thanks with lots of love to my parents and my husband for their encouragement and support to complete the Ph.D study.

Table of Contents

Acknowledgements	i
Table of Contents.....	ii
Summary.....	v
List of Tables.....	vii
List of Figures.....	x
List of Symbols.....	xiv
List of Appendices	xvi
1 Introduction	1
1.1 Balancing Exploration and Exploitation.....	1
1.2 Ensemble of Optimization Techniques	2
1.3 Optimization problems in Power Industry	3
1.4 Major Contributions of the Thesis	3
1.5 Organization of the Thesis	5
2 Literature Review	7
2.1 Particle Swarm Optimization.....	8
2.1.1 Prominent PSO Variants	9
2.1.2 Applications.....	16
2.2 Unit Commitment Problem.....	17
2.2.1 Unit Commitment Problem Formulation.....	18
2.2.2 Optimization Methodologies	20
3 Heterogeneous Comprehensive Learning PSO.....	23
3.1 Introduction.....	23
3.2 Comprehensive Learning (CL) Strategy	25
3.3 Heterogeneous CLPSO with Enhanced Exploration and Exploitation.....	26

3.4	Experimental Set Up and Results	31
3.4.1	<i>Parameter Tuning</i>	33
3.4.2	<i>Comparative study with PSO variants</i>	36
3.4.3	<i>Discussions</i>	56
3.5	Conclusion	58
4	Ensemble Particle Swarm Optimizer	60
4.1	Introduction.....	60
4.2	Ensemble of Particle Swarm Optimization algorithms.....	63
4.2.1	<i>PSO variants in the ensemble approach</i>	63
4.2.2	<i>Ensemble Particle Swarm Optimizer (EPSO)</i>	67
4.3	Experimental Set Up and Results	70
4.3.1	<i>Experimental Set Up</i>	70
4.3.2	<i>Experimental Results</i>	72
4.3.3	<i>Algorithm Computational Complexity</i>	88
4.3.4	<i>Different search behaviours of EPSO on different benchmark problems</i>	90
4.3.5	<i>Discussion</i>	96
4.4	Conclusion	97
5	Hybrid Solution to UC Problem.....	98
5.1	Introduction.....	98
5.2	Hybrid Solution to UC	99
5.2.1	<i>UC Problem Formulation and Problem Statement</i>	99
5.2.2	<i>Unit-scheduling</i>	100
5.2.3	<i>Economic Dispatch</i>	102
5.3	Experimental Setup and Results	104
5.3.1	<i>Comparison of proposed time-ahead PL method and</i>	105

<i>Other PL methods.....</i>	<i>105</i>
5.3.2 <i>Comparison of PSOs vs. HCLPSO on ED sub-problem</i>	<i>106</i>
5.3.3 <i>Comparison of TPL-HCLPSO vs. hybrid PSO models</i>	<i>106</i>
5.3.4 <i>Comparison of proposed (TPL-HCLPSO) and other optimization methods.....</i>	<i>108</i>
5.4 <i>Conclusion</i>	<i>108</i>
6 Binary HCLPSO To UC with Renewable Resources	110
6.1 <i>Introduction.....</i>	<i>110</i>
6.2 <i>Binary HCLPSO to UC problem with Renewable Resources</i>	<i>111</i>
6.2.1 <i>UC Problem Formulation.....</i>	<i>111</i>
6.2.2 <i>Binary HCLPSO Approach to UC Problem.....</i>	<i>112</i>
6.2.3 <i>Experimental Setup.....</i>	<i>115</i>
6.3 <i>Conclusion</i>	<i>118</i>
7 Conclusion and Future Works	120
7.1 <i>Conclusions.....</i>	<i>120</i>
7.2 <i>Future Works</i>	<i>122</i>
7.2.1 <i>Futures works for PSO algorithm</i>	<i>122</i>
7.2.2 <i>Future works on solving power system problems by PSO</i>	<i>124</i>
Bibliography.....	126
Author's Publications.....	146
Appendix 1: Benchmark Functions.....	148
Appendix 2: Convergence Characteristic Graphs of HCLPSO Algorithm for CEC2005 Benchmark Functions	151
Appendix 3: Convergence Characteristic Graphs of EPSO Algorithm for CEC2005 Benchmark Functions	156

Summary

To meet increasing electricity demand and reduce tremendous greenhouse gas emission at the same time, the power infrastructure is integrated with renewable resources. Due to intermitted and unpredictable nature of renewable resources, the amount of generated power generated from renewable resources is uncertain and unreliable. The power scheduling problem becomes so complex that traditional optimization methods are no longer suitable for the current power systems. Computational intelligence (CI) methods have the adaptive ability to address this kind of complex and uncertain optimization problems from changing environments and have been demonstrated in many power system problems such as decision-making, forecasting, unit commitment, optimal power flow, renewable energy power systems, etc. Therefore, the particle swarm optimization algorithm, a variant of CI technique, is investigated and deployed to tackle unit commitment power problem with/without considering uncertainty in renewable resources.

The first part of the thesis is focused on the development of new particle swarm optimization (PSO) algorithms. PSO belongs to a class of evolutionary algorithms as well as swarm intelligence. PSO is known to be easy to implement and effective in solving large-scale non-linear optimization problems. In this thesis, a new heterogeneous comprehensive learning particle swarm optimization (HCLPSO) algorithm is proposed to improve the performance of PSO by balancing exploration and exploitation. In addition, Ensemble of particle swarm optimization algorithms called (EPSO) algorithm that combines various PSO algorithms is also proposed. The EPSO algorithm is focused on combining various PSO algorithms to improve optimization capability. Before addressing unit commitment problem in power system, the proposed algorithms are evaluated on CEC2005, CEC2013 and CEC2014 benchmark problems. A comparative study is conducted between the proposed PSO algorithms and recent algorithms. The experimental results show the performance of newly proposed PSO algorithms is better than other state-of-the-art PSO algorithms.

In the second part, we focus on the unit commitment (UC) problem, an important power optimization problem. A successful practical solution for this problem can benefit the industry in a huge operational cost saving. The problem determines start-

up and shut-down schedule of power generating units over a scheduled period while meeting the system demand and spinning reserve requirements at minimum production cost. In this thesis, a new priority listing method called time-ahead priority listing is introduced to generate a feasible unit schedule and the proposed HCLPSO is used to address economic dispatch of the UC problems of 10-, 20-, 40-, 60-, 80- and 100-bus power systems over 24 hour scheduling horizon. The performance of a hybrid solution of combining time-ahead priority listing and HCLPSO is compared with other hybrid methods. The proposed hybrid model outperforms other hybrid models on all small and large power systems and provides the lowest minimum production cost.

In the third part of the thesis, an independent system operator of 10-bus power system is integrated with renewable resources of solar and wind energies. A set of the scenarios are considered for uncertainty in solar, wind energies and power demands. Binary HCLPSO algorithm is proposed to solve cost-emission optimization of UC problem, considering reducing the greenhouse gas emission. The performance of binary HCLPSO algorithm is compared with integer-coded GA, improved binary PSO and hybrid approach of binary PSO and real-coded PSO. The results showed that binary HCLPSO algorithm is able to handle UC problem under certainty in renewable resources.

List of Tables

Table 2-1 Significant Application Areas of PSO Algorithms	17
Table 3-1 Parameter Settings.....	32
Table 3-2 Calibrating subpopulation sizes (g_1 & g_2) for HCLPSO algorithm	34
Table 3-3 Calibrating acceleration coefficients (c_1 and c_2) for HCLPSO algorithm....	35
Table 3-4 Comparison of HCLPSO vs other PSOs on CEC2005 10 dimensional problems	38
Table 3-5 Wilcoxon Signed Rank Test Results of Single-Problem Analysis with a Significance Level of $\alpha = 0.05$ for CEC2005 10 dimensional problems.....	40
Table 3-6 Comparison of HCLPSO vs other PSOs on CEC2005 30 dimensional problems	41
Table 3-7 Wilcoxon Signed Rank Test Results of Single-Problem Analysis with a Significance Level of $\alpha = 0.05$ for CEC2005 30 dimensional problems.....	43
Table 3-8 Comparison of HCLPSO vs other PSOs on CEC2014 30 dimensional problems.....	48
Table 3-9 Wilcoxon Signed Rank Test Results of Single-Problem Analysis with a Significance Level of $\alpha = 0.05$ for CEC2014 10 dimensional problems.....	50
Table 3-10 Comparison of HCLPSO vs other PSOs on CEC2014 30 dimensional problems	52
Table 3-11 Wilcoxon Signed Rank Test Results of Single-Problem Analysis with a Significance Level of $\alpha = 0.05$ for CEC2014 30 dimensional problems...	54
Table 4-1 Parameter Settings.....	73
Table 4-2 Comparison of EPSO vs. Other PSOs on CEC2005 10 Dimensional Problems....	75
Table 4-3 Comparison of EPSO vs. Other PSOs for CEC2005 30 Dimensional Problems....	77

Table 4-4 Comparison of EPSO vs other PSOs on CEC2013 10 dimensional problems.....	82
Table 4-5 Comparison of EPSO vs other PSOs on CEC2013 30 dimensional problems.....	85
Table 4-6 Computational Complexity of EPSO algorithm	90
Table 4-7 Comparison of Computational Complexity of EPSO vs its PSO variants on 10 D	90
Table 4-8 Comparison of Computational Complexity of EPSO vs its PSO variants on 30 D	90
Table 5-1 Data Specifications of 10-bus Test System.....	104
Table 5-2 Hourly Load Demand.....	104
Table 5-3 Comparison of Time-ahead PL method vs other PL methods	105
Table 5-4 Production Cost Difference of Time-ahead PL vs other PL methods.....	105
Table 5-5 Comparison of (PL-HCLPSO) hybrid model vs other hybrid models.....	106
Table 5-6 Production Cost Difference of PL+HCLPSO vs other hybrid models	106
Table 5-7 Comparison of new hybrid solution vs other hybrid solutions	107
Table 5-8 Production Cost Difference between new hybrid solution and other hybrid solutions.....	107
Table 5-9 Comparison of TPL-HCLPSO vs other methods in terms of production cost	108
Table 5-10 Comparison of TPL-HCLPSO vs other methods in terms of execution time	108
Table 6-1 A discrete distribution set of solar, wind and demand [136].....	116
Table 6-2 Emission Coefficients for 10-bus power system [219]	116
Table 6-3 Comparison of Binary HCLPSO vs other solutions to UC problem.....	118

Table 6-4 Scheduling and Power Dispatch of UC problem without Solar and Wind. 117

Table 6-5 Scheduling and Power Dispatch of UC problem with Solar and Wind 118

List of Figures

Figure 2-1	Hierarchical Topology	13
Figure 3-1	Learning Probability Curve.....	27
Figure 3-2	Diversity comparisons between exploration-subpopulation, exploitation- subpopulation and the whole swarm on Unimodal Shifted Sphere Function	30
Figure 3-3	Diversity comparisons between exploration-subpopulation, exploitation subpopulation and the whole swarm on Multimodal Shifted Rotated Griewank's Function	30
Figure 3-4	Convergence characteristic graphs of CEC2005 30 dimensional unimodal 30 functions: (a) F_4 : Shifted Schwefel's Problem 1.2 with Noise in Fitness and (b) F_5 : Schwefel's Problem 2.6 with Global Optimum on Bounds.....	44
Figure 3-5	Convergence characteristic graphs of CEC2005 30 dimensional multimodal functions: (a) F_{10} : Shifted Rotated Rastrigin's Function and (b) F_{11} : Shifted Rotated Weierstrass Function.....	45
Figure 3-6	Convergence characteristic graphs of CEC2005 30 dimensional expended extended functions: (a) F_{13} : Expanded Extended Griewank's plus Rosenbrock's Function (F_8F_2) and (b) F_{14} : Shifted Rotated Expanded Scaffer's F_6	46
Figure 3-7	Convergence characteristic graphs of CEC2005 30 dimensional composition functions: (a) F_{17} : Rotated Hybrid Composition Function with Noise in Fitness and (b) F_{25} : Rotated Hybrid Composition Function without Bounds.....	46
Figure 3-8	Convergence characteristic graphs of CEC2014 30 dimensional unimodal functions: (a) F_7 : Rotated High Conditional Elliptic Function and (b) F_3 : Rotated Discus Function	55

Figure 3-9	Convergence characteristic graphs of CEC2014 30 dimensional multimodal functions: (a) F_9 : Shifted and Rotated High Rastrigin's Function and (b) F_{10} : Shifted Schwefel's Function.....	55
Figure 3-10	Convergence characteristic graphs of CEC2014 30 dimensional expended functions: (a) F_{15} : Shifted and Rotated Expanded Griewank's plus Rosenbrock's Function and (b) F_{16} : Shifted and Rotated Expanded Scaffer's F_6 Function	56
Figure 3-11	Convergence characteristic graphs of CEC2014 30 dimensional hybrid functions.....	56
Figure 4-1	Convergence characteristic graph of EPSO and its respective self-adaptive PSO strategies for Unimodal Shifted Schwefel's Problem 1.2 function....	69
Figure 4-2	Convergence characteristic graph of EPSO and its respective self-adaptive PSO strategies for Multimodal Shifted Rotated Griewank's Function	69
Figure 4-3	Convergence characteristic graphs of CEC2005 30 dimensional unimodal functions: (a) F_1 : Shifted Sphere Function and (b) F_4 : Shifted Schwefel's Problem 1.2 with Noise in Fitness.....	79
Figure 4-4	Convergence characteristic graphs of CEC2005 30 dimensional multimodal functions: (a) F_6 : Shifted Rosenbrock's Function and (b) F_{12} : Schwefel's Problem 2.13.....	79
Figure 4-5	Convergence characteristic graphs of CEC2005 30 dimensional expended extended functions: (a) F_{13} : Expanded Extended Griewank's plus Rosenbrock's Function (F_8F_2) and (b) F_{14} : Shifted Rotated Expanded Scaffer's F_6	80
Figure 4-6	Convergence characteristic graphs of CEC2005 30 dimensional hybrid composition functions: (a) F_{16} : Rotated Hybrid Composition Function and (b) F_{23} : Non-Continuous Rotated Hybrid Composition Function.....	80

Figure 4-7	Convergence characteristic graphs of CEC2013 30 dimensional unimodal functions: (a) F_2 : Rotated High Conditioned Elliptic Function and (b) F_4 : Rotated Discus Function.....	87
Figure 4-8	Convergence characteristic graphs of CEC2013 30 dimensional multimodal functions: (a) F_6 : Rotated Rosenbrock's Function and (b) F_{14} : Schwefel's Function	88
Figure 4-9	Convergence characteristic graphs of CEC2013 30 dimensional expanded functions: (a) F_{20} : Expanded Scaffer's F_6 Function and composition function: (b) F_{28} : Schwefel's Function.....	88
Figure 4-10	Search behaviour plots for CEC2005 benchmark functions: (a) unimodal function F_1 and two multimodal functions: (b) F_6 and (c) F_8	93
Figure 4-11	Search behaviour plots for CEC2005 benchmark functions: (a) multimodal function F_{10} , (b) shifted expanded function F_{13} and (c) hybrid composition function F_{15}	94
Figure 4-12	Search behaviour plots for CEC2013 benchmark functions: (a) unimodal function (F_5) and multimodal functions: (b) F_8 and (c) F_{11}	95
Figure 4-13	Search behaviour plots for CEC2013 benchmark functions: (a) multimodal function F_{17} , (b) rotated expanded function F_{19} and (c) composition function F_{23}	96
Figure 6-1	The population structure of binary HCLPSO algorithm for UC problem.....	115
Figure 6-2	Flow chart of UC cost-emission optimization using binary HCLPSO algorithm.....	115
Figure 6-3	Irradiance Solar Data obtained from NREL solar prospector [136].....	116
Figure 6-4	Wind Speed Data obtained from NREL wind prospector [136].....	116

List of Symbols

N	number of generators;
T	total scheduling period;
i	index of unit ($i = 1, 2, \dots, N$);
t	index of hour ($t = 1, 2, \dots, T$);
$U_{i,t}$	unit on/off status of unit i at hour t ;
$P_{i,t}$	power generation of unit i at hour t ;
P_i^{max}	maximum power generation limit of unit i ;
P_i^{min}	minimum power generation limit of unit i ;
$P_{D,t}$	load demand at hour t ;
$P_{R,t}$	spinning reserve at hour t ;
T_i^{up}	minimum up time of unit i ;
T_i^{down}	minimum down time of unit i ;
T_i^{on}	duration during which unit i is continuously on;
T_i^{off}	duration during which unit i is continuously off;
T_i^{cold}	cold startup hours of unit i ;
SC_i^{hot}	hot startup cost of unit i ;
SC_i^{cold}	cold startup cost of unit i ;
$ST_{i,t}$	startup cost of unit i at t ;
$F_i(P_{i,t})$	fuel cost of unit i at hour t ;

a, b, c	fuel cost coefficients
$\mathcal{EC}_i(P_{i,t})$	emission cost of unit i at hour t ;
α, β, γ	emission coefficients
PC	production cost;
HR	heat rate;
P_t^{max}	maximum available power generation of power system at hour t ;
P_t^{ava}	available power after satisfying $P_{D,t}$ and $P_{R,t}$ at hour t ;
ψ_i	emission penalty factor;
s	scenario;
S	a discrete set of scenarios;
W_c, W_e	weight factors;
$P_{i,t}^s$	power of unit i at time t considering the scenario s ;
$P_{wind,t}^s$	wind power at time t considering the scenario s ;
$P_{solar,t}^s$	solar power at time t considering the scenario s ;
$P_{D,t}^s$	load demand at time t considering the scenario s ;

List of Appendices

Appendix 1: Benchmark Functions	148
Appendix 2: Convergence Characteristic Graphs of HCLPSO Algorithm for CEC2005 Benchmark Functions	151
Appendix 3: Convergence Characteristic Graphs of EPSO Algorithm for CEC2005 Benchmark Functions	156

1 INTRODUCTION

1.1 Balancing Exploration and Exploitation in Optimization

Optimization problems can be found in all areas of science and engineering such as power systems, computational chemistry, signal processing, communication engineering, etc. They can be classified into continuous/discrete, constrained/unconstrained, single/multi objective, convex/non-convex, differentiable/non-differentiable and deterministic/stochastic problems. In order to solve these different real world optimization problems, researchers have developed optimization algorithms such as particle swarm optimization (PSO), genetic algorithm (GA), differential evolution (DE), evolution strategy (ES), evolutionary programming (EP), artificial bee colony algorithm (ABC) and so on.

In these population-based evolutionary algorithms, finding the optimal solution of a problem is based on two cornerstones, namely exploration or the global search: exploring all over the search space to find promising regions and exploitation or the local search: exploiting the identified promising regions to fine-tune the search to locate the best solution. A good convergence behaviour of a population-based algorithm can be obtained when an appropriate balance between exploration and exploitation processes is found. Emphasizing on exploration will lead to waste of time searching over inferior regions of the search space and slow down the convergence rate. On the other hand, emphasizing on exploitation will cause loss of diversity early in the search process,

thereby possibly being stuck into a local optimum. Therefore, it is important to obtain a balance between exploration and exploitation in evolutionary algorithms [1, 2].

In PSO, the primary focus of this thesis, a population of candidate solutions is randomly initialized and each individual discovers optimal regions of the search space by following the direction of their own previous best experience and the best experience found by any particle in the entire swarm. If the best particle the population followed was near a local optimum and all the particles exploited the experience of the best, the population will face diversity loss and get stuck into the local optimum due to overexploitation. This current limitation of the existing PSO algorithms calls for a new methodology in PSO algorithm.

1.2 Ensemble of Optimization Techniques

Aforementioned in Section 1.1, many optimization algorithms have been developed, however, these successful and well-known evolutionary and swarm intelligence algorithms cannot be expected to perform well on any optimization problems. Different algorithms possess capabilities for solving different types of optimization. It is difficult to predict the best algorithm for each optimization problem. However, ensemble of different optimization algorithms could be a potential solution and more efficient than using one single optimization algorithm for solving complex problems. Inspired by this, ensemble of PSO algorithms is proposed.

Among population-based evolutionary algorithms, PSO is easy to implement and well-known for having the ability to quickly converge to an optimal solution [3]. Many PSO variants have been developed and applied to solve different optimization problems. However, no PSO can be found performing consistently well on various types of optimization problems. For example, a PSO algorithm may perform well on unimodal problems; however, it may perform poorly on complex multimodal problems. Each PSO algorithm has their own advantages and disadvantages on solving different types of optimization problems. By hybridizing them together, they can potentially complement each other to solve various optimization problems. Motivated by this, this thesis aims to provide a new perspective and proposal to tackle this issue through ensembling various PSO techniques.

1.3 Optimization problems in Power Industry

The infrastructure of electric power systems is undergoing changes in structure, operation, planning, control, management, and ownership areas due to increasing global power demands. The existing power infrastructure has been serving us for more than a hundred year and it is rapidly running up against its limitations. The reliability of the grids becomes threatened and the probability of more frequent blackouts and faults can become higher [4]. Besides, the power systems are kept expanding due to global electricity demand and the power grid grows in size, scale and complexity as well as the risks. To reduce the greenhouse gas emission, clean and renewable energy sources are replacing convention non-renewable energy sources. Thus, the existing power infrastructure is facing challenges in integrating with renewable resources. This is the major driving force for evolution of the intelligent power system.

It has been well-known that the renewable resources are intermittent and unpredictable due to uncertainty of weather conditions. This increases complexity in power system operation, planning, control and automation. As a result, the traditional optimization methods may not be appropriate for the current power industry. Hence, the power industry needs solutions that can intelligently manage the system operations, planning and control. One of the plausible solutions is Computational intelligence (CI). CI is “the study of adaptive mechanisms to facilitate intelligent behavior in complex, uncertain and changing environments” [5, 6]. CI techniques have the ability to adapt to the nonlinearity and uncertainty facing in power systems today. In addition, various CI techniques have been proposed for the smart grid and employed to solve the power system problems in decision making, forecasting, unit commitment, optimal power flow, etc. [7-9].

In this thesis, the particle swarm optimization algorithm, a variant of CI technique, is investigated and tackled unit commitment power problem with/without renewable resources.

1.4 Major Contributions of the Thesis

The major contributions of the thesis are:

- To address the balancing between exploration and exploitation issue, heterogeneous comprehensive learning particle swarm optimization algorithm with enhanced exploration and exploitation (HCLPSO) is proposed to improve the performance of PSO by balancing exploration and exploitation. In the proposed HCLPSO algorithm, a heterogeneous swarm of two subpopulations is used and these subpopulations are designated to perform exploration and exploitation process respectively. The exploration-subpopulation explores the search space independently and the exploitation-subpopulation exploits the information of the whole population to search the solution over the search space. Comprehensive learning (CL) strategy is used to generate the exemplars for both subpopulations. In the exploration-subpopulation, the exemplars are generated by using personal best experiences of the particles from the exploration-subpopulation itself. In the exploitation-subpopulation, the personal best experiences of the entire swarm population are used to generate the exemplars. The exploration-subpopulation does not learn from any particles in the exploitation-subpopulation. Thus, it enables the exploration-subpopulation to maintain the population diversity even if the exploitation-subpopulation would converge prematurely to the local optimum.
- The ensemble of particle swarm optimization algorithms, namely EPSO algorithm, is proposed. The EPSO combines different PSO algorithms to solve various kinds of optimization problems. In each generation, a self-adaptive scheme is employed to identify the top algorithms by learning from their previous experiences in generating promising solutions. Consequently, best-performing algorithm can be determined adaptively for each generation and assigned to individuals in the population
- To tackle the unit commitment (UC) problem in power system, a new hybrid method is proposed. The UC problem is divided into two sub-problems: unit scheduling and economic dispatch. A new priority listing method called time-ahead priority listing method is introduced to solve unit-scheduling sub-problem and the proposed HCLPSO algorithm is used to solve the economic dispatch sub-problem.

- In order to study unit commitment with renewable resources under uncertainty, the 10-bus power system is integrated with renewable resources of solar and wind energies and a set of scenarios is considered for the uncertainty of solar and wind renewable resources and load demand in the power system. Then, binary HCLPSO algorithm is proposed to solve UC problem under uncertainty with the objective of reducing the power production cost as well as the emission cost.

1.5 Organization of the Thesis

This thesis is comprised of seven chapters.

- Chapter 1 presents motivations, objectives, contributions, and organization of this thesis.
- Chapter 2 introduces particle swarm optimization (PSO) algorithm and presents a comprehensive literature review of different PSO algorithms and its applications in power system and other engineering areas. A brief introduction of unit commitment power problem and its optimization methodologies are also presented in Chapter 2.
- Chapter 3 proposes heterogeneous comprehensive learning PSO (HCLPSO) algorithm. The performance of the proposed HCLPSO algorithm is evaluated on CEC2005 and 2014 benchmark problems. Non-parametric statistical test results show the HCLPSO algorithm is outperformed over other recent PSO algorithms.
- Chapter 4 presents the ensemble of PSO algorithms called EPSO algorithm to enhance PSO's performance in solving different kinds of optimization problems. The performance of EPSO is evaluated by using unimodal, multimodal, expanded and hybrid composite benchmark problems from CEC2005 and CEC2013. The experimental results show that the proposed EPSO is able to perform well on solving most problems and the statistical test results prove that the performance of the proposed EPSO is better or comparable to its individual PSO as well as recent PSO algorithms.

- Chapter 5 proposes a new hybrid solution to handle unit commitment problem in power system. Unit commitment (UC) problem is divided into two sub optimization problems: unit scheduling problem and economic dispatch problem. Time-ahead priority listing method is proposed to solve unit schedule sub-problem and HCLPSO is applied to solve economic dispatch sub-problem. The proposed hybrid model is tested on 10-, 20-, 40-, 60-, 80-, 100-bus systems and compared with other optimization methods. The results show that proposed hybrid solution obtains the lowest minimum production cost and offers superior performance compared to other methods.
- Chapter 6 presents cost-emission optimization of UC problem with renewable resources under uncertainty. The independent system operator of 10-bus power system is considered for the simulation with renewable sources and binary HCLPSO algorithm is employed to solve the UC problem under certainty to reduce both power production cost and CO₂ emission cost. Integrating the power systems with renewable resources, the production cost becomes lower and the emission is reduced.
- Chapter 7 presents conclusion and future works.

2 LITERATURE REVIEW

Computational intelligence (CI) is an offshoot of artificial intelligence [10, 11], which combines elements of learning, adaptation, evolution, and fuzzy logic and emphasizes on heuristic algorithms. The three main pillars of CI are evolutionary computation (EC), fuzzy systems, and neural networks. CI also covers artificial immune systems, chaotic systems, probabilistic methods, and swarm intelligence. CI is a new modern computation tool for solving complex problems that are challenging for conventional approaches. The PSO algorithms are intelligent general purposed optimization techniques that can handle adaptability and stochasticity. It has been applied to solve various types of optimization problems in many industries.

Among the various CI algorithms, particle swarm optimization (PSO) algorithm has been investigated in this thesis. PSO is a population-based swarm intelligence algorithm and find the optimum by moving a population of candidate solutions that attempt to improve every iteration. Compared to other CI algorithms, PSO is:

- Easy to implement
- Able to converge quickly to the optimum
- Flexible to be integrated with other optimization techniques
- Less sensitive to the nature of the objective function

- Fewer parameters to tune unlike other population-based optimization algorithms
- Able to handle objective functions with stochastic nature

Due to these advantages over other CI optimization techniques, PSO has been proposed to solve the unit commitment problem with/without renewable resources. In this Chapter, the comprehensive literature reviews of PSO algorithm and its application to unit commitment problem in power system will be presented.

2.1 Particle Swarm Optimization

Particle swarm optimization (PSO) is a population-based optimization technique introduced by Eberhart and Kennedy [12]. The PSO algorithm is inspired by swarm social behaviour such as bird flocking and fish schooling. Without colliding with each other, a flock of birds or a school of fish is able to search for the food or shelter. Members share information within the swarm. Each member updates its direction by using its own findings and the whole swarm's information. Imitating this swarm social behaviour, the PSO algorithm was developed in [12]. In PSO algorithm, each particle in the swarm represents a potential solution to a given problem. The particles navigate by adjusting their flying direction using their own and other swarm members' best experiences to find the optimum of the problem. This phenomenon was formulated in [12] as follows:

$$V_i^d = V_i^d + c_1 * rand1_i^d * (pbest_i^d - X_i^d) + c_2 * rand2_i^d * (gbest^d - X_i^d) \quad (2.1)$$

$$X_i^d = X_i^d + V_i^d \quad (2.2)$$

where, i represents each particle in the population ($i = 1, 2, \dots, N$ number of populations) and d represents dimension ($d = 1, 2, \dots, D$ dimensions). X_i^d and V_i^d are the position and velocity in the d^{th} dimension of the i^{th} particle in the population respectively. $pbest_i^d$ is its own best position of i^{th} particle and $gbest^d$ is the best position found by whole swarm population. c_1 and c_2 are acceleration coefficients and control trajectory of i^{th} particle. $rand1_i^d$ and $rand2_i^d$ are randomly generated numbers in the range of $[0, 1]$.

In order to perform search in a controlled manner, the updated velocity V_i^d is limited with a maximum magnitude V_{max} . If $|V_i^d|$ exceeds V_{max} , then $V_i^d = \text{sign}(V_i^d) * V_{max}$ for such dimensions. The constraint V_{max} controls exploration and exploitation ability of each particle in the swarm. The larger V_{max} velocity means global exploration and the smaller V_{max} velocity refers local exploitation. To achieve a balance between global exploration and local exploitation, a new parameter 'w' called inertia weight was introduced into PSO and used to control the flying velocity as follows [13]:

$$V_i^d = w * V_i^d + c_1 * \text{rand}1_i^d * (pbest_i^d - X_i^d) + c_2 * \text{rand}2_i^d * (gbest^d - X_i^d) \quad (2.3)$$

In [13], [3], inertia weight w was defined as a linearly decreasing function of the run time, decreased linearly from the value 0.9 to 0.4. The authors suggested that 'w' could be a positive constant or positive linear/non-linear function of run time and by choosing an appropriate value for w , we could obtain the balance between exploration and exploitation process.

Clerc and Kennedy developed another balance control parameter χ called constriction coefficient into the original PSO to control dynamic characteristics of the particle swarm, including its exploration and exploitation tendencies as follows [14].

$$V_i^d = \chi [V_i^d + c_1 * \text{rand}1_i^d * (pbest_i^d - X_i^d) + c_2 * \text{rand}2_i^d * (gbest^d - X_i^d)] \quad (2.4)$$

where, constriction coefficient is calculated as:

$$\chi = \frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|} \quad \text{if } \varphi > 4 \quad (2.5)$$

$$\varphi = c_1 + c_2 \quad (2.6)$$

The constriction χ is set as $\chi=0.729$ with $c_1=c_2=2.05$ and $\varphi=4.1$ [15]. The inertia weight and constriction coefficients are mathematically equivalent if the inertia weight w is set to be χ and c_1 and c_2 satisfies the conditions ($\varphi = c_1 + c_2, \varphi > 4$) [16].

2.1.1 Prominent PSO Variants

Due to its simplicity and ease of implementation, PSO had been widely used to solve benchmark functions and real-world optimization problems from different appli-

cation fields and research areas. Many variant types of PSO algorithms had been developed in the literature and the performance of PSO was mainly improved by the following approaches:

1. Modifying population structure/topology
2. Adaptation of control parameters
3. Hybridizing with other search techniques/evolutionary operators
4. Different PSO learning Strategies
5. Other PSO Variants

2.1.1.1 Population Topology

As mentioned above, PSO is inspired by swarm social behaviour such as bird flocking and fish schooling [12] and it is based on a social-psychological model of social influence and social learning [17]. A population of candidate solutions is randomly initialized and each individual discovers optimal regions of the space by imitating the behaviour of its most successful neighbour in the population. Thus, the sociometric topology of PSO determines the breadth of influences on the individual and the performance of each individual depends on the population topology used [18]. It is also well accepted in the literature that the population topology significantly influenced the performance of PSO [12]-[20]. Nowadays, many population topologies have been proposed for PSO algorithm and the topologies are studied in terms of static, dynamic, multi-swarm/heterogeneous and hierarchical topology.

In consideration of static topology, Kennedy introduced small-world social network [21] into PSO and investigated different population topologies of circles, wheels, stars, and randomly assigned edges in [18]. The findings showed that the sociometry of a swarm population significantly affects its ability to find the optimum and the effect was dependent on the problem to be optimized. The populations with fewer connections might perform better on solving multimodal problems and highly interconnected population could perform well on solving unimodal problems. Based on those findings, Kennedy and Mendes introduced heterogeneous population structure, with some subpopulations tightly connected and other relatively isolated, and studied the

effect of *gbest*, *lbest*, star, small, pyramid and von Neumann population topologies by varying the number of neighbours or the number of neighbours in common [20]. The authors conceptualized the influence within the swarm population as information flow between the particles and recommended the PSO researchers to try von Neumann sociometry.

Mendes proposed the fully informed particle swarm (FIPS) algorithm in which the information of all the neighbours is used to guide a particle in the swarm, rather than using only the information of the best particle [22]. The performance of FIPS was evaluated using five different topologies of *gbest*/all, ring, four clusters, pyramid and square. The results showed that FIPS with all sociometries except *gbest* topology were able to find the optimum [22]. In [23], the best experiences from local neighborhood and global neighborhood were used in unified particle swarm optimization (UPSO) algorithm by combining their exploration and exploitation abilities. The paper mentioned that the neighborhood size should be selected properly to get trade-off between exploration and exploitation. Clerc proposed the random topology in [24] and the adaptive topology in [14]. Later, the random topologies were studied in detail and introduced to dynamic PSO in [25]. Li suggested that a PSO with ring topology can be used as a niching algorithm without requiring niching parameters in [26].

In the context of dynamic topology, Suganthan proposed a dynamic population topology in [27]. In this topology, a small number of particles are selected as the neighbours at the early iterations and then the neighborhood size is dynamically increased over iterations until the particles are fully connected at the end of the iterations. Nasir developed dynamic neighbourhood learning PSO (DNLPSO) where simple ring topology was used as neighbourhood topology and the neighbours of each particle were changed after certain intervals [28]. The dynamic neighbourhood topologies were proposed in [29] where the neighbours of each particle were dynamically changed over time. At each generation, the neighbours are selected randomly from the population or the nearest neighbours in the search space are selected as neighbours in [29]. In [30], dynamic sociometry was introduced into PSO in which the number of neighbours was randomly increased at any successful iteration to change the swarm search ability from the exploration to exploitation. In [31], weighted Euclidean distance was used to select the neighbours. For each vector of a particle, the particles with the highest fitness-

distance-ratio (FDR) were selected as the neighbors. PSO with dynamic random population topology was proposed in [32] where various random population topologies were generated and used during the evolution. The dynamically changing ring topology and dynamic multi-swarm topology were proposed in [33] and [34] respectively.

In terms of multi-swarm/heterogeneous topology, a heterogeneous particle swarm optimization (HPSO) was proposed in [35]. In HPSO, the particles in heterogeneous swarms were allowed to follow different velocity and position updating rules from a behaviour pool, thereby having the ability to explore and exploit throughout the problem search space. The author also proposed two different models of HPSO, static HPSO (sHPSO) and dynamic HPSO (dHPSO) in [36]. Then, the performance of dHPSO was investigated for optimization in a dynamic environment, using dynamic unconstrained optimization problems in [62]. Nepomuceno proposed self-adaptive HPSO called frequency-based HPSO in which the success-frequency of the different behaviours used by the particles was tracked for a number of iterations and the new behaviour was selected based on its success-frequency [37].

A cooperative approach was applied to PSO (CPSO-Sk) in which the search space dimensionality was split and different swarms were used to search over different dimensions of a solution cooperatively [38]. A multi-swarm PSO using charged particles (PSO-2S) was developed in [39] where the search space was partitioned into two swarms called main and auxiliary swarm. In PSO-2S, the auxiliary swarms are initialized using charged particles in different partitioned areas. After certain number of generations, main swarm is formed with the best individual of the auxiliary swarms to search for the optimum. Multi-swarm idea have also been used for locating multiple optimal solution in [40] and for dynamic environments in [41]-[43]. Clerc proposed an autonomous version of PSO called TRIBES which did not required any parameters [44]. In TRIBES, the entire swarm is divided into subswarms and each subswarm has its own order and topology. The population size and topology evolve over time depending on the performance feedback. Good tribes can reduce their population and bad tribes can recruit a new member. The population size is adaptive and the population structure is modified every $L/2$ iteration, where L is the number of links in the population [44].

Hierarchical topology is also known as tree topology, which has a central node at the highest level of the hierarchy. The central node is connected to one or more

nodes at the next lower level. The central node has one particle and the nodes in the lower hierarchy will have one particle or more than one particle as shown in Figure 2-1. In this way, the hierarchical topology is formed. The information is exchanged only between the particles at the same level.

Figure 2-1 Hierarchical Topology

Janson and Middendorf proposed the hierarchical version of PSO (H-PSO) in which dynamic hierarchy was used to define a neighbourhood structure [45]. In H-PSO, all particles are connected using the hierarchical topology in which each node has exactly one particle. Depending on the quality of their best solutions found so far, the particles move up or down the hierarchy so that the best particle has the highest influence on the swarm [45]. H-PSO was used to solve the dynamic optimization problems in [46] and its adaptive variants were presented in [47]. A new variant of hierarchical PSO called hierarchical dynamic local neighbourhood PSO (H-DLPSO) was proposed in [48]. In H-DLPSO, each level of hierarchy is composed of a number of subswarms of the particles and the information is shared among the subswarms at the same level of hierarchy. The subswarms are randomly regrouped in each generation and dynamically varied along with H-PSO version [48].

2.1.1.2 Control Parameter Adaptation

Compared to other metaheuristic methods, PSO has fewer parameters to be tuned. As described in equation (2.3), inertia weight w , acceleration coefficients c_1 and c_2 are three important control parameters in PSO. Shi and Eberhart introduced inertia weight w into PSO in [13] and analysed the impact of inertia weight ' w ' on the performance of PSO in [49]. The authors provided guidelines to select suitable w to obtain a

balance between PSO global and local search in [49] and introduced a linearly decreasing inertia weight w into PSO in [3]. Later, a fuzzy system is implemented to adjust w dynamically and introduced fuzzy adaptive inertia weight to improve PSO performance in [50]. A survey of different kinds of non-linear [51], linearly increasing [52], random [53], Gaussian [54] and exponential [55] inertia weight w were studied in [56].

Ratnaweera firstly introduced time varying acceleration coefficients in [57] where c_1 was a linear decreasing function of time and c_2 is a linear increasing function of time. Gaussian probability distribution was used to generate c_1 and c_2 in [58]. Adaptive PSO (APSO) was developed to maintain the population diversity by adaption of PSO control parameters (inertia weight and acceleration coefficients) in [59]. Adaptive fuzzy PSO (AFPSO) was proposed in [60] where three fuzzy rules are implemented to dynamically adapt the two acceleration coefficients. Most recently, Zhang introduced an adaptive control parameter mechanism into PSO with multiple adaptive methods and proposed APSO-MAM algorithm in [61]. In APSO-MAM, the parameter control problem is formulated as a convex optimization problem and subgradient method is used to control the three control parameters adaptively.

2.1.1.3 Hybridizing PSO with other search techniques

Each optimization method has its own merits and demerits in solving different optimization problems of high dimensional, multimodal, discontinuous, non-convex and non-differentiable optimization problems. By hybridizing different optimization methods, they can complement each other and will be able to handle different optimization problems, rather than a specific one. Therefore, hybridizing PSO with other evolutionary operators or search techniques is an alternative way to improve the performance of PSO algorithm.

Angeline added genetic selection operator to PSO and introduced the hybrid PSO algorithm in [62]. As a result, the crossover and mutation evolutionary operators were also combined into PSO in [63]-[65]. PSO was hybridized with differential evolution operator in [66] and ant colony algorithm in [67, 68]. Recently, PSO is hybridized with artificial bee colony algorithm and the performance of the hybrid algorithm was tested on CEC2013 benchmark optimization problems in [69]. Cooperative PSO

(CPSO) [38] algorithm was combined with cultural algorithm and applied to back propagation training in neural fuzzy network in [70].

In order to improve PSO's local search ability, several local search techniques were integrated into PSO and studied in [71]-[74]. In addition, the social and nature biological framework were also introduced into PSO such as niching PSO in [75], cultural-based PSO in [76] and PSO with an aging leader and challengers (ALC-PSO) in [77].

2.1.1.4 PSO Learning Strategies

Liang proposed comprehensive learning particle swarm optimizer (CLPSO) in which each particle learnt from other particles' best experiences for different dimensions via a comprehensive learning strategy [78]. In CLPSO, learning probability curve is set so that the particles have different levels of social learning abilities. Orthogonal learning strategy was developed in which a particle learnt from the combination of its own best experience (cognitive learning) and its neighbourhood best experience (social learning) [79]. In self-learning PSO (SLPSO), each particle has a set of four strategies, namely convergence, exploitation, exploration, and jump out, to deal with different situations in the search process [80]. Then, an adaptive learning mechanism was used to enable particles to select the appropriate strategy according to their fitness landscape.

A competitive swarm optimizer (CSO) was developed where neither personal best position nor global best position was involved in updating the particles' positions [81]. In CSO, the two particles are randomly selected to compete and the loser will update its position by learning from the winner and mean position of current swarm. Scatter learning strategy was proposed in a scattering learning particle swarm optimization algorithm (SLPSOA) [82]. In scatter learning strategy, the exemplar pool (EP) is constructed with a number of relatively high-quality solutions scattered in the solution search space. Then, the particles select their exemplars from EP using roulette wheel rule to enable the particles to explore different regions and the selected exemplar is used for a certain number of iterations to exploit the corresponding region thoroughly. Incremental social learning is introduced into PSO in [83]. In IPSO, the population size is growing over time and whenever a new particle is added to the population, social learn-

ing rule is used to initialize the new particle's position, biased towards the best particle [83].

2.1.1.5 Other PSO Variants

Chen proposed PSO with adaptive population size in [84] where the particles were periodically increased or decreased in the form of a ladder function according to the population diversity value. Efficient population utilization strategy (EPUS-PSO) was proposed in [85]. In EPUS-PSO, solution sharing and search range sharing strategies are proposed to avoid the particles from being trapped in a local optimum. A population manager is adopted and varied population size according to the status of the solution search. PSO with variable population size was presented in [86]. In VarPSO, each particle has life time expressed in quantity of iterations and the population size is varied according to the age and the neighbourhood [86].

Kennedy proposed barebones PSO (BBPSO) algorithm in which the velocity and position represented the actual new position of the particle, sampled from probability density functions such as Gaussian distribution [87]. The BBPSO operates by sampling points on the problem space and uses the discovered knowledge to guide the search. In order to jump out of the local optimum during the search, the jump strategy was proposed for BBPSO based on Gaussian and Cauchy probability distribution in [88]. The collapse of BBPSO, a possible swarm pathology which can lead to swarm stagnation, is studied in [89] and the BBPSO with scale matrix adaption was proposed in [90].

Beyond solving single objective optimization problem with continuous search space, the PSO algorithm is also extended to solve multi-objective [72, 91-93], constrained [94-96], discrete and binary [97-99] optimization problems.

2.1.2 Applications

As discussed in previous session, a large number of PSO variant algorithms have been developed over the years and some have been implemented in various application areas. The significant engineering applications of PSO algorithm are listed in Table 2-1.

Table 2-1 Significant Application Areas of PSO Algorithms

Areas	Applications	References
Electrical Power Systems	Unit Commitment	[95, 99-107]
	Economic Dispatch	[108-116]
	Optimal Power Flow	[96, 117-119]
	Reactive Power and Voltage Control	[120-124]
	Distribution Network Reconfiguration	[125-127]
	Power System Controller Design	[128-130]
	Renewable Energy Power Systems	[131-136]
Communication Networks	Wireless Sensor Networks	[137-139]
	Antenna Design	[140-142]
Automatic Control Systems	Robotic	[143, 144]
	Fuzzy Control	[145, 146]
	System Identification	[147, 148]
	Controller Design	[130, 149, 150]
Pattern Recognition and Image Processing	Image Segmentation	[151]
	Image Classification	[152, 153]
	Image Clustering	[154, 155]
Artificial Neural Networks	Neural Network Training	[156-159]
Signal Processing	Speech Recognition	[159, 160]
	Digital Filter Design	[161, 162]
Other Engineering Areas	Civil and Mechanical Engineering	[163, 164]

2.2 Unit Commitment Problem

In a power system, unit commitment (UC) problem is a constrained optimization problem. The problem determines start-up and shut-down schedule of power generating units over a scheduling period while meeting the system demand and spinning reserve requirements at minimum production cost. Solving UC using computational intelligence algorithms requires three important procedures [165]:

- Problem formulation and problem statement
- Generating feasible unit schedule
- Solving economic dispatch problem

After formulating the problem, there are two sub-problems left to solve in UC: unit-scheduling and economic dispatch sub-problem. Unit-scheduling sub-problem is an on/off or 0/1 combinatorial optimization problem to determine the feasible unit schedule that satisfies system load demand and spinning reserve requirements as well as start-up

and shut-down time of each generation unit during the scheduling period. On the other hand, economic dispatch (ED) sub-problem is focused on finding the feasible power output for each generation unit with minimum overall production cost. The ED problem is a non-linear constrained optimization problem.

Therefore, UC problem has been commonly formulated as a nonlinear mixed-integer combinatorial optimization problem with system and unit constraints. As a non-linear problem, the exact solution can be obtained only by complete enumeration. However, this method requires large computation time for realistic power systems. It motivated the researchers to develop new approaches that can be applied to real world large-scale power system with reasonable computational time. Various optimization methodologies have been developed and applied in order to obtain optimal solution to UC problem in literature. The mathematical formulation and optimization methodologies of UC problem will be discussed in the next few sections.

2.2.1 Unit Commitment Problem Formulation

According to the types of the power system, such as regulated or deregulated power systems, integrated with or without renewable energy resources, the mathematical formulation of UC problem differs based on its objective function and constraints [166]. In this report, a mathematical formulation of general UC problem is presented for easy understanding UC problem in the power system. The generic UC problem of a typical power system can be formulated mathematically as production cost minimization problem, subject to system and unit constraints as follows [166]:

Objective Function: Minimize Production Cost (PC)

$$PC = \sum_{i=1}^N \sum_{t=1}^T [F_i(P_{i,t}) + ST_{i,t}(1 - U_{(i-1,t)})]U_{i,t} \quad (2.7)$$

where, N is the number of generators, T is the number of time steps in scheduling period and $P_{i,t}$ is the generation of unit i at time t , $U_{i,t}$ is on/off status of unit i at time t ($on=1$ and $off=0$). $F(P_{i,t})$ is the fuel cost of unit i at time t which is a function of generator output and is generally expressed in the quadratic polynomial form as follows:

$$F_i(P_{i,t}) = a_i + b_i P_{i,t} + c_i P_{i,t}^2 \quad (2.8)$$

where, a_i , b_i , c_i represent the unit operating cost coefficients. $ST_{i,t}$ is start-up cost unit i at time t . The generator start-up cost depends on the time (T_i^{off}) the unit has been off prior to start up as follows:

$$ST_{i,t} = \begin{cases} SC_i^{hot} & \text{if } T_i^{off} \leq T_i^{down} + T_i^{cold} \\ SC_i^{cold} & \text{if } T_i^{off} > T_i^{down} + T_i^{cold} \end{cases} \quad (2.9)$$

where, T_i^{down} is the minimum down time of unit i . T_i^{cold} is the cold start-up time and defines the number of hours that takes the generator unit to cool down. SC_i^{hot} and SC_i^{cold} are hot and cold start-up cost of unit i respectively. The cold start-up cost is higher than the hot start-up cost.

Constraints: UC problem is subject to the following system and unit constraints:

System Constraints

1. *Power Balance Constraint:* The power generated from the committed units must satisfy the system power load demand ($P_{D,t}$) at every time step t .

$$\sum_{i=1}^N P_{i,t} U_{i,t} = P_{D,t} \quad (2.10)$$

2. *Spinning Reserve Constraint:* In order to prevent load interruption from certain equipment outages, spinning reserve constraint $P_{R,t}$ is required in the operation of a power system. The reserve is usually considered 5 or 10 percentage of the forecasted load demand.

$$\sum_{i=1}^N P_i^{max} U_{i,t} \geq P_{D,t} + P_{R,t} \quad (2.11)$$

Unit Constraints

3. *Power Generation Limit Constraint:* Generation power of each unit is limited by the minimum P_i^{min} and maximum P_i^{max} power values as follows:

$$P_i^{min} \leq P_{i,t} \leq P_i^{max} \quad (2.12)$$

4. *Unit Minimum Up and Down Time Constraints:* Each unit must be on/off for a certain number of hours before shutting down/turning on.

$$T_{i,t}^{on} \geq T_i^{up} \quad (2.13)$$

$$T_{i,t}^{off} \geq T_i^{down} \quad (2.14)$$

5. *Unit Initial Status*: The initial on/off status of units at the start of the scheduling period must be taken into account in optimization.

2.2.2 Optimization Methodologies

In literature, a large number of optimization methods have been applied to solve the UC according to the nature of the power system and its unit commitment model. Methodologies applied to the UC can be classified into three categories as follows:

1. Deterministic Methodologies
2. Metaheuristic Methodologies
3. Hybrid Methodologies based on the two categories above

2.2.2.1 Deterministic Methodologies

The most popular and commonly used deterministic methods are exhaustive enumeration [167], priority listing [168, 169], dynamic programming [170-172], lagrangian relaxation [173, 174], branch and bound [175, 176], integer/linear/mixed-integer linear programming [177-180] methods. In exhaustive enumeration method, all possible combinations of generation units are enumerated and the combination with the least cost of operation is chosen as the optimal solution to UC problem. Although the exact solution to the UC problem can be obtained by exhaustive enumeration method, it is not applicable to the large-scale power systems due to its excessive computational time. Priority listing (PL) methods are simple and fast, but highly heuristic.

Dynamic programming (DP) methods are flexible and capable of handling a variety of problem sizes, but computationally expensive. Lagrangian relaxation (LR) methods provide a faster solution and are capable of solving the power systems with a large number of units. However, when the problem is non-convex, LR method suffers from numerical convergence problem. Besides, the solution quality of LR strongly de-

depends on the method used to initialize and update Lagrange multipliers. Thus, LR sometimes generates the unsatisfactory solutions. Branch and bound (BB) method searches for the optimal solution between the candidate solutions by discarding the candidates that are beyond the estimated upper and lower bounds of the problem. The issue in BB is the exponential growth in the execution time with the size of the problem. Integer and mixed-integer programming adopts linear programming method to solve the UC problem. The assumptions made by these methods limit the solution space limiting their applicability to systems with a small number of units.

2.2.2.2 Metaheuristic Methodologies

The shortcoming of deterministic methods is that they can provide a feasible solution but may not yield the qualified solution making them unsuitable for large-scale power systems. Recently, metaheuristic methods had gained popularity in order to overcome the shortcomings of deterministic methods. The metaheuristic methods applied to solve UC problem are expert systems [181], Fuzzy logic systems [182], artificial neural network [183], tabu search [184], simulated annealing [185], evolutionary programming [186], genetic algorithm [187], differential evolution [188], ant colony algorithm [189], particle swarm optimization [104] methods.

Among these metaheuristic methods, genetic algorithms (GA), simulated annealing (SA) and particle swarm optimization (PSO) are frequently applied to solve the UC problem. GA method is a general-purpose stochastic and parallel search method and has the potential of obtaining a global optimal. However, the disadvantage of this method is long execution time. SA is a powerful optimization technique that can theoretically converge asymptotically to a global optimum solution but it takes a large execution time to reach the near-global minimum. PSO is a stochastic search method inspired by social behaviour of fish schooling and bird flocking. It is easy to implement and able to converge quickly to an optimum. However, PSO sometimes suffers from the premature convergence due to the diversity loss near a local optimum. In spite of that, PSO is capable of efficiently exploring the search space and generating good quality solutions.

2.2.2.3 Hybrid Methodologies

Compared to the deterministic methods, metaheuristic methods provide better solution with suitable execution time for a reasonable size of the power system. However, they consume a lot of execution time when the size power system increases. Another solution approach to UC is hybrid methodologies based on deterministic and metaheuristic methods. On one hand, an optimization method may be simple but may offer suboptimal solution. On the other hand, the method may be complex but accurate. The fundamental idea of hybrid methodologies is to combine different methods so that one method can complement each other. Therefore, hybrid methods combining the above-mentioned optimization methods (either by combining different methodologies together or the same methodology (deterministic/metaheuristic) together) have been proposed and deployed to solve the UC problem in [102, 190-192]. The results showed that the hybrid methods were capable of obtaining better solutions while handling constraints that are more complicated.

3 HETEROGENEOUS COMPREHENSIVE LEARNING PSO

3.1 Introduction

Among population-based evolutionary algorithms, PSO is easy to implement and has performed well on many optimization problems. PSO is also known for having the ability to quickly converge to an optimum [3]. However, in PSO, all particles share its swarm's best experience (the global best) that can lead the particles to cluster around the global best. If the global best is located near a local minimum, escaping from local optimum becomes difficult and PSO suffers diversity loss near the local minimum [13]. Thus, it is obvious that PSO suffers from the lack of the balance between the exploration and exploitation.

In order to balance the exploration behavior (global search) and exploitation nature (local search) in PSO, researchers have addressed this issue by suggesting different methods of population topology in [17, 18, 20, 35], adaptation of control parameters in [13, 15, 57] and different learning strategies in [78, 79, 82], as mentioned in Chapter 2.1.1. Inspired by those methods, a comprehensive learning PSO with two sub-population groups, called heterogeneous CLPSO (HCLPSO), is proposed in this chap-

ter. Instead of relying on one method to balance the exploration and exploitation ability of PSO, this paper addresses the issue by the following methods: 1) using adaptive control parameters, 2) controlling the information sharing (or topology) among the particles, 3) using the learning strategy, and 4) using heterogeneous swarm rather than homogeneous.

In the proposed HCLPSO algorithm, a heterogeneous swarm is used where the swarm is divided into two subpopulations. Each subpopulation is assigned to carry out the exploration and exploitation search respectively. Exploration and exploitation processes are enhanced without one process crippling the other. In both subpopulations, comprehensive learning strategy [78] is used to generate an exemplar for the particles to learn. Generally, the conventional PSO learnt from the two exemplars, personal best and whole swarm's best. This learning strategy can cause two problems. One is "oscillation phenomenon" [193] that can occur if the two experiences were in opposite directions. This makes the search ability inefficient and slows down the convergence speed of the algorithm. Another is "two steps forward, one step back phenomenon" [38] that causes the solution vector to improve on some dimensions and to decline on other dimensions as one exemplar may have good values on some dimensions and others may have good values on some other dimensions. Thus, comprehensive learning (CL) strategy is used to generate a promising exemplar in the proposed algorithm.

In comprehensive learning (CL) strategy, the exploration-subpopulation group learns for different dimensions from its own members' previous best experiences and its particles have high level of exploration ability. The exploitation-subpopulation benefits from learning best experiences of all particles in the swarm including the whole swarm's best experience, therefore, its particles have strong exploitation ability. Different learning probability values are specified for each particle in the swarm such that the particles from the exploration-subpopulation are not influenced by the exploitation-subpopulation. In such way, the information sharing among the particle is controlled. The exploitation-subpopulation will be exploiting instantly the new good regions discovered by the exploration-subpopulation. In addition, adaptive control parameters are used in the subpopulation groups to enhance exploration and exploitation respectively. Therefore, this novel heterogeneous subpopulation structure is able to emphasize exploration and exploitation simultaneously without one process influences another.

3.2 Comprehensive Learning (CL) Strategy

In the proposed HCLPSO algorithm, the swarm population is divided into two subpopulations to curtail the adverse influence of exploration and exploitation on each other. The comprehensive learning (CL) strategy [78] is chosen among PSO learning strategies [79-82] to determine the exemplar for the particles in each subpopulation. In CLPSO algorithm, every particle in the swarm is able to learn from all other particles' best experiences instead of following the best individual alone. Besides, exploration and exploitation level of the particles can be specified through learning probability curve. Therefore, CL strategy is selected in the proposed algorithm to generate the exemplar for the particles in both exploration and exploitation subpopulation groups.

The flying direction of a particle is guided by its own previous best experience ($pbest_i^d$) and the whole swarm's best experience ($gbest^d$) in the original PSO. However, $gbest^d$ may be far from the global optimum and may represent an inferior local optimum of a multimodal problem. To resolve this situation, a comprehensive learning strategy was proposed in CLPSO algorithm [78]. In CLPSO, a particle's velocity is updated using all the particles' $pbests$. Each dimension of a particle learns from different particles' $pbests$ instead of learning from the same exemplar for all dimensions. This enhances the diversity of the population. The velocity of the i^{th} particle is updated with the following equation as in [78]:

$$V_i^d = wV_i^d + c * rand_i^d * (pbest_{f_i(d)}^d - X_i^d) \quad (3.1)$$

where, $f_i(d) = [f_i(1), f_i(2), \dots, f_i(D)]$ indicates if the i^{th} particle follows its own or other particle's previous best experience for each dimension d . D is the dimension of the solution search space. The i^{th} particle's position is updated using the equation (2.2). The exemplar for each dimension is decided according to learning probability Pc values. Different Pc values are used for different particles and explorative particles and exploitative particles are specified according to these learning probability values. The Pc value for each particle is calculated with the following equation [78]:

$$Pc_i = 0.05 + 0.45 * \frac{(\exp(\frac{10(i-1)}{ps-1}) - 1)}{(\exp(10) - 1)} \quad (3.2)$$

where, ' ps ' represents the population size. According to equation (3.2), each particle from first to N^{th} of particles has a learning probability value ranging from 0.05 to 0.5 [78]. To choose either its own or other's $pbest_i^d$ for each corresponding dimension of the i^{th} particle, a random number is generated for each dimension and compared with its learning probability Pc_i value. If random number is smaller than Pc_i value, the i^{th} particle is guided by other particle's $pbest_i^d$ position that is determined by tournament selection of size 2, i.e. two particles are selected randomly, and the particle with better fitness is chosen for the corresponding dimension. If random number is larger than Pc_i , the particle will follow its own $pbest$ position for that dimension. Therefore, the exemplar $pbest_{fi(d)}$ is a new position where each dimension learns from several particles' $pbest$ positions. In order to ensure that a particle's movement improves its $pbest$, a certain number of evaluations is defined as refreshing gap m in CLPSO and a new exemplar $pbest_{fi(d)}$ will be generated if there is no improvement for m (the refreshing gap) consecutive moves. The search range is also restricted in CLPSO with the bound $[X_{min}, X_{max}]$. If the updated position of the particle is out of the bound, its fitness value, and its $pbest$ are not updated.

3.3 Heterogeneous CLPSO with Enhanced Exploration and Exploitation

Exploration emphasizes on finding various potential solution regions of the entire search space and exploitation focuses on refining the promising solutions in the potential solution regions to attain the optimal solution. With the exploration and exploitation ability, the particles can search throughout the search space to find the global optimum. In CLPSO, each dimension of the particle learns from either its own best position or other particles' best position. The exemplar selection was decided by comparing a random number with the learning probability Pc value. With different Pc values, the particles have different level of exploration and exploitation ability. However, particles with high exploration tendency can be adversely influenced by particles with high exploitation tendency. To address this issue and compromise the exploration and exploitation, CLPSO is enhanced with exploration-subpopulation and exploitation-subpopulation group and the algorithm named heterogeneous CLPSO (HCLPSO) is proposed in this chapter.

In HCLPSO, the swarm is divided into two heterogeneous subpopulations. The first subpopulation is enhanced for exploration and the second subpopulation is enhanced for exploitation. In both exploration and exploitation subpopulations, the exemplar is generated using comprehensive learning (CL) strategy with the learning probability P_c curve as shown in Figure 3-1 below.

Figure 3-1 Learning Probability Curve

In CLPSO, each particle has a learning probability value between 0.05 and 0.5. Thus, the particle has 50% chance of guidance by other particles when a random number is compared to the learning probability value of each particle. For the proposed HCLPSO algorithm, the chance for selection has been tuned with 25% and 50% and the performance of the algorithm is better with 25% chance. Therefore, the learning probability values for the particles in the swarm are calculated using equation (3.2) with the range of learning probability values from 0.0 to 0.25. The velocity of the exploration-enhanced subpopulation is updated using equation (3.1) while the velocity of exploitation-enhanced subpopulation is updated using the following equation:

$$\begin{aligned}
 V_i^d = & w * V_i^d + c_1 * rand1_i^d * (pbest_{fi(d)}^d - X_i^d) \\
 & + c_2 * rand2_i^d * (gbest^d - X_i^d)
 \end{aligned} \tag{3.3}$$

The position of the whole population is updated using equation (2.2). To keep the balance between exploration and exploitation process, the adaptive control parameters (w ,

c , c_1 , and c_2) are used in the proposed HCLPSO algorithm. The inertia weight w is used as a linearly decreasing function of run time in the range of $0.99 \sim 0.2$ [13]. The acceleration coefficient c , c_1 and c_2 for the subpopulation group 1 and group 2 are tuned with different values from the literature and the results are shown in section 3.4. The HCLPSO achieved the best performance with $c = 3 \sim 1.5$, $c_1 = 2.5 \sim 0.5$ and $c_2 = 0.5 \sim 2.5$ [57]. Therefore, time varying acceleration coefficient $c = 3 \sim 1.5$ in equation (3.1) is used for subpopulation group 1 to enhance the exploration. For subpopulation group 2, the time varying acceleration coefficients $c_1 = 2.5 \sim 0.5$ and $c_2 = 0.5 \sim 2.5$ [57] are used in the equation (3.3) to enhance exploitation.

As shown in equations (3.1) and (3.3), all the particles in both subpopulations are guided by the exemplars obtained from comprehensive learning (CL) strategy. As described in Chapter 3.2, the random number is generated for each dimension of a particle and compared with its respective learning probability Pc_i value. If random number is smaller than Pc_i value, the particle will learn from another particle's $pbest$. The exemplar is determined with the tournament selection procedure in which the two particles are randomly selected from the subpopulation group 1 and the corresponding dimension will learn from the particle with better fitness. In the case of the random number is larger than Pc_i value, the corresponding dimension will learn from its own $pbest$. According to the learning probability Pc curve illustrated in Figure 3-1, the subpopulation group 1 has low learning probability values close to zero. If random number generated for each dimension of a particle were compared with low learning probability values, the particle in the subpopulation group 1 will learn mostly from its own $pbest$ for most of the dimensions. If all the exemplars of a particle were its own $pbest$, we will randomly choose one dimension to learn from other particle $pbest$'s corresponding dimension. Besides, time varying acceleration coefficient c , started at 3 and linearly reduced to 1.5, is also used to enhance the exploration ability of the particles. Therefore, the particles are explorative and the subpopulation group 1 has strong exploration ability.

Unlike the subpopulation group 1, the particles from the subpopulation group 2 learn not only from the exemplar generated by using CL strategy but also from the swarm best experience $gbest$ as described in equation (3.3). Thus, the subpopulation group 2 has high exploitative ability. If the particles are learning from their own $pbest$ s and $gbest$, it has higher chance of getting stuck into local optima. To avoid such condi-

tion, a particle will adopt other particles' $pbests$ (up to 25%) as shown in Figure 3-1. However, 25% probability is still low and the particle might end up in learning its own $pbest$. To ensure that the particle learns from other particles' $pbests$, at least one dimension will be adopted from other particles if the particle happens to learn from its own particle. For the acceleration coefficients from equation (3.3), c_1 is used in the range of $2.5 \sim 0.5$ to preserve the diversity satisfactorily in the early search stages. The acceleration coefficients c_2 is used in the range of $0.5 \sim 2.5$ to emphasize on increasing exploitation of the entire swarm's best experience. Thus, the particles are exploitative and the subpopulation group 2 has strong exploitation ability.

As mentioned beforehand, the swarm is composed of explorative particles and exploitative particles. The exploration and exploitation processes are performed by the first subpopulation and the second subpopulation, respectively. Since the explorative particles are not allowed to access the information of the exploitative particles, there is no information flow from the exploitation subpopulation group to the exploration subpopulation group. Thus, rapid information flow is avoided. Even if the exploitation group suffers premature convergence, the exploration group has the potential to rescue the exploitation-oriented group from the local optimum. Therefore, a compromise between exploration and exploitation is achieved in the proposed HCLPSO algorithm. As in CLPSO, if there is no improvement for refreshing gap m , a new exemplar ($pbest_{fi(d)}$) will be generated by learning from its own population itself for the subpopulation group 1 and by learning from the whole population for the subpopulation group 2.

Swarm diversity can be used to identify if populations are conducting exploration or exploitation [194]. Diversity of each subpopulation group and whole population is studied for one unimodal function (shifted sphere function) and one multimodal function (shifted rotated Griewank's function) on 30 dimensions. The diversity measure of the swarm is taken according to equation (3.4) and (3.5) [194]. The number of function evaluations of 300000 and population size of 40 are used in the diversity study. The diversity graphs from a single test run are shown in Figure 3-2 and 3-3. It can be clearly seen in diversity graphs that diversity of the exploration-subpopulation group is relatively higher than that of exploitation-subpopulation group and the diversity is preserved

even if exploitation-subpopulation group converges to the acceptable solution. The performance of proposed PSO algorithm is evaluated by comparing with other the-state-of-the-art PSO algorithms and the experimental results are discussed in the next section.

$$Diversity(S(t)) = \frac{1}{N} \sum_{i=1}^N \sqrt{\sum_{d=1}^D (X_i^d(t) - \overline{X^d(t)})^2} \quad (3.4)$$

$$\overline{X^d(t)} = \frac{\sum_{i=1}^N X_i^d(t)}{N} \quad (3.5)$$

where, S is the swarm and $N=|S|$ is the swarm size. D is the dimensionality of the problem. X_i^d is the d dimensional value of i^{th} particle in the population and $\overline{X^d(t)}$ is the average of d^{th} dimension over all the particles in the swarm.

Figure 3-2 Diversity comparisons between exploration-subpopulation, exploitation-subpopulation and the whole swarm on Unimodal Shifted Sphere Function

Figure 3-3 Diversity comparisons between exploration-subpopulation, exploitation-subpopulation and the whole swarm on Multimodal Shifted Rotated Griewank's Function

3.4 Experimental Set Up and Results

In this section, the performance of the proposed HCLPSO algorithm is evaluated on a large number of complex shifted rotated CEC2005 benchmark problems [195] and recent CEC2014 benchmark problems [196]. The test functions are single objective benchmark problems that can be transformed into dynamic, niching composition, computationally expensive and many other classes of problems. The main difference between CEC2005 benchmark functions and recent CEC2014 benchmark functions is the absence and presence of partial separability which presents in real world problems (such as image processing, circuit theory and transportation networks, etc.) [197]. The previous CEC2005 benchmark problems consisted of the separable and non-separable functions but excluded partially separable functions. Considering this, recent CEC2014 functions adopted a set of partially separable hybrid functions and it made the benchmark problems more closely emulate real world problems. Therefore, CEC2005 benchmark problems as well as recent CEC2014 problems are used to evaluate the performance and characteristic of the newly proposed HCLPSO algorithm on various problems. The evaluation of the proposed algorithm is performed according to the evaluation criteria of CEC2005 and CEC2014 benchmarking competitions and compared with other PSO algorithms such as:

- global version of PSO (PSO) [3]
- Fully-informed PSO (FIPS) [22]
- Unified PSO (UPSO) [23]
- Comprehensive Learning PSO (CLPSO) [78]
- Self-organizing hierarchical PSO with time varying acceleration coefficients (HPSO-TVAC) [57]
- Orthogonal Learning PSO (OLPSO) [79]
- Static Heterogeneous Swarm Optimization (sHPSO) [35]

The global version of PSO used linearly decreasing inertia weight to balance the exploration and exploitation ability in finding the global optimum. In FIPS algorithm, information of all neighbors was used to guide a particle and various topological structures of the population are tested to control the exploration and exploitation process. UPSO used both the swarm's best experience and the particle's neighbourhood best experience to adjust the exploration and exploitation tendencies. CLPSO used CL strategy that encouraged a particle to learn from different particles with different levels of exploration and exploitation ability. HPSO-TVAC introduced time varying acceleration coefficients to encourage the particles for stronger exploration in the early search and for stronger exploitation at the end of the search. In OLPSO, a particle is guided by an exemplar constructed from its personal best and the global best using an orthogonal learning strategy. In static heterogeneous particle swarm optimizer (sHPSO), different velocity updating rules of PSO [12], cognitive-only velocity update [19], social-only velocity update [19], barebones PSO [87] and modified barebones PSO [87] were used. The behaviors are randomly assigned to the particles and kept unchanged throughout the search process in sHPSO algorithm. All the algorithms are tested on both sets of CEC2005 and recent CEC2014 benchmark functions. The detail parameter settings of the proposed HCLPSO and other algorithms are presented in Table 3-1 below and refreshing gap m is set at five for all the experiments of the proposed HCLPSO and CLPSO algorithms.

Table 3-1 Parameter Settings

Algorithm	Inertia weight	Constriction Coefficients	Acceleration Coefficients	Population size		Function Evaluations (FES)		Reference
	w	χ	c_1, c_2, c	D=10	D=30	D=10	D=30	
PSO	0.9 ~ 0.4	-	$c_1 = 2, c_2 = 2$	20	40	100000	300000	[13]
FIPS	-	0.729	$c = 2$	20	40	100000	300000	[31]
UPSO	-	0.729	$c = 1.49445$	20	40	100000	300000	[23]
CLPSO	0.9 ~ 0.4	-	$c = 1.49445$	20	40	100000	300000	[78]
OLPSO	0.9 ~ 0.4	-	$c = 2$	20	40	100000	300000	[79]
HPSO-TVAC	-	-	$c_1 = 2.5 \sim 0.5$ $c_2 = 0.5 \sim 2.5$	20	40	100000	300000	[57]
sHPSO	0.72	-	$c_1 = 2.5 \sim 0.5$ $c_2 = 0.5 \sim 2.5$	20	40	100000	300000	[35]
HCLPSO	0.99 ~ 0.2	-	$c_1 = 2.5 \sim 0.5$ $c_2 = 0.5 \sim 2.5$ $c = 3 \sim 1.5$	20	40	100000	300000	

3.4.1 Parameter Tuning

In the proposed PSO algorithm, there are four main parameters to be adjusted for balancing exploration and exploitation between two subpopulation groups. The first parameter to tune is subpopulation group size and the next parameters to adjust are the acceleration coefficients, c from exploration-subpopulation velocity update equation (3.1), and c_1 and c_2 from exploitation-subpopulation velocity update equation (3.3). The parameters are tuned using the first 14 CEC2005 test functions with the same parameter settings of run time 30, the population size 40, and the number of function evaluations (FES) 300000 and dimension 30.

The experiment results of tuning subpopulation group sizes are shown in Table 3-2. The results are ranked based on average error mean and standard deviation values and summarized with final rank resulted from averaging over the ranks. According to the final rank, the subpopulation size of ($g_1 = 5$, $g_2 = 35$) obtained the best performance for all 14 benchmark functions compared to other subpopulation settings. However, as presented in Table 3-2, this parameter setting provides unsatisfactory performance on test functions F_6 and F_9 as the second best parameter setting ($g_1 = 10$, $g_2 = 30$) does functions F_6 and F_9 . On the other hand, the parameter setting ($g_1 = 15$, $g_2 = 25$) provides the satisfactory and consistent performance throughout 14 problems.

For acceleration coefficient c from exploration-subpopulation velocity update equation (3.1), c is used as a constant value of 1.49445 in [78]. In the proposed HCLPSO algorithm, c is calibrated using constant and time varying values and its calibration results are presented in Table 3-3. For c_1 and c_2 from exploitation-subpopulation velocity update equation (3.3), time varying values of c_1 started at 2.5, linearly reduced to 0.5 and c_2 started at 0.5, linearly increased to 2.5 from literature [57] are used. In order to balance the search between exploration-subpopulation and exploitation-subpopulation groups, different c values are calibrated by pairing with $c_1 = 2.5 \sim 0.5$, $c_2 = 0.5 \sim 2.5$. In calibration Table 3-3, $c = 3$ provides the best performance according to final rank value. However, it performs poorly on function F_6 and F_9 as underlined. Among three different settings, the second best parameter setting $c = 3 \sim 1.5$ provides superior performance consistently for all 14 test functions.

Therefore, the subpopulation sizes ($g_1=15$, $g_2=25$) and time varying acceleration coefficients ($c_1= 0.5\sim 2.5$, $c_2=2.5$ to 0.5 and $c = 3 \sim 1.5$) are selected for further performance evaluation of the proposed HCLPSO algorithm.

Table 3-2 Calibrating subpopulation sizes (g_1 & g_2) for HCLPSO algorithm

F	Criteria	5+35	10+30	15+25	20+20	25+15	30+10	35+5
F_1	mean	0	0	0	0	0	0	0
	std	0	0	0	0	0	0	0
	rank	1	1	1	1	1	1	1
F_2	mean	8.69E-10	2.09E-08	1.70E-06	2.34E-04	1.82E-02	0.35	3.61
	std	1.35E-09	3.97E-08	1.71E-06	3.67E-04	1.48E-02	0.41	4.77
	rank	1	2	3	4	5	6	7
F_3	mean	5.37E+05	5.89E+05	6.42E+05	8.78E+05	1.41E+06	2.39E+06	3.09E+06
	std	1.73E+05	1.86E+05	2.61E+05	3.56E+05	6.14E+05	9.61E+05	1.39E+06
	rank	1	2	3	4	5	6	7
F_4	mean	2.20E+02	4.52E+02	5.22E+02	6.35E+02	8.24E+02	1.57E+03	2.43E+03
	std	1.21E+02	2.60E+02	3.09E+02	4.82E+02	3.86E+02	7.59E+02	9.70E+02
	rank	1	2	3	4	5	6	7
F_5	mean	2.96E+03	3.04E+03	2.97E+03	3.14E+03	3.38E+03	3.39E+03	3.49E+03
	std	5.81E+02	6.08E+02	4.55E+02	4.20E+02	6.49E+02	4.87E+02	4.80E+02
	rank	1	3	2	4	5	6	7
F_6	mean	10.28	3.22	2.39	2.33	0.72	0.31	0.43
	std	15.29	3.53	4.27	3.76	0.97	0.58	0.64
	rank	7	6	5	4	3	1	2
F_7	mean	0.01	0.02	0.02	0.02	0.02	0.02	0.02
	std	0.01	0.02	0.02	0.02	0.02	0.02	0.02
	rank	1	2	2	2	2	2	2
F_8	mean	20.82	20.86	20.87	20.91	20.91	20.91	20.93
	std	0.09	0.07	0.09	0.06	0.06	0.09	0.05
	rank	1	2	2	4	5	6	7
F_9	mean	1.19	0.10	0	0	0	0	0
	std	1.12	4.01E-01	0	0	0	0	0
	rank	3	2	1	1	1	1	1
F_{10}	mean	59.33	60.84	56.08	55.34	58.35	68.88	74.53
	std	16.96	14.08	12.90	12.13	12.73	17.13	15.88
	rank	4	5	2	1	3	6	7
F_{11}	mean	19.47	20.15	20.32	20.04	20.90	21.29	22.77
	std	4.16	3.86	2.94	2.86	2.54	2.15	2.00
	rank	1	3	5	2	7	4	6
F_{12}	mean	3.87E+03	3.20E+03	3.91E+03	3.58E+03	4.92E+03	6.74E+03	9.75E+03
	std	3.24E+03	3.43E+03	3.69E+03	3.50E+03	3.75E+03	3.42E+03	4.75E+03
	rank	3	1	3	2	4	5	6
F_{13}	mean	1.74	1.54	1.45	1.65	1.72	2.00	2.14
	std	0.28	0.28	0.28	0.32	0.35	0.37	0.26
	rank	5	2	1	3	4	6	7

	mean	11.58	11.86	11.93	11.95	12.01	12.16	12.29
F_{14}	std	0.57	0.46	0.58	0.45	0.54	0.44	0.44
	rank	1	2	3	4	5	6	7
Average rank		2.21	2.50	2.57	2.86	3.93	4.43	5.29
Final rank		1	2	3	4	5	6	7

Table 3-3 Calibrating acceleration coefficients (c_1 and c_2) for HCLPSO algorithm

Functions	Criteria	$c_1=2.5 \sim 0.5$	$c_1=2.5 \sim 0.5$	$c_1=2.5 \sim 0.5$
		$c_2=0.5 \sim 2.5$	$c_2=0.5 \sim 2.5$	$c_2=0.5 \sim 2.5$
		$c=1.5$	$c=3 \text{ to } 1.5$	$c=3$
F_1	mean	0	0	0
	std	0	0	0
	rank	1	1	1
F_2	mean	8.60E-06	1.70E-06	2.27E-08
	std	1.36E-05	1.71E-06	3.89E-08
	rank	3	2	1
F_3	mean	8.77E+05	6.42E+05	5.42E+05
	std	4.85E+05	2.61E+05	2.41E+05
	rank	3	2	1
F_4	mean	4.57E+02	5.22E+02	2.46E+02
	std	2.68E+02	3.09E+02	1.30E+02
	rank	2	3	1
F_5	mean	3.11E+03	2.97E+03	2.96E+03
	std	4.97E+02	4.55E+02	5.49E+02
	rank	3	2	1
F_6	mean	2.86	2.39	10.72
	std	6.54	4.27	14.62
	rank	2	1	3
F_7	mean	0.02	0.02	0.02
	std	0.01	0.02	0.02
	rank	1	2	2
F_8	mean	20.87	20.87	20.87
	std	0.08	0.09	0.09
	rank	1	2	2
F_9	mean	0.10	0	0.07
	std	0.40	0	2.52E-01
	rank	3	1	2
F_{10}	mean	66.37	56.08	58.10
	std	24.35	12.90	17.18
	rank	3	1	2
F_{11}	mean	22.15	20.32	19.96
	std	2.34	2.94	3.19
	rank	3	2	1
F_{12}	mean	5.65E+03	3.91E+03	4.43E+03
	std	6.31E+03	3.69E+03	3.94E+03
	rank	2	1	1

F_{13}	mean	1.59	1.45	1.60
	std	0.24	0.28	0.31
	rank	2	1	3
F_{14}	mean	11.97	11.93	11.78
	std	0.60	0.58	0.55
	rank	3	2	1
Average rank		2.29	1.64	1.57
Final rank		3	2	1

3.4.2 Comparative study with PSO variants

In this section, the performance of proposed HCLPSO algorithm is compared with other state-of-the-art PSO algorithms. Their performances are evaluated following the evaluation criteria from CEC2005 and CEC2014 benchmark competition. Firstly, the performance of each algorithm is measured and ranked in terms of error mean and standard deviation values of their solutions. The final rank for each algorithm is defined according to their average rank values over the benchmark problems.

Secondly, the statistical difference between the results of proposed HCLPSO and the results of the other PSO variants is examined using non-parametric Wilcoxon signed-rank test [198-200] with the significance level of 5%. In this single-problem analysis, pairwise comparison is performed over the results obtained over 30 simulation runs on 10 and 30 dimensional problems. The symbol (+) represents that HCLPSO performs significantly better than the compared algorithm, the symbol (0) represents that there is no significant difference between HCLPSO and the compared algorithm and the symbol (-) represents that the compared algorithm performs significantly better than HCLPSO algorithm.

In addition, the convergence progress of all the algorithms is also analysed for all benchmark test functions. As specified in CEC2005 benchmark competition [195], the median performance of 30 runs is used to analyse the convergence performance of the proposed HCLPSO algorithm and other PSO variants. As the convergence characteristic graphs of 10 D and 30 D problems are similar to each other, the convergence graphs of only 30 D benchmark problems are presented for two benchmark sets in this thesis.

3.4.2.1 Comparative study on CEC2005 benchmark set

The CEC2005 benchmark set includes 25 test functions and the test suit is composed of unimodal, multimodal, expanded and hybrid composite functions [195]. Function $F_1 \sim F_5$ are unimodal. Functions $F_6 \sim F_{12}$ are multimodal problems. F_{13} and F_{14} are expanded functions. Finally, $F_{15} \sim F_{25}$ are hybrid composition of several test functions. The experiments are conducted on 10 and 30 dimensional problems and run 30 times for each dimension. For 10 dimensional problems, the performance of the proposed HCLPSO is compared with PSO, FIPS, UPSO, HPSO-TVAC, CLPSO and sHPSO algorithms. For 30 dimensional problems, HCLPSO algorithm is compared with those PSO variants as well as OLPSO algorithm. The 30 dimensional simulation results of OLPSO algorithm were provided by OLPSO authors. The 30 dimensional experiments for the proposed HCLPSO algorithm and other PSO variants are conducted using the same number of function evaluations, population size and run time of OLPSO algorithm. The convergence characteristic graphs of 30 dimensional CEC2005 benchmark functions are shown in Appendix 2.

A. Results for 10 Dimensional Problems

The experiments of 10 dimensional CEC2005 benchmark functions are performed using population size 20 and run 30 times for each function. The maximum number of function evaluations for 10 dimensions is set at 100,000 in each run. The 10 dimensional experimental results of the proposed HCLPSO and other PSO variants are illustrated in Table 3-4. The exploration-subpopulation group size $g_1 = 8$ and exploitation-subpopulation group size $g_2 = 12$ are employed in the experiment of 10 dimension problems. As shown in comparison Table 3-4, on 10 dimensional unimodal problems, all PSO algorithms perform best on function F_1 and F_2 . HPSO-TVAC provides best solution on function F_3 while the PSO and FIPS does best on function F_4 . The CLPSO yields the best solution on function F_5 . The proposed HCLPSO algorithm performs second best and ranked second on those three unimodal functions F_3 , F_4 , and F_5 .

On multimodal, expanded and hybrid composite functions, HCLPSO algorithm outperforms other algorithms on function $F_6, F_7, F_{11}, F_{12}, F_{14} \sim F_{18}, F_{21} \sim F_{23}, F_{25}$ i.e. 13 out of 20 test functions. CLPSO also obtains best performance on function $F_9, F_{10}, F_{13}, F_{19}$ and F_{24} i.e. 5 out of 20 functions. The sHPSO performs best on function F_8 . The

PSO offers best solution on function F_{20} . The number of (Best/2nd Best/Worst) is counted for each algorithm and worst ranks are underlined in Table 3-4. It can be observed that there is no inferior performance on any benchmark functions for the proposed HCLPSO algorithm.

Overall, the HCLPSO algorithm provides best performance on 15 out of 25 benchmark problems and is ranked first over other PSO algorithms. The 10 dimensional Wilcoxon signed rank test results of HCLPSO algorithm against other algorithms are shown in Table 3-5. The number of (+/0/-) are presented in last row of Wilcoxon table and it can be observed that HCLPSO performs significantly better than other PSO algorithms.

Table 3-4 Comparison of HCLPSO vs other PSOs on CEC2005 10 dimensional problems

F	Criteria	HCLPSO	PSO	FIPS	UPSO	CLPSO	HPSO-TVAC	sHPSO
F_1	mean	0	0	0	0	0	0	0
	std	0	0	0	0	0	0	0
	rank	1	1	1	1	1	1	1
F_2	mean	0	0	0	0	1.86E-02	0	0
	std	0	0	0	0	2.24E-02	0	0
	rank	1	1	1	1	2	1	1
F_3	mean	6.83E+04	1.60E+05	2.24E+05	1.05E+05	3.49E+05	6.36E+04	8.82E+04
	std	4.63E+04	1.10E+05	1.22E+05	1.16E+05	2.21E+05	3.81E+04	6.48E+04
	rank	2	5	6	4	7	1	3
F_4	mean	8.59E-03	0	0	1.47E+02	5.20	2.15E+02	2.31E+03
	std	1.29E-02	0	0	3.38E+02	11.02	2.96E+02	2.23E+03
	rank	2	1	1	4	3	5	6
F_5	mean	7.51	1.04E+03	1.32E+02	6.03E+02	6.95	4.96E+02	1.27E+03
	std	8.53	2.07E+02	1.11E+02	5.34E+02	16.01	3.61E+02	2.10E+03
	rank	2	6	3	5	1	4	7
F_6	mean	0.81	3.47	4.70	18.50	0.86	11.83	786.51
	std	1.17	7.76	11.11	87.60	1.94	22.29	4017.41
	rank	1	3	4	6	2	5	7
F_7	mean	0.11	0.22	0.22	0.19	0.21	2.56	11.58
	std	0.06	0.14	0.13	0.15	0.11	1.84	15.61
	rank	1	5	4	2	3	6	7
F_8	mean	20.21	20.29	20.36	20.36	20.27	20.17	20.14
	std	0.07	0.09	0.08	0.06	0.05	0.11	0.12
	rank	3	4	7	6	5	2	1
F_9	mean	0.10	3.55	0.23	7.89	0	1.76	18.67
	std	0.30	1.40	0.57	4.26	0	1.47	15.16
	rank	2	5	3	6	1	4	7

F_{10}	mean	9.43	14.36	14.27	15.95	9.24	34.76	34.22
	std	3.04	5.21	6.40	5.75	2.97	11.37	12.14
	rank	2	4	3	5	1	7	6
F_{11}	mean	3.29	3.36	4.55	5.98	4.53	5.91	6.92
	std	1.27	1.42	1.61	0.94	0.78	1.19	1.69
	rank	1	2	4	7	3	5	6
F_{12}	mean	1.38E+01	1.83E+03	2.61E+02	7.78E+01	7.22E+01	5.70E+02	1.43E+03
	std	2.62E+01	4.31E+03	3.62E+02	2.71E+02	5.20E+01	7.52E+02	3.21E+03
	rank	1	7	4	3	2	5	6
F_{13}	mean	0.29	0.63	1.17	0.81	0.28	0.66	1.61
	std	0.06	0.21	0.25	0.23	0.06	0.22	1.00
	rank	2	3	6	5	1	4	7
F_{14}	mean	2.70	2.86	2.93	3.30	3.00	3.20	3.60
	std	0.45	0.41	0.34	0.33	0.27	0.35	0.41
	rank	1	2	3	6	4	5	7
F_{15}	mean	2.51E+00	2.34E+02	2.08E+02	2.63E+02	4.81E+00	2.15E+02	3.82E+02
	std	1.38E+01	1.76E+02	1.76E+02	1.07E+02	1.46E+01	2.04E+02	1.93E+02
	rank	1	5	3	6	2	4	7
F_{16}	mean	1.08E+02	1.29E+02	1.12E+02	1.35E+02	1.20E+02	1.91E+02	1.87E+02
	std	1.63E+01	1.64E+01	9.91E+00	2.45E+01	9.92E+00	3.91E+01	4.58E+01
	rank	1	4	2	5	3	7	6
F_{17}	mean	1.13E+02	1.42E+02	1.24E+02	1.49E+02	1.25E+02	1.95E+02	2.05E+02
	std	1.36E+01	6.81E+01	1.46E+01	1.86E+01	1.29E+01	4.47E+01	7.31E+01
	rank	1	4	2	5	3	6	7
F_{18}	mean	6.80E+02	7.10E+02	8.06E+02	8.97E+02	6.85E+02	1.01E+03	9.79E+02
	std	2.16E+02	2.53E+02	1.34E+02	1.20E+02	1.87E+02	8.67E+01	7.28E+01
	rank	1	3	4	5	2	7	6
F_{19}	mean	6.76E+02	6.99E+02	7.59E+02	8.62E+02	6.54E+02	9.95E+02	9.89E+02
	std	2.03E+02	2.45E+02	1.69E+02	1.31E+02	1.88E+02	9.26E+01	1.42E+02
	rank	2	3	4	5	1	7	6
F_{20}	mean	6.24E+02	6.16E+02	7.81E+02	8.89E+02	7.46E+02	9.86E+02	1.01E+03
	std	2.38E+02	2.66E+02	1.35E+02	9.67E+01	1.62E+02	1.12E+02	8.42E+01
	rank	2	1	3	5	4	6	7
F_{21}	mean	3.93E+02	4.51E+02	7.65E+02	8.23E+02	4.52E+02	1.04E+03	1.09E+03
	std	1.41E+02	1.93E+02	2.84E+02	3.22E+02	1.40E+02	2.83E+02	2.23E+02
	rank	1	2	4	5	3	6	7
F_{22}	mean	7.24E+02	7.64E+02	7.98E+02	8.12E+02	7.27E+02	8.86E+02	8.86E+02
	std	1.15E+02	9.30E+01	6.13E+01	7.19E+01	1.42E+02	1.30E+02	1.06E+02
	rank	1	3	4	5	2	7	6
F_{23}	mean	5.58E+02	6.77E+02	8.57E+02	1.01E+03	5.58E+02	1.12E+03	1.17E+03
	std	5.55E+01	1.43E+02	2.66E+02	2.38E+02	7.16E+01	2.63E+02	1.69E+02
	rank	1	3	4	5	2	6	7
F_{24}	mean	2.10E+02	2.80E+02	3.77E+02	4.20E+02	2.00E+02	8.23E+02	6.62E+02
	std	5.48E+01	1.56E+02	3.90E+00	1.11E+02	1.63E-08	4.59E+02	4.56E+02
	rank	2	3	4	5	1	7	6

	mean	2.00E+02	3.10E+02	3.77E+02	4.46E+02	2.00E+02	7.87E+02	7.38E+02
F_{25}	std	1.21E-12	1.75E+02	2.33E+00	1.95E+02	2.59E-09	4.37E+02	4.85E+02
	rank	1	3	4	5	2	7	6
	Average rank	1.44	3.32	3.52	4.68	2.44	5.00	5.72
	Final rank	1	3	4	5	2	6	7
	Best/2nd Best/Worst	15/9/0	4/3/1	3/2/1	2/1/1	7/7/2	3/1/7	3/0/12
	Algorithms	HCLPSO	PSO	FIPS	UPSO	CLPSO	HPSO-TVAC	sHPSO

Table 3-5 Wilcoxon Signed Rank Test Results of Single-Problem Analysis with a Significance Level of $\alpha = 0.05$ for CEC2005 10 dimensional problems

Functions	Pairwise Comparison HCLPSO versus					
	PSO	FIPS	UPSO	CLPSO	HPSO-TVAC	sHPSO
F_1	+	+	+	0	+	+
F_2	+	+	+	0	+	+
F_3	+	+	+	0	+	+
F_4	0	+	+	0	+	+
F_5	+	+	+	+	+	+
F_6	+	+	+	0	+	+
F_7	+	+	+	+	+	+
F_8	0	+	+	+	+	+
F_9	+	+	+	+	+	+
F_{10}	+	+	+	+	+	+
F_{11}	+	+	+	0	+	+
F_{12}	+	+	+	0	+	+
F_{13}	+	+	+	0	+	+
F_{14}	0	+	+	0	+	+
F_{15}	0	+	+	0	+	+
F_{16}	+	+	+	+	+	+
F_{17}	+	+	+	0	+	+
F_{18}	+	+	+	0	+	+
F_{19}	+	+	+	+	+	+
F_{20}	+	+	+	+	+	+
F_{21}	+	+	+	+	+	+
F_{22}	+	+	+	+	+	+
F_{23}	0	0	+	0	+	+
F_{24}	+	+	+	+	+	+
F_{25}	+	+	+	+	+	+
+ / 0 / -	20 / 5 / 0	24 / 1 / 0	25 / 0 / 0	12 / 13 / 0	25 / 0 / 0	25 / 0 / 0
	PSO	FIPS	UPSO	CLPSO	HPSO-TVAC	sHPSO

B. Results for 30 Dimensional Problems

For 30 dimensional CEC2005 benchmark problems, we conducted the experiments using a population size of 40 and 30 repetitions. The maximum number of func-

tion evaluations for 30 dimensions is set at 300,000 in each run for each function. The comparison between the 30 dimensional results of the proposed HCLPSO and other PSO variants including OLPSO algorithm is presented in Table 3-6.

As described in comparison Table 3-6, all algorithms perform well on the shifted sphere function F_1 . UPSO obtains the best performance on function F_2 . HCLPSO achieves the best solution on function F_3 . PSO performs best on function F_4 and FIPS provides best performance on function F_5 . The proposed HCLPSO algorithm consistently performed well and obtained the best and the second best performance on all unimodal functions F_1 to F_5 . Overall, it provides outstanding and robust performance over other PSO algorithms. Among unimodal functions, the convergence graph of two unimodal functions, F_4 and F_5 , are presented in Figure 3-4. It can be clearly observed that the proposed HCLPSO obtains better result than other algorithms.

Table 3-6 Comparison of HCLPSO vs other PSOs on CEC2005 30 dimensional problems

F	Criteria	HCLPSO	PSO	FIPS	UPSO	CLPSO	OLPSO	HPSO-TVAC	sHPSO
F_1	mean	0	0	0	0	0	0	0	0
	std	0	0	0	0	0	0	0	0
	rank	1	1	1	1	1	1	1	1
F_2	mean	1.70E-06	0.37	77.94	2.65E-07	1.14E+03	13.79	3.79E-06	1.44E-02
	std	1.71E-06	0.32	27.05	2.42E-07	2.53E+02	8.33	2.82E-06	7.10E-02
	rank	2	5	7	1	8	6	3	4
F_3	mean	6.42E+05	6.53E+06	2.45E+07	1.54E+06	1.22E+07	1.60E+07	7.72E+05	8.75E+05
	std	2.61E+05	4.17E+06	6.29E+06	4.75E+05	3.34E+06	7.04E+06	2.96E+05	5.34E+05
	rank	1	5	8	4	6	7	2	3
F_4	mean	5.22E+02	3.81E+02	1.15E+03	7.28E+03	8.77E+03	2.18E+03	2.48E+04	2.02E+04
	std	3.09E+02	3.31E+02	3.73E+02	2.79E+03	1.85E+03	1.09E+03	5.71E+03	9.94E+03
	rank	2	1	3	5	6	4	8	7
F_5	mean	2.97E+03	3.85E+03	2.22E+03	6.32E+03	4.47E+03	3.30E+03	9.20E+03	6.94E+03
	std	4.55E+02	8.00E+02	5.14E+02	1.63E+03	4.26E+02	3.75E+02	1.81E+03	1.43E+03
	rank	2	4	1	6	5	3	8	7
F_6	mean	2.39	70.16	37.70	68.20	2.39	20.68	50.41	115.27
	std	4.27	95.09	35.03	96.41	3.84	24.97	50.54	228.56
	rank	1	6	3	5	1	2	4	7
F_7	mean	0.02	0.76	0.03	0.02	0.70	0.01	0.01	0.04
	std	0.02	1.41	0.02	0.01	0.15	0.01	0.01	0.04
	rank	2	6	3	2	5	1	1	4
F_8	mean	20.87	20.90	20.94	20.95	20.92	20.96	20.71	20.18
	std	0.09	0.07	0.06	0.05	0.06	0.08	0.15	0.19
	rank	3	4	6	7	5	8	2	1

F_9	mean	0	19.00	57.11	85.16	0	0	10.71	82.54
	std	0	5.37	14.55	16.90	0	0	4.96	24.35
	rank	1	3	4	6	1	1	2	5
F_{10}	mean	56.08	100.32	177.86	93.63	99.78	109.23	275.73	242.83
	std	12.90	58.63	9.25	18.82	12.48	18.94	45.49	89.17
	rank	1	4	6	2	3	5	8	7
F_{11}	mean	20.32	23.18	38.36	30.94	24.42	25.05	30.62	32.29
	std	2.94	2.81	1.52	1.96	1.78	3.14	2.58	3.84
	rank	1	2	8	6	3	4	5	7
F_{12}	mean	3.91E+03	5.07E+04	5.62E+04	4.31E+03	1.47E+04	1.22E+04	4.47E+03	2.43E+04
	std	3.69E+03	3.50E+04	2.00E+04	3.44E+03	3.45E+03	5.42E+03	4.66E+03	2.66E+04
	rank	1	7	8	2	5	4	3	6
F_{13}	mean	1.45	3.02	12.51	5.78	1.86	1.86	3.85	6.14
	std	0.28	0.69	0.96	1.59	0.18	0.28	1.11	2.31
	rank	1	4	8	6	2	3	5	7
F_{14}	mean	11.93	12.65	13.12	12.79	12.64	13.13	12.31	13.07
	std	0.58	0.43	0.21	0.33	0.22	0.20	0.37	0.39
	rank	1	4	7	5	3	8	2	6
F_{15}	mean	88.04	299.71	283.02	465.59	40.10	244.88	321.29	457.16
	std	113.02	107.62	39.65	70.22	37.90	84.09	107.12	138.94
	rank	2	5	4	8	1	3	6	7
F_{16}	mean	104.07	235.84	201.66	199.42	163.61	135.54	336.21	376.70
	std	36.41	152.98	14.63	61.91	27.87	55.48	97.20	99.46
	rank	1	6	5	4	3	2	7	8
F_{17}	mean	109.59	262.63	224.39	278.90	219.31	204.56	391.68	367.25
	std	34.01	153.27	10.71	79.14	40.09	46.67	97.34	133.86
	rank	1	5	4	6	3	2	8	7
F_{18}	mean	894.42	927.77	832.24	840.92	907.76	906.63	908.24	997.99
	std	43.04	2.86	1.78	6.18	26.75	20.23	79.00	60.04
	rank	3	7	1	2	5	4	6	8
F_{19}	mean	913.49	926.69	831.09	841.77	915.09	906.73	912.45	982.94
	std	2.39	2.49	1.50	7.42	1.36	20.25	79.78	78.69
	rank	5	7	1	2	6	3	4	8
F_{20}	mean	914.03	927.06	831.86	841.62	911.24	906.40	905.92	974.13
	std	2.36	2.35	1.66	6.22	20.60	20.19	74.72	67.96
	rank	6	7	1	2	5	4	3	8
F_{21}	mean	500.00	510.00	862.53	761.78	500.00	500.00	1173.50	996.01
	std	0	54.77	1.03	174.96	0	0	229.74	328.97
	rank	1	2	4	3	1	1	6	5
F_{22}	mean	910.68	938.66	521.17	615.37	974.32	941.70	1216.89	1123.41
	std	15.75	12.96	1.27	113.50	17.22	16.30	73.68	103.68
	rank	3	4	1	2	6	5	8	7
F_{23}	mean	534.16	574.17	865.57	816.07	534.16	534.16	1258.30	996.79
	std	4.07E-04	122.10	1.03	147.72	1.53E-04	5.04E-04	34.57	323.48
	rank	1	2	4	3	1	1	6	5

F_{24}	mean	200.00	200.00	214.39	225.24	200.00	200.00	1276.11	995.63
	std	0	0	0.58	4.87	0	0	205.10	409.92
	rank	1	1	2	3	1	1	5	4
F_{25}	mean	200.00	1113.45	214.21	225.56	200.00	207.46	1331.03	1037.43
	std	0	14.41	0.50	7.19	0	7.09	26.00	423.59
	rank	1	6	3	4	1	2	7	5
Average rank		1.8	4.32	4.12	3.88	3.48	3.4	4.8	5.76
Final rank		1	6	5	4	3	2	7	8
Best/2 nd		15/5/0	3/3/1	5/1/4	1/7/2	8/1/1	6/4/2	2/4/9	2/0/5
Best/Worst									
Algorithms		HCLPSO	PSO	FIPS	UPSO	CLPSO	OLPSO	HPSO-TVAC	sHPSO

Table 3-7 Wilcoxon Signed Rank Test Results of Single-Problem Analysis with a Significance Level of $\alpha = 0.05$ for CEC2005 30 dimensional problems

Functions	Pairwise Comparison HCLPSO versus						
	PSO	FIPS	UPSO	CLPSO	OLPSO	HPSO-TVAC	sHPSO
F_1	0	0	0	0	0	0	0
F_2	+	+	−	+	+	+	0
F_3	+	+	+	+	+	0	0
F_4	0	+	+	+	+	+	+
F_5	+	−	+	+	+	+	+
F_6	+	+	+	0	+	+	+
F_7	+	+	0	+	0	0	0
F_8	0	+	+	+	+	−	+
F_9	+	+	+	0	0	+	+
F_{10}	+	+	+	+	+	+	+
F_{11}	+	+	+	+	+	+	+
F_{12}	+	+	+	+	+	+	+
F_{13}	+	+	+	+	+	+	+
F_{14}	+	+	+	+	+	+	+
F_{15}	+	+	+	0	+	+	+
F_{16}	+	+	+	+	0	+	+
F_{17}	+	+	+	+	+	+	+
F_{18}	+	−	−	+	0	0	+
F_{19}	+	−	−	+	−	0	+
F_{20}	+	−	−	0	−	−	+
F_{21}	+	+	+	0	0	+	+
F_{22}	+	−	−	+	+	+	+
F_{23}	+	+	+	−	0	+	+
F_{24}	0	+	+	0	0	+	+
F_{25}	+	+	+	+	+	+	+
+/0/−	21/4/0	19/1/5	18/2/5	17/7/1	15/8/2	18/5/2	21/4/0
	PSO	FIPS	UPSO	CLPSO	OLPSO	HPSO-TVAC	sHPSO

In multimodal functions, HCLPSO performs well on all the shifted and rotated multimodal functions. The proposed HCLPSO provides best performance on multimodal F_6 , F_9 , F_{10} , F_{11} , F_{12} , and second best on multimodal F_7 and F_8 . On the other hand, CLPSO yields the best solution on F_6 while OLPSO, HPSO-TVAC perform best on function F_7 . The sHPSO achieves the best performance on function F_8 . The HPSO-TVAC offers the best solution on F_{12} .

It can be observed that the HCLPSO is consistently performing well throughout multimodal problems and obtains the best overall performance. Among multimodal functions, the convergence graphs of two multimodal functions, F_{10} and F_{11} , are presented in Figure 3-5 (a) and 3-5 (b) respectively. In both cases, the proposed HCLPSO algorithm is able to jump out of local minimum and achieves the best performance. In expanded hybrid composite test functions, HCLPSO performs best on both expanded extended function (F_{13}) and shifted rotated expanded function (F_{14}) as shown in Figure 3-6.

Figure 3-4 Convergence characteristic graphs of CEC2005 30 dimensional unimodal 30 functions: (a) F_4 : Shifted Schwefel's Problem 1.2 with Noise in Fitness and (b) F_5 : Schwefel's Problem 2.6 with Global Optimum on Bounds

Figure 3-5 Convergence characteristic graphs of CEC2005 30 dimensional multimodal functions: (a) F_{10} : Shifted Rotated Rastrigin's Function and (b) F_{11} : Shifted Rotated Weierstrass Function

In hybrid composite test functions, the HCLPSO yields best performance on hybrid composition function of F_{16} , F_{17} , F_{21} , F_{23} , F_{24} and F_{25} . On the other hand, the FIPS gives the best results on function F_{18} , F_{19} , F_{20} and F_{22} . Similarly, OLPSO performs best on F_{21} , F_{23} , and F_{24} and CLPSO does best on F_{15} , F_{21} , F_{23} , F_{24} and F_{25} . However, the HCLPSO performs consistently within top three ranges throughout 11 problems except for functions F_{19} and F_{20} .

According to final rank and the number of Best/2nd Best/Worst in Table 3-7, proposed HCLPSO ranked first by performing best 15 out of 25 functions and achieved comparable performance on other unimodal, multimodal, expanded and hybrid composite functions. The last row in Wilcoxon sign rank test Table 3-7 also shows that HCLPSO performs better than other PSO algorithms on most of the 30 dimensional problems.

Among hybrid composite functions, the convergence graphs of F_{17} and F_{25} are presented in Figure 3-7(a) and 3-7(b) respectively. Figure 3-7(a) shows that the proposed HCLPSO achieves the outstanding performance. However, the HCLPSO obtains the comparable performance to other algorithms in F_{25} as shown in Figure 3-7(b). In summary, the convergence analysis shows that the proposed HCLPSO algorithm outperforms other PSO variants in most test cases.

Figure 3-6 Convergence characteristic graphs of CEC2005 30 dimensional expanded extended functions: (a) F_{13} : Expanded Extended Griewank's plus Rosenbrock's Function (F_8F_2) and (b) F_{14} : Shifted Rotated Expanded Scaffer's F_6

Figure 3-7 Convergence characteristic graphs of CEC2005 30 dimensional composition functions: (a) F_{17} : Rotated Hybrid Composition Function with Noise in Fitness and (b) F_{25} : Rotated Hybrid Composition Function without Bounds

3.4.2.2 Comparative study on CEC2014 Benchmark Set

The CEC2014 benchmark set consists of 30 benchmark functions [196]. The 30 benchmark functions can be divided into four groups: unimodal functions ($F_1 \sim F_3$), multimodal functions including expanded extended functions ($F_4 \sim F_{16}$), partially separable hybrid functions ($F_{17} \sim F_{22}$) and composition functions ($F_{23} \sim F_{30}$). Similar to the comparative study on CEC2005 benchmark set, the comparative study on CEC2014

benchmark set is carried out for 10 and 30 dimensions and the results of the proposed HCLPSO algorithm are compared with the same PSO variants mentioned above.

A. Results for 10 dimensional problems

For 10 D problems, the performance of the proposed HCLPSO algorithm is compared with PSO, FIPS, UPSO, HPSO-TVAC, CLPSO and sHPSO algorithms, except OLPSO algorithm. The population size of 20, 30 repetitions and the number of function evaluations of 100,000 are used for all 10 dimensional problems. The comparison of the proposed HCLPSO and other PSO variants on 10 D problems are shown in Table 3-8. For the unimodal functions ($F_1 \sim F_3$), UPSO algorithm offers the best performance on function F_1 and CLPSO yields the best results on function F_2 and F_3 . The proposed HCLPSO algorithm performs second best on all three unimodal functions and outperforms PSO, FIPS, HPSO-TVAC and sHPSO algorithms.

The performance of HCLPSO improves on shifted and rotated multimodal functions ($F_4 \sim F_{16}$). The proposed HCLPSO algorithm outperforms all compared PSO algorithms on functions $F_5, F_{12}, F_{13}, F_{14}, F_{15}$ and F_{16} and performs the second best on functions F_6, F_8, F_9, F_{20} and F_{21} . FIPS performs the best on function F_6 and CLPSO achieves the best results on functions $F_7 \sim F_{11}$. On the partially separable hybrid functions ($F_{17} \sim F_{22}$), HCLPSO, PSO and CLPSO offers the best results on functions (F_{17}, F_{20}), F_{21} and (F_{18}, F_{19}, F_{22}) respectively. On the composition functions ($F_{23} \sim F_{30}$), sHPSO algorithm performs the best on function F_{23} , FIPS on functions (F_{24} and F_{25}), HCLPSO on functions (F_{26} and F_{27}) and HPSO-TVAC on functions (F_{28} and F_{29}) and CLPSO on function F_{30} .

Table 3-8 also ranks the performance of the algorithms in terms of mean error value. According to the ranking, the proposed HCLPSO algorithm ranks either first or second and outperforms compared PSO algorithms on most of the problems. It can be observed from the final ranking. The proposed HCLPSO ranks first and provides the best overall performance while the CLPSO is the second best, followed by FIPS, PSO, UPSO, HPSO-TVAC and sHPSO algorithms.

As mentioned in Section 3.4.2, the results are also statistically verified by non-parametric Wilcoxon signed rank test. The Wilcoxon signed rank test results are pre-

sented in Table 3-9 and the number of (+/0/-) is shown in the last row of the table. It can be clearly seen from the results in Table 3-9 that the proposed HCLPSO achieves superior performance over compared PSO algorithms.

Table 3-8 Comparison of HCLPSO vs other PSOs on CEC2014 10 dimensional problems

F	Criteria	HCLPSO	PSO	FIPS	UPSO	CLPSO	HPSO-TVAC	sHPSO
F_1	mean	1.12E+04	2.02E+04	2.01E+04	4.46E+03	5.12E+04	3.84E+04	3.26E+04
	std	1.15E+04	3.69E+04	1.67E+04	4.84E+03	8.73E+04	6.14E+04	1.09E+05
	rank	2	4	3	1	7	6	5
F_2	mean	8.29E+01	2.54E+03	5.72E+02	2.08E+03	6.04E+01	1.61E+03	1.83E+07
	std	1.61E+02	1.72E+03	5.83E+02	2.34E+03	1.18E+02	1.67E+03	1.31E+08
	rank	2	6	3	5	1	4	7
F_3	mean	8.92E+01	2.24E+02	3.32E+02	3.12E+02	6.54E+01	8.73E+02	2.30E+03
	std	1.07E+02	4.67E+02	2.61E+02	6.11E+02	7.66E+01	1.48E+03	3.06E+03
	rank	2	3	5	4	1	6	7
F_4	mean	4.61	19.53	18.14	12.77	0.13	1.80	18.57
	std	1.04E+01	1.89E+01	1.70E+01	1.59E+01	1.92E-01	1.79E+00	1.80E+01
	rank	3	7	5	4	1	2	6
F_5	mean	18.68	20.28	19.32	20.01	19.04	19.61	20.04
	std	5.08	0.08	3.99	0.01	3.88	2.80	0.11
	rank	1	7	3	5	2	4	6
F_6	mean	0.07	1.42	0.02	2.61	0.27	3.51	3.67
	std	0.14	1.11	0.03	1.21	0.21	1.20	1.79
	rank	2	4	1	5	3	6	7
F_7	mean	0.05	0.15	0.07	0.05	0.04	0.41	2.32
	std	0.03	0.07	0.05	0.02	0.02	0.20	5.95
	rank	3	5	4	2	1	6	7
F_8	mean	0.10	2.45	0.37	9.52	0.00	0.47	11.62
	std	0.30	1.65	0.48	4.55	0.00	0.70	7.74
	rank	2	5	3	6	1	4	7
F_9	mean	5.06	10.32	5.80	14.32	3.74	14.61	21.51
	std	1.73	4.29	3.26	5.19	1.08	5.54	10.70
	rank	2	4	3	5	1	6	7
F_{10}	mean	5.59	137.78	15.45	321.24	0.01	162.27	258.75
	std	21.56	98.67	23.66	158.21	0.03	85.20	186.29
	rank	2	4	3	7	1	5	6
F_{11}	mean	1.16E+02	2.99E+02	2.53E+02	5.94E+02	8.97E+01	3.61E+02	7.09E+02
	std	8.82E+01	2.11E+02	1.75E+02	2.16E+02	7.90E+01	2.07E+02	3.17E+02
	rank	2	4	3	6	1	5	7
F_{12}	mean	0.10	0.49	1.06	0.18	0.14	0.18	0.36
	std	0.04	0.33	0.18	0.11	0.03	0.13	0.24
	rank	1	6	7	4	2	3	5
F_{13}	mean	0.09	0.16	0.12	0.14	0.13	0.39	0.42
	std	0.04	0.33	0.18	0.11	0.03	0.13	0.24
	rank	1	5	2	4	3	6	7

F_{14}	mean	0.09	0.13	0.15	0.20	0.13	0.45	0.43
	std	0.02	0.04	0.05	0.07	0.03	0.22	0.31
	rank	1	3	4	5	2	7	6
F_{15}	mean	0.68	1.02	1.41	0.99	0.70	2.79	4.93
	std	0.21	0.44	0.33	0.35	0.12	1.49	8.14
	rank	1	4	5	3	2	6	7
F_{16}	mean	1.50	2.07	2.14	2.39	1.73	2.45	2.87
	std	0.34	0.63	0.39	0.48	0.33	0.42	0.43
	rank	1	3	4	5	2	6	7
F_{17}	mean	9.67E+02	3.99E+03	1.94E+03	2.20E+03	4.04E+03	1.78E+03	1.83E+04
	std	1.05E+03	3.75E+03	1.85E+03	2.02E+03	3.71E+03	2.28E+03	8.97E+04
	rank	1	5	3	4	6	2	7
F_{18}	mean	4.85E+02	7.92E+03	3.33E+03	2.23E+03	8.40E+01	7.58E+03	8.51E+03
	std	6.96E+02	1.04E+04	3.18E+03	3.62E+03	2.05E+02	6.99E+03	8.72E+03
	rank	2	6	4	3	1	5	7
F_{19}	mean	0.52	1.49	0.52	1.41	0.38	1.95	2.35
	std	0.43	0.88	0.36	0.33	0.32	0.80	1.44
	rank	3	5	2	4	1	6	7
F_{20}	mean	4.00E+01	1.68E+02	1.64E+03	4.62E+02	6.57E+01	1.62E+03	6.35E+03
	std	8.85E+01	5.80E+02	2.17E+03	6.28E+02	7.25E+01	1.77E+03	8.63E+03
	rank	1	3	6	4	2	5	7
F_{21}	mean	1.73E+02	8.90E+01	3.52E+02	2.66E+03	4.63E+02	1.64E+03	3.51E+03
	std	3.07E+02	8.61E+01	3.32E+02	3.27E+03	6.17E+02	2.16E+03	5.01E+03
	rank	2	1	3	6	4	5	7
F_{22}	mean	4.90	11.83	12.09	61.11	1.47	78.25	101.02
	std	21.76	10.05	26.81	58.78	5.09	61.39	77.16
	rank	2	3	4	5	1	6	7
F_{23}	mean	3.29E+02	3.29E+02	3.29E+02	3.29E+02	3.05E+02	3.29E+02	2.99E+02
	std	2.27E-13	2.60E-13	2.89E-13	2.89E-13	7.62E+01	6.85E-12	5.55E+01
	rank	5	5	3	3	2	7	1
F_{24}	mean	1.14E+02	1.20E+02	1.09E+02	1.28E+02	1.12E+02	1.34E+02	1.60E+02
	std	3.67E+00	4.89E+00	3.70E+00	1.97E+01	2.98E+00	2.38E+01	3.48E+01
	rank	3	4	1	5	2	6	7
F_{25}	mean	1.36E+02	1.86E+02	1.34E+02	1.79E+02	1.39E+02	1.90E+02	1.92E+02
	std	2.80E+01	3.11E+01	2.42E+01	2.05E+01	1.95E+01	1.92E+01	1.99E+01
	rank	2	5	1	4	3	6	7
F_{26}	mean	1.00E+02	1.00E+02	1.03E+02	1.00E+02	1.00E+02	1.02E+02	1.00E+02
	std	5.14E-02	2.36E-02	1.82E+01	3.42E-02	1.41E-01	1.40E+01	2.87E-02
	rank	3	2	7	4	5	6	1
F_{27}	mean	2.92E+02	2.90E+02	2.68E+02	1.35E+02	3.09E+02	2.97E+02	1.19E+02
	std	1.37E+02	1.19E+02	1.64E+02	1.62E+02	1.80E+02	1.69E+02	1.57E+02
	rank	5	4	3	2	7	6	1
F_{28}	mean	4.27E+02	3.78E+02	5.08E+02	3.71E+02	3.38E+02	4.68E+02	3.60E+02
	std	5.76E+01	3.41E+01	8.26E+01	2.38E+01	5.10E+01	2.04E+02	7.95E+01
	rank	5	4	7	3	1	6	2

F_{29}	mean	1.13E+06	4.72E+02	1.26E+05	2.88E+02	2.32E+02	1.97E+05	2.91E+02
	std	1.65E+06	7.11E+01	4.80E+05	9.12E+01	5.12E+01	6.07E+05	4.88E+01
	rank	7	4	5	2	1	6	3
F_{30}	mean	8.09E+02	9.42E+02	1.26E+03	5.87E+02	9.49E+02	1.16E+03	6.65E+02
	std	2.92E+02	1.36E+02	3.19E+02	7.60E+01	5.43E+02	9.43E+02	9.63E+01
	rank	3	4	7	1	5	6	2
Average rank		1.93	4.47	3.43	4.60	2.17	4.97	6.37
Final rank		1	4	3	5	2	6	7
Best/2nd Best/Worst		10/14/0	1/0/3	3/3/1	1/1/4	12/10/1	2/2/3	1/0/19
Algorithms		HCLPSO	PSO	FIPS	UPSO	CLPSO	HPSO-TVAC	sHPSO

Table 3-9 Wilcoxon Signed Rank Test Results of Single-Problem Analysis with a Significance Level of $\alpha = 0.05$ for CEC2014 10 dimensional problems

Functions	Pairwise Comparison HCLPSO versus					
	PSO	FIPS	UPSO	CLPSO	HPSO-TVAC	sHPSO
F_1	0	+	—	+	+	0
F_2	+	+	+	0	+	+
F_3	0	+	0	0	+	+
F_4	+	+	0	—	0	+
F_5	+	+	+	+	—	0
F_6	+	0	+	+	+	+
F_7	+	0	0	—	+	+
F_8	+	+	+	—	+	+
F_9	+	+	+	—	+	+
F_{10}	+	+	+	—	+	+
F_{11}	+	+	+	0	+	+
F_{12}	+	+	+	+	+	+
F_{13}	+	+	+	+	+	+
F_{14}	+	+	+	+	+	+
F_{15}	+	+	+	0	+	+
F_{16}	+	+	+	+	+	+
F_{17}	+	+	+	+	+	+
F_{18}	+	+	+	—	+	+
F_{19}	+	0	+	0	+	+
F_{20}	—	0	+	+	+	+
F_{21}	0	+	+	+	+	+
F_{22}	+	0	—	—	+	+
F_{23}	0	—	+	0	+	=
F_{24}	+	—	+	0	+	+
F_{25}	+	+	+	+	+	+
F_{26}	+	+	+	+	+	+
F_{27}	+	—	+	0	+	+
F_{28}	+	0	+	0	0	+

F_{29}	+	+	+	0	-	+
F_{30}	+	0	+	-	+	+
+ / 0 / -	25/4/1	20/7/3	25/3/2	12/10/8	26/2/2	27/2/1
	PSO	FIPS	UPSO	CLPSO	HPSO-TVAC	sHPSO

B. Results for 30 dimensional problems

In addition, the proposed HCLPSO algorithm is also compared with OLPSO algorithm on 30 dimensions. The 30 dimensional experimental results of OLPSO and CLPSO algorithms are obtained from [201]. In order to compare fairly, the 30D experiments for the proposed HCLPSO and other PSO variants are carried out using the same number of function evaluations of 300,000, the population size 40 and 30 repetitions as mentioned in [201]. The comparison between the results of the proposed HCLPSO algorithm and other PSO algorithms is presented in Table 3-10.

From the comparison in Table 3-10, it can be observed that the algorithms obtained similar ranking as in 10 dimensional problems. Compared to 10 D problems, the 30 D problems are more difficult to solve and the results are not as good as in 10 D problems. However, the proposed HCLPSO algorithm performs well and surpasses all compared PSO algorithms on functions $F_1, F_2, F_8, F_{11}, F_{12}, F_{13}, F_{14}, F_{15}, F_{16}, F_{18}, F_{19}, F_{21}$ and F_{25} . Especially, the HCLPSO significantly improves the results on functions F_7, F_8, F_{18}, F_{23} and finds the global optimum on F_8 .

The algorithms are ranked according to their mean solution error values and their average ranks are calculated and shown in Table 3-10. It can be observed in ranking that HCLPSO ranks first and its average rank is much different from the average rank of the second ranked OLPSO algorithm. The results in Table 3-10 are also verified by Wilcoxon signed rank test results as presented in Table 3-11. In Wilcoxon signed rank test table, the number of (+/0/-) also showed that the proposed HCLPSO algorithm performs significantly better than the compared PSO algorithms.

The convergence progress of mean solution values of 30 runs is also plotted and shown in Figures 3-8, 3-9, 3-10 and 3-11 for unimodal, multimodal, expanded and hybrid functions, respectively. It can be observed from the figures that HCLPSO shows more outstanding convergence performance than other PSOs, especially on multimodal problems.

Table 3-10 Comparison of HCLPSO vs other PSOs on CEC2014 30 dimensional problems

F	Criteria	HCLPSO	PSO	FIPS	UPSO	CLPSO	OLPSO	HPSO-TVAC	sHPSO
F_1	mean	2.12E+05	6.35E+06	4.64E+06	3.41E+05	9.41E+06	6.12E+06	3.18E+05	4.69E+05
	std	1.70E+05	6.83E+06	1.33E+06	2.73E+05	3.02E+06	3.58E+06	2.53E+05	4.45E+05
	rank	1	7	5	3	8	6	2	4
F_2	mean	1.63E+02	4.90E+01	3.24E+03	5.93E+03	2.76E+02	1.28E+03	3.78E+03	5.19E+03
	std	4.72E+02	1.19E+02	2.31E+03	4.47E+03	6.84E+02	1.48E+03	3.04E+03	7.76E+03
	rank	2	1	5	8	3	4	6	7
F_3	mean	9.94E+01	1.12E+02	1.37E+03	8.94E+02	3.24E+02	3.23E+02	4.63E+02	5.90E+03
	std	1.13E+02	1.24E+02	6.69E+02	1.16E+03	2.79E+02	5.69E+02	3.52E+02	6.30E+03
	rank	1	2	7	6	4	3	5	8
F_4	mean	3.98E+01	1.59E+02	7.70E+01	1.91E+01	8.07E+01	8.64E+01	5.89E+01	5.11E+01
	std	3.32E+01	3.90E+01	1.62E+01	3.03E+01	1.58E+01	2.22E+01	2.50E+01	4.79E+01
	rank	2	8	5	1	6	7	4	3
F_5	mean	2.02E+01	2.08E+01	2.09E+01	2.01E+01	2.05E+01	2.03E+01	2.00E+01	2.00E+01
	std	4.88E-02	1.07E-01	7.25E-02	7.15E-02	4.95E-02	1.28E-01	1.68E-04	3.30E-04
	rank	4	7	8	3	6	5	1	2
F_6	mean	4.71E+00	1.17E+01	3.79E+00	2.43E+01	1.43E+01	5.09E+00	2.04E+01	2.21E+01
	std	2.30E+00	2.85E+00	2.07E+00	2.61E+00	1.38E+00	1.48E+00	2.74E+00	2.42E+00
	rank	2	4	1	8	5	3	6	7
F_7	mean	5.77E-04	1.91E-02	6.60E-05	2.40E-03	6.68E-05	1.02E-13	9.57E-03	2.29E-01
	std	1.44E-03	1.73E-02	2.81E-04	5.25E-03	5.86E-05	3.47E-14	1.84E-02	3.38E-01
	rank	4	7	2	5	3	1	6	8
F_8	mean	0.00E+00	2.11E+01	5.29E+01	9.70E+01	3.95E-11	0.00E+00	8.42E+00	6.99E+01
	std	0.00E+00	7.17E+00	8.02E+00	1.74E+01	5.48E-11	0.00E+00	3.93E+00	2.02E+01
	rank	1	5	6	8	3	1	4	7
F_9	mean	4.30E+01	6.56E+01	1.42E+02	1.12E+02	6.11E+01	4.06E+01	1.05E+02	1.46E+02
	std	7.05E+00	1.46E+01	1.43E+01	2.22E+01	7.92E+00	7.02E+00	1.96E+01	3.76E+01
	rank	2	4	7	6	3	1	5	8
F_{10}	mean	8.14E+00	6.76E+02	1.40E+03	2.65E+03	3.13E+00	8.72E-02	4.94E+02	1.74E+03
	std	3.01E+01	2.61E+02	4.68E+02	5.50E+02	1.52E+00	2.04E-01	1.77E+02	6.02E+02
	rank	3	5	6	8	2	1	4	7
F_{11}	mean	1.82E+03	2.86E+03	5.86E+03	3.30E+03	2.87E+03	2.28E+03	2.79E+03	3.43E+03
	std	3.39E+02	4.71E+02	3.71E+02	4.59E+02	2.73E+02	4.66E+02	5.55E+02	7.21E+02
	rank	1	4	8	6	5	2	3	7
F_{12}	mean	1.85E-01	1.67E+00	2.36E+00	3.15E-01	5.38E-01	2.28E-01	2.93E-01	5.52E-01
	std	5.00E-02	5.27E-01	2.80E-01	1.03E-01	7.21E-02	6.38E-02	1.21E-01	2.50E-01
	rank	1	7	8	4	5	2	3	6
F_{13}	mean	2.18E-01	4.76E-01	3.06E-01	3.10E-01	3.32E-01	2.59E-01	3.92E-01	4.59E-01
	std	4.71E-02	1.09E-01	3.91E-02	5.03E-02	3.46E-02	3.20E-02	9.17E-02	1.26E-01
	rank	1	8	3	4	5	2	6	7
F_{14}	mean	2.20E-01	3.10E-01	2.57E-01	2.36E-01	2.78E-01	2.41E-01	2.68E-01	3.29E-01
	std	2.87E-02	1.32E-01	2.53E-02	4.53E-02	2.98E-02	2.66E-02	4.89E-02	1.50E-01
	rank	1	7	4	2	6	3	5	8
F_{15}	mean	4.50E+00	6.29E+00	1.49E+01	7.67E+00	8.62E+00	6.67E+00	4.56E+01	3.33E+01
	std	1.45E+00	2.33E+00	7.86E-01	2.10E+00	1.09E+00	1.62E+00	1.90E+01	2.44E+01
	rank	1	2	6	4	5	3	8	7

F_{16}	mean	9.28E+00	1.12E+01	1.18E+01	1.16E+01	1.06E+01	1.17E+01	1.11E+01	1.23E+01
	std	5.78E-01	5.99E-01	2.84E-01	5.38E-01	3.80E-01	5.48E-01	5.57E-01	4.48E-01
	rank	1	4	7	5	2	6	3	8
F_{17}	mean	1.08E+05	5.91E+05	3.57E+05	7.84E+04	8.59E+05	7.98E+05	6.49E+04	1.70E+05
	std	1.54E+05	3.45E+05	1.85E+05	5.17E+04	3.58E+05	4.13E+05	5.00E+04	1.54E+05
	rank	3	6	5	2	8	7	1	4
F_{18}	mean	1.32E+02	6.42E+04	2.37E+03	7.39E+02	1.69E+02	3.58E+02	1.74E+03	5.15E+03
	std	1.15E+02	3.21E+05	2.22E+03	1.34E+03	5.85E+01	5.12E+02	1.80E+03	6.73E+03
	rank	1	8	6	4	2	3	5	7
F_{19}	mean	5.65E+00	8.13E+00	6.09E+00	9.80E+00	8.35E+00	6.13E+00	1.48E+01	1.00E+01
	std	9.72E-01	2.36E+00	6.36E-01	1.13E+01	8.04E-01	8.20E-01	2.96E+00	1.97E+00
	rank	1	4	2	6	5	3	8	7
F_{20}	mean	9.14E+02	3.48E+02	1.10E+03	3.80E+03	3.26E+03	5.58E+03	2.30E+03	7.74E+03
	std	9.86E+02	2.37E+02	4.66E+02	2.50E+03	1.70E+03	4.01E+03	1.77E+03	6.18E+03
	rank	2	1	3	6	5	7	4	8
F_{21}	mean	3.03E+04	9.66E+04	9.70E+04	4.20E+04	8.08E+04	1.07E+05	4.69E+04	1.10E+05
	std	3.24E+04	8.56E+04	5.38E+04	3.47E+04	3.99E+04	8.33E+04	3.56E+04	1.00E+05
	rank	1	5	6	2	4	7	3	8
F_{22}	mean	2.07E+02	2.68E+02	1.78E+02	3.73E+02	1.75E+02	2.20E+02	5.61E+02	5.43E+02
	std	8.58E+01	1.53E+02	4.95E+01	1.28E+02	7.77E+01	1.07E+02	2.07E+02	2.31E+02
	rank	3	5	2	6	1	4	8	7
F_{23}	mean	3.15E+02	3.16E+02	3.15E+02	3.15E+02	3.15E+02	3.15E+02	3.14E+02	3.00E+02
	std	2.80E-12	2.00E-01	2.63E-13	1.45E-13	4.86E-05	1.23E-10	1.87E-11	3.98E+01
	rank	3	8	6	6	3	3	2	1
F_{24}	mean	2.24E+02	2.30E+02	2.23E+02	2.28E+02	2.25E+02	2.24E+02	2.30E+02	2.31E+02
	std	1.47E+00	6.70E+00	6.21E-01	3.62E+00	1.29E+00	5.47E-01	1.93E+00	2.20E+01
	rank	3	7	1	5	4	2	6	8
F_{25}	mean	2.05E+02	2.09E+02	2.05E+02	2.14E+02	2.08E+02	2.09E+02	2.17E+02	2.10E+02
	std	1.38E+00	1.97E+00	5.33E-01	5.86E+00	1.19E+00	1.75E+00	4.15E+00	9.20E+00
	rank	1	4	2	7	3	5	8	6
F_{26}	mean	1.00E+02	1.23E+02	1.30E+02	1.14E+02	1.00E+02	1.00E+02	1.17E+02	1.37E+02
	std	6.84E-02	6.32E+01	4.66E+01	3.45E+01	7.35E-02	4.44E-02	3.77E+01	4.88E+01
	rank	3	6	7	4	1	1	5	8
F_{27}	mean	4.02E+02	5.82E+02	3.62E+02	4.61E+02	4.17E+02	3.26E+02	7.17E+02	7.27E+02
	std	1.33E+00	9.83E+01	4.40E+01	1.68E+02	5.24E+00	3.80E+01	2.70E+02	3.47E+02
	rank	3	6	2	5	4	1	7	8
F_{28}	mean	8.64E+02	1.04E+03	8.60E+02	2.61E+03	8.98E+02	8.73E+02	6.56E+02	1.93E+03
	std	8.22E+01	2.33E+02	4.43E+01	6.85E+02	5.32E+01	2.97E+01	3.18E+02	1.32E+03
	rank	3	6	2	8	5	4	1	7
F_{29}	mean	9.33E+02	1.60E+06	4.34E+03	1.17E+03	1.29E+03	1.36E+03	2.71E+02	1.03E+06
	std	7.83E+01	6.11E+06	2.47E+03	2.59E+02	1.69E+02	2.82E+02	2.89E+02	2.90E+06
	rank	2	8	6	3	4	5	1	7
F_{30}	mean	1.89E+03	3.33E+03	1.85E+03	2.51E+03	3.63E+03	2.39E+03	7.76E+02	2.29E+03
	std	4.67E+02	1.16E+03	3.59E+02	7.13E+02	1.00E+03	5.99E+02	2.23E+02	1.48E+03
	rank	3	7	2	6	8	5	1	4
Average rank		2.00	5.43	4.67	5.03	4.27	3.57	4.37	6.47

Final rank	1	7	5	6	3	2	4	8
Best/2 nd	13/6/0	2/2/5	2/6/3	1/3/5	2/3/3	6/4/0	5/2/4	1/1/11
Best/Worst								
Algorithms	HCLPSO	PSO	FIPS	UPSO	CLPSO	OLPSO	HPSO-TVAC	sHPSO

Table 3-11 Wilcoxon Signed Rank Test Results of Single-Problem Analysis with a Significance Level of $\alpha = 0.05$ for CEC2014 30 dimensional problems

Functions	Pairwise Comparison HCLPSO versus					
	PSO	FIPS	UPSO	CLPSO	HPSO-TVAC	sHPSO
F_1	+	+	+	+	+	+
F_2	0	+	+	0	+	+
F_3	0	+	+	+	+	+
F_4	+	+	—	+	+	+
F_5	+	+	—	+	—	—
F_6	+	0	+	+	+	+
F_7	+	0	0	0	+	+
F_8	+	+	+	—	+	+
F_9	+	+	+	+	+	+
F_{10}	+	+	+	0	+	+
F_{11}	+	+	+	0	+	+
F_{12}	+	+	+	+	+	+
F_{13}	+	+	+	+	+	+
F_{14}	+	+	0	+	+	+
F_{15}	+	+	+	+	+	+
F_{16}	+	+	+	+	+	+
F_{17}	+	+	0	+	—	+
F_{18}	+	+	+	0	+	+
F_{19}	+	0	+	+	+	+
F_{20}	—	0	+	+	+	+
F_{21}	+	+	+	+	0	+
F_{22}	0	0	—	0	+	+
F_{23}	+	—	0	+	+	+
F_{24}	+	—	+	0	+	0
F_{25}	+	+	+	+	+	+
F_{26}	+	+	+	+	+	+
F_{27}	+	—	+	+	+	+
F_{28}	+	0	+	+	—	+
F_{29}	+	+	+	0	—	+
F_{30}	+	0	+	+	—	0
+/0/—	26/3/1	20/7/3	23/4/3	21/8/1	24/1/5	27/2/1
	PSO	FIPS	UPSO	CLPSO	HPSO-TVAC	sHPSO

Figure 3-8 Convergence characteristic graphs of CEC2014 30 dimensional unimodal functions: (a) F_1 : Rotated High Conditional Elliptic Function and (b) F_3 : Rotated Discus Function

Figure 3-9 Convergence characteristic graphs of CEC2014 30 dimensional multimodal functions: (a) F_9 : Shifted and Rotated High Rastrigin's Function and (b) F_{10} : Shifted Schwefel's Function

(a) F_{15}

(b) F_{16}

Figure 3-10 Convergence characteristic graphs of CEC2014 30 dimensional expended functions: (a) F_{15} : Shifted and Rotated Expanded Griewank's plus Rosenbrock's Function and (b) F_{16} : Shifted and Rotated Expanded Scaffer's F_6 Function

(a) F_{18}

(b) F_{21}

Figure 3-11 Convergence characteristic graphs of CEC2014 30 dimensional hybrid functions

3.4.3 Discussions

The performance of the proposed HCLPSO algorithm is evaluated on CEC2005 and recent CEC2014 benchmark problems. The benchmark problems include a wide variety of different types of unimodal, multimodal, expanded, partially separable hybrid and composition problems. The whole problem set consists of 55 problems in total. The experiments have been conducted on 10 and 30 dimensional problems. The

proposed HCLPSO algorithm is compared with the standard PSO as well as the other PSO variants. Among 55 test problems, HCLPSO performs best on 25 problems and second best on 23 problems in 10 dimensions while it obtains the best results on 28 problems and second best on 11 problems in 30 dimensions. Once again, the outperformance of the HCLPSO over the other PSO algorithms is validated using non-parametric Wilcoxon signed rank test. In addition, the results are also reaffirmed using the convergence test. The convergence progress plots showed that the performance of HCLPSO is superior over compared PSO algorithms.

Therefore, the experimental results and comparisons verify that HCLPSO overall performs better than original PSO and improved PSO variants on CEC2005 as well as recent CEC2014 benchmark problems. HCLPSO performs consistently well and achieves high performance on all shifted rotated unimodal and multimodal problems. The algorithm also offers best performance on expanded and shifted rotated expanded problems as well as on hybrid functions and rotated hybrid composition problems. However, the algorithm does not perform well on two rotated hybrid composition functions on CEC2005 benchmark set. This can be explained by No Free Lunch Theorem (NFLT) [202]. According to no free lunch theorem, a general-purpose universal optimization algorithm is theoretically impossible. Therefore, no strategy can be expected to outperform another on all types of optimization problems.

When comparing with CLPSO, the results show that HCLPSO can find more accurate solutions with faster convergence speed in most unimodal, multimodal, expanded and hybrid composite functions. Thus, HCLPSO offers higher solution accuracy and faster convergence speed than CLPSO. Compared to other PSO algorithms, HCLPSO either performs the best or achieves the comparable performance.

This implies that HCLPSO has gained benefits from the methods described in Section 3.1. By using heterogeneous subpopulations that are assigned for exploration and exploitation processes separately, it enhances the exploration and exploitation processes without crippling one another. Using CL learning strategy and its velocity update equation [78], the particles can learn from different dimensions of different particles and avoid two problems, "oscillation phenomenon" [193] and "two steps forward, one step back phenomenon" [38]. With generating the exemplar for the exploration sub-

population by using its own members' best experiences, information sharing among the particles is controlled and able to prevent premature convergence. Lastly, the respective exploration/exploitation ability for each subpopulation group is enhanced by using time varying acceleration coefficients [57]. Therefore, the trade-off between exploration and exploitation is achieved and the HCLPSO performs better or comparable to other PSO variants on shifted rotated unimodal, multimodal, expanded and hybrid composition functions.

One of the limitations on HCLPSO algorithm is using CL strategy for solving unimodal problems. Although learning of different dimensions from different exemplars is efficient for searching the global optimum of multimodal problems, tuning the algorithm to find the optimal solution on unimodal problems is ineffective and results in low convergence speed. In accordance with no free lunch theorem, there is no general-purpose optimization algorithm that can solve all classes of problems. All the algorithms have pros and cons on solving optimization problems. Secondly, compared to other PSO algorithms, the HCLPSO algorithm is more complex and requires additional computations. However, the computational resources are mostly spent on the evaluating of objective function in EA algorithms. Hence, the additional computational requirement of the optimization strategy for HCLPSO is not significant. In addition, the HCLPSO offers faster convergence rate and this can be translated into additional saving on computation time.

Though HCLPSO is a promising optimization algorithm, as mentioned above comprehensive learning strategy has to be tuned to get better performance. To address this issue, adaptive or self-learning strategy will be investigated in the future.

3.5 Conclusion

In this Chapter, CLPSO with enhanced exploration and exploitation is presented in which the entire population is divided into two heterogeneous subpopulation groups. Each subpopulation group is especially designed to enhance exploration and exploitation performances respectively. To prevent loss of diversity, the explorative particles do not interact with the exploitative particles. Furthermore, linearly decreasing inertia weight and time varying acceleration coefficients are used in both groups to bias

the exploration and exploitation balance from exploration in early stages to exploitation in the later stages. Intensive performance evaluation of the proposed HCLPSO algorithm is conducted using 55 benchmark problems. The experiment results showed that the proposed algorithm outperforms the current state-of-the-art PSO variants.

4 ENSEMBLE PARTICLE SWARM OPTIMIZER

4.1 Introduction

According to no free lunch (NFL) theorem, no strategy can be expected to perform better than any other on all classes of problems [202]. Hence, there is no single optimization algorithm for solving all types of optimization problems. An algorithm may outperform another on some problems, but its performance may be worse off on other problems. Thus, researchers have focused on enhancing the universality and robustness performance of optimization algorithms in recent years. In order to find the best algorithm for a particular problem, it requires numerous trial-and-error runs to be conducted. Hence, it requires unrealistic computational time. Instead of relying on a single optimization method with numerous trial-and-error-based experiments, ensemble of different optimization algorithms could be a potential solution to solve a variety of complex optimization problems. Therefore, the idea of ensemble method has been gained much attention among the researchers to improve the universality and robustness performance of optimization algorithms.

Olorunda and Engelbrecht investigated the possibility of combining different evolutionary algorithms with the concept of cooperation. The authors proposed heterogeneous cooperative algorithm in which DE, GA and PSO algorithms are combined to exploit useful search characteristics of each algorithm [203]. The heterogeneous model

showed consistent and competitive performance compared to non-heterogeneous models. In [35], Engelbrecht proposed heterogeneous PSO (HPSO) where a pool of different PSO search behaviours was constructed. The particles were allocated by randomly selecting position and velocity update rules from the behaviour pool. The empirical results proved that heterogeneous PSO with ensemble of different search behaviours outperformed homogeneous PSO algorithms in terms of solution quality and scalability.

Multi-strategy ensemble ABC (MEABC) algorithm was recently proposed in [204]. In MEABC, throughout the search process, the strategy pool was constructed by employing three different ABC solution search strategies. Initially, the strategies were randomly assigned to each food source. During the search process, the strategy for each food source is dynamically changed according to the quality of new candidate solutions. If the quality of a new candidate solution was better than the current candidate solution, the current strategy was selected for follow-up search process. If it failed to keep improving the quality of the solutions, new strategy was randomly selected from the strategy pool. In this way, bees use different strategies to find new solutions.

Recently, researchers have used self-adaptive mechanism to enhance the performance of optimization algorithms. In order to avoid time-consuming trial-and-error process, Qin and Suganthan proposed self-adaptive differential DE (SaDE) algorithm in which different trial vector generation strategies and their associated parameter settings were gradually self-adapted by learning from their previous experiences in generating promising solutions [205]. As a result, the best-matched trial vector generation strategy as well as its associated parameters can be determined for different stages of the search process. Ensemble strategies with adaptive EP was introduced in [206] where this ensemble approach also got benefits from different mutation operators with different parameter values whenever they were effective during the different stages of the search process.

Inspired by those ensemble methods mentioned above, ensemble of PSO algorithms is proposed in this Chapter. Among the population-based optimization algorithms, PSO has attracted considerable attention due to its simplicity and fast convergence speed [207] and many PSO variants have been successfully applied in real world applications [208]. However, it is extremely difficult to find a generalized PSO algo-

rithm that outperforms on various types of optimization problems. For instance, the *inertia weight* PSO performs well on unimodal problems, whereas it does not perform well on complex multimodal problems due to premature convergence to a local optimum [13]. As an alternative, popular comprehensive learning particle swarm optimizer (CLPSO) offers best performance on multimodal problems though its convergence rate on unimodal problems is unsatisfactory [78]. Recent global orthogonal learning PSO (OLPSO-G) algorithm outperforms local orthogonal learning PSO (OLPSO-L) on unimodal functions. OLPSO-L provides better performance on multimodal problems [79].

Therefore, it can be observed that different PSO variants have their own pros and cons. and each algorithm is more suitable for specific nature of optimization problems. Hybridizing these algorithms together, they can benefit from each other and enable us to solve various kinds of optimization problems. Inspired by this idea, ensemble of PSOs with strategy adaptation called EPSO is proposed in this Chapter. PSO variants of basic PSO with inertia weight (PSO) [3], comprehensive learning PSO (CLPSO) [78], fitness-distance-ratio based PSO (FDR-PSO) [31], self-organizing hierarchical PSO with time-varying acceleration coefficients (HPSO-TVAC) [57] and distance-based locally informed PSO (LIPS) [209] are hybridized together in the proposed EPSO algorithm.

As mentioned before, a number of PSO variants are hybridized to complement each other in the proposed EPSO. Basic PSO algorithm with inertia weight is used to ensure fast convergence speed. HPSO-TVAC is used to prevent the premature convergence. The CLPSO is used to maintain the population diversity throughout the optimization process. Furthermore, FDR-PSO is employed to enhance local search capability. Finally, LIPS is deployed to search multiple local optima in multimodal optimization. The details of these algorithms are discussed in Section 4.2. In terms of the problem type, *inertia weight* PSO and HPSO-TVAC are effective for solving unimodal problems while CLPSO, LIPS, and FDR-PSO are efficient in solving multimodal problems. Combining various PSO variants will complement each other and enhance robustness and generalization capabilities.

In order to select the most suitable PSO variant for a particle to update their direction in current phase of the search process, self-adaptive mechanism from [205] is introduced in the proposed ensemble PSO. In EPSO, a pool of PSO strategies is con-

structured and a probabilistic selection is performed based on the success rate of each PSO strategy in generating improved solutions. Initially, each PSO strategy is set with equal probability and stochastic universal sampling selection [210] is used to select a PSO strategy from the pool. As in [205], a fix number of generations called learning period (LP) is defined for PSO strategies in EPSO. Within LP period, if a PSO strategy can generate improved solutions, it is recorded in its success memory. If it fails to generate improved solutions, it is recorded in its failure memory. After LP generations, based on those success and failure memories, the success rate of each PSO strategy is updated at each subsequent generation, followed by updating the selection probability. The PSO strategy with higher success rate will have larger probability to be chosen. In this way, a suitable PSO strategy can be determined adaptively to match the stage of the search process and assigned to a particle in the swarm to update its search direction.

4.2 Ensemble of Particle Swarm Optimization algorithms

Depending on the characteristics of the algorithm as well as of the problem type, the performance of an optimization algorithm varies and it is very difficult to find a suitable optimization algorithm for a given problem. Hence, a high-level heuristic optimization technique, that aims to solve different classes of problems, is proposed in this Chapter by forming an ensemble of PSO algorithms namely, *inertia weight* PSO, HPSO-TVAC, FDR-PSO, LIPS and CLPSO algorithms.

4.2.1 PSO variants in the ensemble approach

4.2.1.1 *Inertia weight* PSO (PSO)

In light of premature convergence of the original PSO, Shi and Eberhart incorporated a new parameter called inertia weight ‘ w ’ into the PSO algorithm [13]. Inertia weight w is defined as a linearly decreasing function of time to adjust the balance between the global search and local search. In *inertia weight* PSO, the position and velocity of the particles are updated according to the equation (2.2) and (2.3) [3] presented in Chapter 2.

A large value of inertia weight w encourages the global search while a small value of w enhances the local search during the search process. By decreasing inertia weight w linearly from a relatively large value to a small value through the optimization process, *inertia weight* PSO has more global search ability at the early stages of the optimization process and more local search ability at the latter stages. *Inertia weight* PSO can converge quickly to the global optima and performs significantly better than the original PSO [3]. With the decreasing inertia weight “ w ” value, the *inertia weight* PSO algorithm lacks the global search ability at the latter stages of the search even when the global search ability is required to jump out of the local optimum.

4.2.1.2 Comprehensive learning particle swarm optimizer (CLPSO)

In PSO, the trajectory towards the global optimum is adjusted by the particles' own $pbest_i^d$ and $gbest^d$. Though $gbest$ is the whole population's best experience, it may be an inferior local optimum for a multimodal problem and far from the global optimum. To address this issue, a comprehensive learning particle swarm optimizer (CLPSO) was proposed in [78]. In CLPSO, all particles' best experiences are used to guide the search of a particle. A particle learns from different particles' $pbests$ for different dimensions. The velocity of the i^{th} particle is updated according to the equation (3.1) [78]. The details of CLPSO algorithm was presented in Chapter 3.2. Beside CLPSO velocity update rule, CLPSO velocity update rule with $gbest$ (equation (3.3)) presented in the HCLPSO algorithm is also used in the proposed EPSO algorithm.

By combining the CL strategy with the whole swarm's best experience, the new $gbest$ -guided CLPSO velocity update rule will encourage the particles to converge to the global optimum while maintaining the population diversity.

4.2.1.3 Fitness-distance-ratio based particle swarm optimization (FDR-PSO)

FDR-PSO was proposed to tackle the problem of premature convergence observed in PSO [31]. In comparison with PSO, FDR-PSO added a social learning component that learns from the neighbouring particle's experience ($nbest$). The neighbouring particles are selected with two criteria: (1) they must be near to the particle being updated and (2) they must have better fitness than the particle being updated. Whether a

neighbouring particle satisfies these criteria or not is decided by the ratio of fitness distance to one-dimensional distance called Fitness-Distance-Ratio. For maximization problem, the particle that can maximize Fitness-Distance-Ratio is selected as the neighbouring particle. Let the neighbouring particle $nbest$ be the particle j that maximizes the following Fitness-Distance-Ratio:

$$\frac{Fitness(P_j) - Fitness(X_j)}{|P_j^d - X_j^d|} \quad (4.1)$$

where, $|\dots|$ is the absolute value. For a minimization problem, the particle that can minimize Fitness-Distance-Ratio is selected as the neighbour particle and $(Cost(P_j) - Cost(X_j))$ is instead used in the numerator of the above expression in which $Cost(\)$ is the objective function of a given maximization problem [31]. After selecting the neighbouring particle $nbest$, the velocity component in the d^{th} dimension of the i^{th} particle is updated according to the following equation:

$$V_i^d = w * V_i^d + c_1 * rand1_i^d * (pbest_i^d - X_i^d) + c_2 * rand2_i^d * (gbest^d - X_i^d) + c_3 * (nbest_j^d - X_i^d) \quad (4.2)$$

where, $c_1 = 1$, $c_2 = 1$ and $c_3 = 2$. $pbest$ is the cognitive component, the particle's own experience and $gbest$ is the social component, the best experience found so far by the whole population, $nbest$ is the experience of neighbouring particle which maximize/minimize Fitness-Distance-Ratio. FDR-PSO was tested on six benchmark functions and the experimental results showed that FDR-PSO performed significantly better than PSO, being capable of avoiding premature convergence and less likely to be stuck in a local optimum [31].

4.2.1.4 Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients (HPSO-TVAC)

In order to efficiently control the local search and convergence to the global optimum solution, Ratnaweera et al. [57] introduced time-varying acceleration coefficients (TVAC). The authors also introduced self-organizing hierarchical particle swarm optimizer (HPSO) together with TVAC, in which only cognitive and social parts are

considered to estimate the new velocity of each particle. The particles are reinitialized whenever they are stagnated in the search space. Authors claims that in the absence of the previous velocity term, the particles rapidly move toward a local optimum and stagnate due to lack of the momentum [57]. Therefore, HPSO is introduced to provide the required momentum for the particles to find the global optimum solution, in the absence of the previous velocity term in equation (2.1). In HPSO-TVAC algorithm, the velocity of a particle is updated according to the following equation:

$$V_i^d = c_1 * rand1_i^d * (pbest_i^d - X_i^d) + c_2 * rand2_i^d * (gbest^d - X_i^d) \quad (4.3)$$

where, the previous velocity term from the equation (2.1) is kept at zero and the modulus of velocity vector of a particle is reinitialized with a random velocity, i.e. $V_i^d = rand * v$ (*reinitialization velocity*) should it stagnate ($V_i^d = 0$) in the search space [57]. *rand* values are uniform distribution random numbers in the range of [0,1].

Therefore, in HPSO, a series of particle swarm optimizers are generated inside the main particle swarm optimizer according to the behaviour of the particles in the search space. The reinitialization velocity is set to be proportional to the maximum allowable velocity V_{max} in HPSO algorithm. In [57], two unimodal and three multimodal functions were used to evaluate the performance of the HPSO-TVAC algorithm. Though HPSO-TVAC performs much better than PSO, the performance of the algorithm is found to be poor in solving multimodal problems.

4.2.1.5 Distance-based locally informed PSO (LIPS)

Instead of the swarm's best experience (*gbest*), LIPS uses local bests, the best experiences of the neighbouring particles to guide the particles for finding the optimum over the search space [209]. The distances between the particle and other particles are measured by Euclidean distance. Then, the nearest neighbouring particles are selected to guide the particle. Using the local information from its nearest neighbours, the particle's velocity is adjusted using the formula as follows:

$$V_i^d = \chi * (V_i^d + \varphi(P_i^d - X_i^d)) \quad (4.4)$$

where,

$$P_i = \frac{\sum_{j=1}^{nsize} \varphi_j * nbest_j}{\varphi} \quad (4.5)$$

$$\varphi_j \sim U \left(0, \frac{4.1}{nsize} \right) \quad (4.6)$$

$$\varphi = \sum_{j=1}^{nsize} \varphi_j \quad (4.7)$$

where, $nbest_j$ is the j^{th} nearest neighbourhood to the i^{th} particle's $pbest$. $nsize$ is the neighbourhood size and it is dynamically increased from 2 to 5. χ is defined as constriction coefficient and $\chi = 0.7298$. ' φ_j ' is random positive number drawn from a uniform distribution in the range of $(0, 4.1/nsize)$. φ is defined as acceleration weight and summation of φ_j . The position is updated using the same equation as in equation (2.2). Euclidean distance based neighbourhood selection is $(O(N^2))$, where N is the number of particles in the current population. With the Euclidean distance based neighbourhood selection, LIPS uses the information of the neighbours from the same region and this improves local search ability of the algorithm, especially at the latter stages of the search. This ability enables the particles to converge quickly to global optimum with high accuracy. Compared to other PSO variants, LIPS performs well on multimodal optimization [209].

4.2.2 Ensemble Particle Swarm Optimizer (EPSO)

In the proposed ensemble of PSO algorithms, the population is divided into two subpopulation groups, small and large subpopulations, to enhance the diversity throughout the optimization process. The CLPSO algorithm is used in a small subpopulation to maintain the diversity whereas *inertia weight* PSO, HPSO-TVAC, FDR-PSO, LIPS and *gbest*-guided CLPSO algorithms are employed in large subpopulation through the ensemble approach. This structure will enable the proposed algorithm to maintain the diversity throughout the optimization process while providing fast convergence rate. A good candidate PSO strategy in large subpopulation is selected according to the probability learnt from the success rate of the previous experiences and the selected PSO strategy is used to update the particle's flying direction. Different PSO strategies in the large subpopulation are selected for each generation in the form of self-adaptive manner [205] as follows:

- Step 1. Let p_k be the probability of each PSO strategy and initialize each p_k as $1/K$ s that they have equal probability to be chosen, where, $k = 1, 2, \dots, K$ and K = total number of PSO strategies from the pool.
- Step 2. Stochastic universal selection method is used to select the candidate PSO strategy for the particles.
- Step 3. For generation G , if the selected PSO strategy can generate improved solutions, the selected PSO strategy is promoted by recording in success memories ($n_{sk,G}$). Otherwise, the PSO strategy is demoted by recording as failure memories ($n_{fk,G}$).
- Step 4. The success and failure memories are updated for a fixed number of generations called the learning period (LP). If the memories overflow after LP generations, the earliest record stored in the memories will be removed for the numbers obtained in the current generation to be stored in the memory.
- Step 5. After every LP generation, the probability of selecting PSO strategies from the pool will be updated as follows:
- Step 6.

$$p_{k,G} = \frac{S_{k,G}}{\sum_{k=1}^K S_{k,G}} \quad (4.8)$$

where,

$$S_{k,G} = \frac{\sum_{g=G-LP}^{G-1} n_{sk,g}}{\sum_{g=G-LP}^{G-1} n_{sk,g} + \sum_{g=G-LP}^{G-1} n_{fk,g}} + \varepsilon \quad (4.9)$$

where, $k = 1, 2, \dots, K$, $G > LP$ and $S_{k,G}$ represents the success rate of the solution generated by the k^{th} PSO strategy. $\varepsilon = 0.01$ is used to avoid the possible null success rate.

Figure 4-1 Convergence characteristic graph of EPSO and its respective self-adaptive PSO strategies for Unimodal Shifted Schwefel's Problem 1.2 function

Figure 4-2 Convergence characteristic graph of EPSO and its respective self-adaptive PSO strategies for Multimodal Shifted Rotated Griewank's Function

The PSO strategy with the larger success rate will have the larger probability to be selected to guide the particle at the current generation. In other words, the suitable PSO strategy algorithm for a given problem is automatically chosen by adaptations. To study the changes of PSO strategies, the proposed ensemble PSO algorithm (EPSO) with strategy adaptation is experimented with unimodal and multimodal problems from CEC2005 benchmark functions [195]. The convergence characteristic graphs and their respective strategy adaptation for unimodal and multimodal problems are presented in Figure 4-1 and 4-2.

As shown in the figures, the suitable PSO strategy is adapted in the different phases of the search process while CLPSO algorithm maintains the diversity. In addi-

tion, the experimental results show that the proposed EPSO can jump out of local optima and further search for the global optimum. The comparative study of the proposed EPSO and its individual PSO variants, as well as recent state-of-the-art PSO algorithms, is conducted for further performance investigation. The performance investigation results are discussed in the next section.

4.3 Experimental Set Up and Results

4.3.1 Experimental Set Up

The performance of the proposed EPSO algorithm is evaluated using the shifted and rotated CEC2005 [195] and CEC2013 [211] benchmark problems. The recent CEC2013 benchmark set is developed based on the CEC2005 problem set. A major difference from CEC2005 benchmark set is that the properties of sub-functions are merged better and the regions around the global/local optima is kept continuous in the recent CEC2013 composition functions. In addition, the new unimodal and multimodal problems are introduced in the recent CEC2013 test suite. Therefore, in order to assess the performance of EPSO on various problems, the algorithm is tested on both CEC2005 and recent CEC2013 benchmark problem sets. The test problems can be divided into four groups: unimodal, multimodal, expanded and hybrid composite functions. A total of 53 bench problems are used in this experiment. The search range, initialization range, bias values, and fixed accuracy values of CEC2005 and CEC2013 benchmark functions are described in Appendix 1. The performance of the proposed EPSO algorithm is compared with its individual PSO variants and the two other competitive PSO algorithms:

- *Inertia weight* PSO (PSO) [3]
- Comprehensive Learning PSO (CLPSO) [78]
- Self-organizing hierarchical PSO with time varying acceleration coefficients (HPSO-TVAC) [57]
- Fitness-Distance-Ratio based PSO (FDR-PSO) [31]

- Distance-based locally informed PSO (LIPS) [209]
- Orthogonal Learning PSO (OLPSO) [79]
- Static Heterogeneous Swarm Optimization (sHPSO) [35]
- Self-adaptive Heterogeneous PSO (f_k -PSO) [37]

The first five PSO algorithms are individual PSO variants in the proposed Ensemble PSO algorithm. The another one is recent start-of-art OLPSO algorithm. In OLPSO, orthogonal learning strategy is used to generate the exemplar from the particle's previous best experience and the swarm's best experience [79]. The next one is static heterogeneous PSO (sHPSO) which characteristic is similar to the proposed Ensemble PSO (EPSO) algorithm. Similar to the proposed EPSO algorithm, sHPSO constructed a pool of different PSO behaviours. However, unlike EPSO, the behaviours are selected randomly from the pool and they are kept unchanged throughout the search process. The last algorithm is the self-adaptive heterogeneous PSO called frequency-based HPSO (f_k -PSO) algorithm. Similar to EPSO algorithm, the f_k -PSO algorithm allows the particles to use different search behaviours and the behaviours are selected based on their success and failure performance. Unlike the EPSO algorithm, TVAC-PSO, TVIW-PSO, sPSO, cPSO, modified barebones PSO and the quantum PSO are used as the search behaviours in the f_k -PSO algorithm [37]. The major difference between EPSO and f_k -PSO is that EPSO has a small subpopulation group to maintain the population diversity even if the PSO search behaviours in the large subpopulation group fail to improve their performance for a number of iterations.

The experiments are conducted on 10 D and 30 D from CEC2005 as well as recent CEC2013 benchmark problems. Each algorithm is executed on each function for the number of trials as specified in two benchmark sets (30 trials for CEC2005 benchmark set and 51 trials for CEC2013 benchmark set). The number of function evaluations is set at (10000*D) for each trial of each function. For the CEC2005 benchmark problems, EPSO algorithm is compared with its individual variants as well as recent OLPSO algorithm. The 30 dimensional results of OLPSO algorithm were provided from the authors of OLPSO. For the CEC2013 test suite, besides its individual PSO variants, EPSO is also compared with the recent f_k -PSO algorithm. The 10 and 30 di-

mensional results of f_k -PSO algorithm are acquired from the self-adaptive heterogeneous PSO paper [37]. Except OLPSO and f_k -PSO algorithms, the parameter settings of all other algorithms are listed in Table 4-1. The same parameter setting is used for the individual PSO variants in the EPSO algorithm.

Table 4-1 Parameter Settings

Algorithm	Inertia weight w	Constriction Coefficients χ	Acceleration Coefficients c_1, c_2, c	Neighbourhood size	Reference
<i>Inertia weight</i> PSO	0.9 ~ 0.2	-	$c_1 = 2, c_2 = 2$	-	[13]
LIPS	-	0.729	$c = 2$	3	[209]
FDR-PSO	0.9 ~ 0.2	0.729	$c_1 = 1, c_2 = 1, c_3 = 2$	-	[31]
CLPSO	0.9 ~ 0.2	-	$c = 3 \sim 1.5$	-	[78]
HPSO-TVAC	-	-	$c_1 = 2.5 \sim 0.5$ $c_2 = 0.5 \sim 2.5$	-	[57]
sHPSO	0.72	-	$c_1 = 2.5 \sim 0.5$ $c_2 = 0.5 \sim 2.5$	-	[35]
CLPSO with <i>gbest</i>	0.9 ~ 0.2	-	$c_1 = 2.5 \sim 0.5$ $c_2 = 0.5 \sim 2.5$	-	-

In order to assess the performance of EPSO algorithm, the statistical analysis is carried out using the non-parametric Wilcoxon signed-rank test [195]. The evaluation of Wilcoxon signed rank test is conducted between the results of the proposed EPSO algorithm and the results of the other PSO variants with the significance level of 5%. The symbol (+) represents that EPSO performs significantly better than the compared algorithm, the symbol (0) represents that there is no significant difference between EPSO and the compared algorithm and the symbol (-) represents that the compared algorithm performs significantly better than EPSO algorithm. In addition, the convergence progress of all the algorithms is also analysed for both benchmark sets and presented in the next section. The median performance of 30 trial runs is used to analyse the convergence characteristics of EPSO and other PSO algorithms.

4.3.2 Experimental Results

In this section, the 10 D and 30 D experimental results are presented for both benchmark sets and EPSO is compared with other PSO algorithms in terms of mean error values, statistical test results, and convergence test on 30 D problems. The com-

parison study on CEC2005 benchmark set is presented in Section 4.3.2.1 and the comparison study on CEC2013 benchmark set is shown in Section 4.3.2.2.

4.3.2.1 Comparison study on CEC2005 benchmark set

For the CEC2005 benchmark test suite, the 10 dimensional experiments are conducted using the population size 20 for all the algorithms, and EPSO is compared with its individual PSO variants and sHPSO algorithms. For 30 dimensional experiments, the population size 40 is used as mentioned in OLPSO simulation results. Besides the PSO algorithms compared in 10 dimensions, EPSO is also compared with OLPSO algorithm in 30 dimensions. The convergence characteristic graphs of 30 dimensional CEC2005 benchmark functions are presented in Appendix 3.

A. Results for 10 Dimensional Problems

The comparison study on 10 dimensional problems is illustrated in Table 4-2. The Wilcoxon signed rank test results of EPSO algorithm against its compared algorithms are also shown in Table 4-2 directly below the mean and standard deviation values. For unimodal problems ($F_1 \sim F_5$), the proposed Ensemble PSO outperforms its individual PSO variant and sHPSO algorithm on 4 out of 5 unimodal functions, except for FDR-PSO. Ensemble PSO performs better than FDR-PSO on function F_1 and similarly on function F_2 . FDR-PSO performs best on function F_3 , F_4 and F_5 .

For multimodal problems ($F_6 \sim F_{12}$), the proposed Ensemble PSO algorithm yields best results on function F_7 , F_9 , F_{10} and F_{12} . The EPSO outperforms PSO and LIPS on 5 out of 7 multimodal functions. The EPSO also performs better than HPSO-TVAC and sHPSO algorithms on 6 out of 7 functions. Compared to CLPSO, EPSO offers better results on function F_{10} , F_{11} and F_{12} , and similar results on functions F_6 and F_7 . CLPSO performs better on the two functions, F_8 and F_9 . EPSO outperforms FDR-PSO on 4 out of 7 multimodal functions.

For two expanded functions (F_{13} and F_{14}), EPSO outperforms all other PSO variants on both problems, except CLPSO which performs best on function F_{13} . For hybrid composite functions ($F_{15} \sim F_{25}$), the proposed EPSO algorithm performs better than LIPS, HPSO and sHPSO on all hybrid composite functions and FDR-PSO on 10

out 11 functions. EPSO provides better results than CLPSO on 4 out of 11 hybrid composite functions and PSO on 7 out of 11 hybrid composite functions. Overall, compared to its individual variants, EPSO consistently performs well on all four types of 10 dimensional benchmark functions.

Table 4-2 Comparison of EPSO vs Other PSOs on CEC2005 10 Dimensional Problems

F	Criteria	EPSO	PSO	LIPS	CLPSO	HPSO-TVAC	FDR-PSO	sHPSO
F_1	mean	0	0	0	0	0	0	0
	std	0	3.20E-14	0	0	1.17E-13	2.88E-14	7.22E+02
	h		+	=	=	+	+	+
F_2	mean	0	0	6.46E-07	1.86E-02	0	0	0
	std	6.46E-14	1.07E-13	3.54E-06	2.24E-02	7.52E-14	4.72E-13	2.69E+02
	h		+	+	+	+	=	+
F_3	mean	7.00E+04	1.60E+05	7.45E+05	3.49E+05	6.36E+04	3.60E+04	8.82E+04
	std	4.35E+04	1.10E+05	2.56E+06	2.21E+05	2.58E+04	3.81E+04	6.48E+04
	h		+	+	+	=	-	=
F_4	mean	1.97E-08	2.08E-13	1.16E+03	5.20	2.15E+02	3.74E-10	2.31E+03
	std	9.90E-08	1.19E-13	8.06E+02	11.02	1.97E-09	2.96E+02	2.23E+03
	h		-	+	+	+	-	+
F_5	mean	0.01	1.04E+03	2.60E+03	6.95	4.96E+02	1.19E-09	1.27E+03
	std	0.02	2.07E+02	8.97E+02	16.01	2.73E-09	3.61E+02	2.10E+03
	h		+	+	+	+	-	+
F_6	mean	1.43	3.47	13.58	0.86	11.83	0.54	786.51
	std	2.92	7.76	31.47	1.94	1.34	22.29	4017.41
	h		+	+	=	+	-	+
F_7	mean	0.15	0.22	3.67	0.21	2.56	0.20	11.58
	std	0.08	0.14	3.76	0.11	0.10	1.84	15.61
	rank		+	+	=	+	=	+
F_8	mean	20.34	20.29	20.32	20.27	20.17	20.30	20.14
	std	0.06	0.09	0.12	0.05	0.07	0.11	0.12
	h		-	=	-	-	-	-
F_9	mean	0.03	3.55	10.58	0	1.76	3.12	18.67
	std	0.18	1.40	4.30	0	1.824039	1.47	15.16
	h		+	+	-	+	+	+
F_{10}	mean	8.19	14.36	10.31	9.24	34.76	10.68	34.22
	std	3.03	5.21	5.61	2.97	3.99	11.37	12.14
	h		+	=	+	+	+	+
F_{11}	mean	3.21	3.36	3.97	4.53	5.91	2.27	6.92
	std	1.01	1.42	1.09	0.78	1.05	1.19	1.69
	h		=	+	+	+	-	+
F_{12}	mean	3.98E+01	1.83E+03	2.73E+03	7.22E+01	5.70E+02	1.56E+03	1.43E+03
	std	8.38E+01	4.31E+03	5.03E+03	5.20E+01	2.79E+03	7.52E+02	3.21E+03
	h		+	+	+	+	+	+
F_{13}	mean	0.47	0.63	0.82	0.28	0.66	0.65	1.61
	std	0.12	0.21	0.42	0.06	0.16	0.22	1.00
	h		+	-	-	+	+	+

F_{14}	mean	2.44	2.86	3.28	3.00	3.20	2.81	3.60
	std	0.49	0.41	0.49	0.27	0.65	0.35	0.41
	h		+	+	+	+	+	+
F_{15}	mean	9.40E+00	2.34E+02	2.17E+02	4.81E+00	2.15E+02	2.57E+02	3.82E+02
	std	2.34E+01	1.76E+02	1.05E+02	1.46E+01	2.36E+02	2.04E+02	1.93E+02
	h		+	+	=	+	+	+
F_{16}	mean	1.10E+02	1.29E+02	1.21E+02	1.20E+02	1.91E+02	1.33E+02	1.87E+02
	std	9.82E+00	1.64E+01	1.89E+01	9.92E+00	5.63E+01	3.91E+01	4.58E+01
	h		+	+	+	+	+	+
F_{17}	mean	1.12E+02	1.42E+02	1.38E+02	1.25E+02	1.95E+02	1.48E+02	2.05E+02
	std	1.00E+01	6.81E+01	4.21E+01	1.29E+01	9.13E+01	4.47E+01	7.31E+01
	h		+	+	+	+	+	+
F_{18}	mean	6.18E+02	7.10E+02	8.43E+02	6.85E+02	1.01E+03	8.79E+02	9.79E+02
	std	2.03E+02	2.53E+02	2.10E+02	1.87E+02	1.25E+02	8.67E+01	7.28E+01
	h		=	+	=	+	+	+
F_{19}	mean	6.45E+02	6.99E+02	8.76E+02	6.54E+02	9.95E+02	8.01E+02	9.89E+02
	std	2.11E+02	2.45E+02	1.75E+02	1.88E+02	2.20E+02	9.26E+01	1.42E+02
	h		=	+	=	+	+	+
F_{20}	mean	6.66E+02	6.16E+02	8.83E+02	7.46E+02	9.86E+02	8.35E+02	1.01E+03
	std	2.02E+02	2.66E+02	1.53E+02	1.62E+02	1.84E+02	1.12E+02	8.42E+01
	h		=	+	=	+	+	+
F_{21}	mean	3.87E+02	4.51E+02	7.37E+02	4.52E+02	1.04E+03	6.95E+02	1.09E+03
	std	1.22E+02	1.93E+02	2.76E+02	1.40E+02	3.70E+02	2.83E+02	2.23E+02
	h		=	+	+	+	+	+
F_{22}	mean	7.59E+02	7.64E+02	8.26E+02	7.27E+02	8.86E+02	7.82E+02	8.86E+02
	std	1.36E+01	9.30E+01	6.53E+01	1.42E+02	3.76E+01	1.30E+02	1.06E+02
	h		+	+	=	+	+	+
F_{23}	mean	5.91E+02	6.77E+02	9.64E+02	5.58E+02	1.12E+03	8.98E+02	1.17E+03
	std	9.55E+01	1.43E+02	2.51E+02	7.16E+01	2.36E+02	2.63E+02	1.69E+02
	h		+	+	=	+	+	+
F_{24}	mean	2.00E+02	2.80E+02	3.34E+02	2.00E+02	8.23E+02	4.33E+02	6.62E+02
	std	8.09E-13	1.56E+02	2.00E+02	1.63E-08	3.00E+02	4.59E+02	4.56E+02
	h		+	+	=	+	+	+
F_{25}	mean	2.00E+02	3.10E+02	4.36E+02	2.00E+02	7.87E+02	2.90E+02	7.38E+02
	std	8.15E-13	1.75E+02	3.72E+02	2.59E-09	1.95E+02	4.37E+02	4.85E+02
	h		+	+	+	+	=	+
+/-/-			19/4/2	22/3/0	9/11/3	23/1/1	16/3/6	23/1/1
Algorithms		EPSO	PSO	LIPS	CLPSO	HPSO-TVAC	FDR-PSO	sHPSO

B. Results for 30 Dimensional Problems

The comparison study on 30 dimensional problems is illustrated in Table 4-3 together with the Wilcoxon signed rank test results. For 30 dimensional unimodal problems, EPSO performs the best and outperforms compared PSO algorithms on all unimodal problems, except for function F_5 . Also on multimodal problems, EPSO performs better than PSO, LIPS, sHPSO and OLPSO on 5 out 7 and CLPSO, HPSO-TVAC on 4

out of 7 functions. EPSO yields similar results with FDR-PSO on five multimodal functions.

Except FDR-PSO, EPSO performs better than all other PSO variants on expanded functions. EPSO performs the best on 10 out of 11 hybrid composite functions. Therefore, overall, regardless of dimension size, EPSO performs consistently better than other PSO variants for solving different classes of optimization problems.

Table 4-3 Comparison of EPSO vs Other PSOs for CEC2005 30 Dimensional Problems

F	Criteria	EPSO	PSO	LIPS	CLPSO	HPSO-TVAC	FDR-PSO	sHPSO	OLPSO
F_1	mean	0	0	0	0	0	0	0	0
	std	0	0	0	0	0	0	0	0
	h		=	=	=	=	=	=	=
F_2	mean	0	0.37	83.52	1.14E+03	3.79E-06	0	1.44E-02	13.79
	std	1.88E-12	3.21E-01	326.6323	2.53E+02	2.82E-06	1.01E-11	7.10E-02	8.33
	h		+	+	+	+	+	+	+
F_3	mean	3.62E+05	6.53E+06	1.00E+07	1.22E+07	7.72E+05	4.14E+05	8.75E+05	1.60E+07
	std	1.35E+05	4.17E+06	1.35E+07	3.34E+06	2.96E+05	1.65E+05	5.34E+05	7.04E+06
	h		+	+	+	+	+	+	+
F_4	mean	2.38E+02	3.81E+02	3.32E+04	8.77E+03	2.48E+04	4.20E+02	2.02E+04	2.18E+03
	std	1.51E+02	3.31E+02	8.20E+03	1852.49	5.71E+03	2.47E+02	9.94E+03	1.09E+03
	h		+	+	+	+	+	+	+
F_5	mean	4.63E+03	3.85E+03	9.35E+03	4.47E+03	9.20E+03	3.20E+03	6.94E+03	3.30E+03
	std	8.08E+02	8.00E+02	1.75E+03	426.30	1.81E+03	6.20E+02	1.43E+03	3.75E+02
	h		+	+	=	+	-	+	-
F_6	mean	3.36	70.16	106.33	2.39	50.41	3.55	115.27	20.68
	std	4.15	95.09	205.89	3.84	50.54	7.04	228.56	24.97
	h		+	+	=	+	=	+	+
F_7	mean	0.02	0.76	85.38	0.70	0.01	0.02	0.04	0.01
	std	0.01	1.41	45.37	0.15	0.01	0.01	0.04	0.01
	rank		+	+	+	-	=	=	=
F_8	mean	20.91	20.90	20.92	20.92	20.71	20.88	20.18	20.96
	std	0.08	0.07	0.12	0.06	0.15	0.07	0.19	0.08
	h		=	=	=	-	=	-	+
F_9	mean	0	19.00	49.49	0	10.71	30.25	82.54	0
	std	0	5.37	12.40	0	4.96	8.63	24.35	0
	h		+	+	=	+	+	+	=
F_{10}	mean	54.56	100.32	67.16	99.78	275.73	55.63	242.83	109.23
	std	16.98	58.63	19.23	12.48	45.49	18.68	89.17	18.94
	h		+	+	+	+	=	+	+
F_{11}	mean	22.23	23.18	21.08	24.42	30.62	16.92	32.29	25.05
	std	2.58	2.81	2.68	1.78	2.58	3.72	3.84	3.14
	h		=	=	+	+	-	+	+

F_{12}	mean	4.10E+03	5.07E+04	8.01E+04	1.47E+04	4.47E+03	9.52E+03	2.43E+04	1.22E+04
	std	4.19E+03	3.50E+04	7.02E+04	3.45E+03	4.66E+03	1.53E+04	2.66E+04	5.42E+03
	h		+	+	+	=	-	+	+
F_{13}	mean	1.80	3.02	4.08	1.86	3.85	2.97	6.14	1.86
	std	0.43	0.69	1.20	0.18	1.11	0.64	2.31	0.28
	h		+	+	=	+	+	+	=
F_{14}	mean	11.86	12.65	12.35	12.64	12.31	11.66	13.07	13.13
	std	0.77	0.43	0.50	0.22	0.37	0.58	0.39	0.20
	h		+	+	+	+	=	+	+
F_{15}	mean	56.00	299.71	379.36	40.10	321.29	306.28	457.16	244.88
	std	62.90	107.62	88.12	37.90	107.12	150.98	138.94	84.09
	h		+	+	=	+	+	+	+
F_{16}	mean	104.18	235.84	164.23	163.61	336.21	293.38	376.70	135.54
	std	34.19	152.98	93.55	27.87	97.20	182.92	99.46	55.48
	h		+	+	+	+	+	+	+
F_{17}	mean	106.28	262.63	288.11	219.31	391.68	280.55	367.25	204.56
	std	46.50	153.27	92.59	40.09	97.34	192.62	133.86	46.67
	h		+	+	+	+	+	+	+
F_{18}	mean	905.74	927.77	969.31	907.76	908.24	918.16	997.99	906.63
	std	36.00	2.86	16.05	26.75	79.00	3.47	60.04	20.23
	h		+	+	=	=	=	+	-
F_{19}	mean	901.87	926.69	971.95	915.09	912.45	916.00	982.94	912.45
	std	40.82	2.49	16.13	1.36	79.78	4.72	78.69	79.78
	h		+	+	=	=	=	+	-
F_{20}	mean	890.17	927.06	973.73	911.24	905.92	913.53	974.13	905.92
	std	50.69	2.35	14.53	20.57	74.72	21.79	67.96	74.72
	h		+	+	=	=	=	+	=
F_{21}	mean	496.98	510.00	520.00	500.00	1173.50	539.42	996.01	500.00
	std	16.55	54.77	76.11	0.00	229.74	141.69	328.97	0
	h		=	+	=	+	=	+	+
F_{22}	mean	923.34	938.66	1009.91	974.32	1216.89	927.56	1123.41	941.70
	std	21.97	12.96	29.64	17.22	73.68	17.45	103.68	16.30
	h		+	+	+	+	=	+	+
F_{23}	mean	534.16	574.17	647.24	534.16	1258.30	559.82	996.79	534.16
	std	3.57E-04	122.10	155.06	2.33E-04	34.57	100.45	323.48	5.04E-04
	h		+	+	=	+	+	+	=
F_{24}	mean	200.00	200.00	357.67	200.00	1276.11	210.00	995.63	200.00
	std	0	0	323.80	0	205.10	54.77	409.92	0
	h		=	+	=	+	+	+	=
F_{25}	mean	200.00	1113.45	411.68	200.00	1331.03	200.00	1037.43	207.46
	std	8.18E-13	1.44E+01	3.73E+02	5.07E-11	2.60E+01	1.21E-12	423.59	7.09
	h		+	+	+	+	+	+	+
+/-/-			20/5/0	22/3/0	12/13/0	18/5/2	12/11/2	22/2/1	15/7/3
Algorithms		EPSO	PSO	LIPS	CLPSO	HPSO-TVAC	FDR-PSO	sHPSO	OLPSO

(a) F_1

(b) F_4

Figure 4-3 Convergence characteristic graphs of CEC2005 30 dimensional unimodal functions:
(a) F_1 : Shifted Sphere Function and (b) F_4 : Shifted Schwefel's Problem 1.2 with Noise in Fitness

(a) F_6

(b) F_{12}

Figure 4-4 Convergence characteristic graphs of CEC2005 30 dimensional multimodal functions:
(a) F_6 : Shifted Rosenbrock's Function and (b) F_{12} : Schwefel's Problem 2.13

Figure 4-5 Convergence characteristic graphs of CEC2005 30 dimensional expanded extended functions: (a) F_{13} : Expanded Extended Griewank's plus Rosenbrock's Function (F_8F_2) and (b) F_{14} : Shifted Rotated Expanded Scaffer's F_6

Figure 4-6 Convergence characteristic graphs of CEC2005 30 dimensional hybrid composition functions: (a) F_{16} : Rotated Hybrid Composition Function and (b) F_{23} : Non-Continuous Rotated Hybrid Composition Function

4.3.2.2 Comparison study on CEC2013 benchmark set

For the CEC2013 benchmark test suite, the experiments are conducted using the population size 50 for both 10 and 30 dimensional problems as described in self-adaptive heterogeneous algorithm [37]. Each algorithm is executed on each function for 51 trials in 10D and 30D for (10000*D) number of function evaluations.

The mean and standard deviation values are only presented in the self-adaptive heterogeneous PSO (f_k -PSO) paper [37] and the results for 51 trials are not available to conduct the statistical analysis test between the proposed EPSO algorithm and f_k -PSO algorithm. Therefore, the ranking of the algorithms are determined based on their mean solution values and their average ranks over all the problems as well as their final ranks are presented in the comparison tables (Table 4-4 and Table 4-5).

A. Results for 10 dimensional problems

Similar to CEC2005 benchmark set, the recent CEC2013 benchmark set includes five unimodal problems. In addition to similar Sphere function (F_1) and High Conditioned Elliptic Function (F_3), the recent CEC2013 benchmark set introduces three new unimodal functions: Rotated Bent Cigar function (F_2), Rotated Discus function (F_4) and Different Powers function (F_5). As shown in Table 4-4, all algorithms perform well on simple unimodal function (F_1) and EPSO, CLPSO, f_k -PSO and LIPS algorithms are able to find the mean solution values equal to the global optimum. The proposed EPSO algorithm offers the best performance on 4 out of 5 unimodal problems, except on function (F_3). The f_k -PSO algorithm performs the best on the unimodal function (F_3).

For the multimodal problems ($F_6 \sim F_{18}$), seven new multimodal functions are introduced in the CEC2013 competition on top of simple six multimodal functions that are similar to the CEC2005 benchmark set,. All algorithms perform well on the 10 dimensional multimodal problems, except traditional PSO, HPSO-TVAC, and sHPSO algorithms. The proposed EPSO algorithm performs the best on functions F_6 and F_{11} while f_k -PSO algorithm yields the best results on functions F_7 , F_8 and F_{16} . FDR-PSO algorithm provides the best performance on functions F_9 and F_{10} . CLPSO algorithm offers the best on functions F_{11} , F_{14} , and F_{17} while LIPS algorithm performs the best on functions F_{12} , F_{15} , F_{16} , and F_{18} . For the two multimodal expanded functions, CLPSO and the proposed EPSO algorithms perform the best on functions F_{19} and F_{20} , respectively.

Compared to CEC2005 benchmark composition problems, the new CEC2013 compositions merge the properties of base functions better and maintain continuity around the global/local optima [211]. For the eight composition functions ($F_{21} \sim F_{28}$), the proposed EPSO algorithm performs better than compared PSO algorithms. The

EPSO algorithm performs the best on functions F_{21} , $F_{24} \sim F_{26}$ and F_{28} , on 5 out of 8 functions. On the remaining three functions (F_{22} , F_{23} and F_{27}), CLPSO, f_k -PSO and FDR-PSO algorithms provide the best results while EPSO is the second best, followed by PSO, HPSO-TVAC, and sHPSO algorithms.

The Wilcoxon signed rank test, the average and final ranks of all the algorithms are shown in the last rows of the comparison Table 4-4. It can be observed that the proposed EPSO algorithm performs better than other PSO algorithms on most of the problems. At the same time, the EPSO algorithm outranks the recent f_k -PSO algorithm in terms of the average ranks over all the problems.

Table 4-4 Comparison of EPSO vs other PSOs on CEC2013 10 dimensional problems

F	Criteria	EPSO	PSO	LIPS	CLPSO	HPSO-TVAC	FDR-PSO	sHPSO	f_k -PSO
F_1	mean	0.00E+00	8.92E-14	0.00E+00	0.00E+00	4.95E-13	4.01E-14	4.46E-14	0.00E+00
	std	0.00E+00	1.12E-13	0.00E+00	0.00E+00	6.85E-13	8.75E-14	9.12E-14	0.00E+00
	rank	1	7	1	1	8	5	6	1
	h		=	=	=	+	+	+	
F_2	mean	2.74E+04	3.64E+05	2.83E+05	1.05E+06	5.19E+04	4.05E+04	5.88E+04	1.44E+05
	std	1.98E+04	3.97E+05	2.06E+05	5.31E+05	4.57E+04	3.98E+04	5.67E+04	1.04E+05
	rank	1	7	6	8	3	2	4	5
	h		+	+	+	+	+	+	
F_3	mean	2.37E+06	7.33E+05	3.84E+06	3.27E+06	1.18E+08	7.23E+05	4.19E+07	6.75E+05
	std	4.88E+06	1.86E+06	7.50E+06	3.25E+06	2.07E+08	2.76E+06	8.65E+07	1.96E+06
	rank	4	3	6	5	8	2	7	1
	h		—	+	+	+	+	+	
F_4	mean	2.92E+02	1.02E+03	1.48E+04	7.81E+03	3.12E+03	1.84E+03	2.17E+03	4.16E+02
	std	1.89E+02	6.13E+02	5.69E+03	3.10E+03	1.96E+03	1.47E+03	2.30E+03	3.37E+02
	rank	1	3	8	7	6	4	5	2
	h		+	+	+	+	+	+	
F_5	mean	0.00E+00	1.11E-13	8.60E-09	7.58E-14	8.67E-12	8.02E-14	1.27E-13	0.00E+00
	std	0.00E+00	4.82E-14	3.39E-08	5.41E-14	2.21E-11	6.14E-14	6.69E-14	0.00E+00
	rank	1	5	8	3	7	4	6	1
	h		+	+	+	+	+	+	
F_6	mean	0.34	7.93	6.40	2.53	3.04	4.48	7.43	2.64
	std	1.36	10.69	11.43	2.65	9.30	4.88	11.59	2.05
	rank	1	8	6	2	4	5	7	3
	h		+	+	+	+	+	+	
F_7	mean	3.62	3.43	16.05	16.72	39.18	5.00	39.79	1.92
	std	4.63	3.74	1.01	5.54	1.55	7.47	2.02	2.70
	rank	3	2	5	6	7	4	8	1
	h		0	+	+	+	=	+	

F_8	mean	20.36	20.31	20.35	20.37	20.34	20.32	20.36	20.30
	std	0.07	0.07	0.07	0.07	0.07	0.08	0.11	0.09
	rank	6	2	5	8	4	3	7	1
	h		–	=	=	=	=	=	
F_9	Mean	3.33	3.28	3.07	4.50	5.54	2.47	5.26	2.75
	Std	0.99	1.37	0.89	0.79	1.05	1.06	1.63	1.09
	rank	5	4	3	6	8	1	7	2
	h		=	=	+	+	–	+	
F_{10}	mean	0.46	0.49	0.39	1.02	4.26	0.35	1.42	0.51
	std	0.30	0.32	0.21	0.19	2.21	0.20	1.18	0.32
	rank	3	4	2	6	8	1	7	5
	h		=	=	+	+	–	+	
F_{11}	mean	0.00E+00	2.17E+00	7.32E+00	0.00E+00	1.50E+00	2.15E+00	6.52E+00	1.76E-01
	std	0.00E+00	1.30E+00	3.26E+00	0.00E+00	1.66E+00	1.86E+00	5.11E+00	3.79E-01
	rank	1	6	8	1	4	5	7	3
	h		+	+	=	+	+	+	
F_{12}	mean	9.57	13.81	6.47	12.42	34.18	10.79	21.78	7.04
	std	3.58	6.05	2.63	3.15	13.01	4.77	9.21	2.95
	rank	3	6	1	5	8	4	7	2
	h		+	–	+	+	=	+	
F_{13}	mean	14.28	21.23	17.39	16.56	40.22	14.54	33.89	11.50
	std	5.92	6.39	7.04	5.25	1.18E	5.56	1.14	4.77
	rank	2	6	5	4	8	3	7	1
	h		+	+	+	+	=	+	
F_{14}	mean	9.45	131.60	363.28	0.11	245.54	271.23	299.44	37.80
	std	9.08	104.57	124.32	0.06	123.27	129.54	150.75	42.10
	rank	2	4	8	1	5	6	7	3
	h		+	+	–	+	+	+	
F_{15}	mean	5.34E+02	8.48E+02	4.28E+02	7.98E+02	6.51E+02	6.26E+02	9.75E+02	4.54E+02
	std	2.11E+02	2.60E+02	1.88E+02	1.49E+02	2.90E+02	2.53E+02	3.04E+02	1.67E+02
	rank	3	7	1	6	5	4	8	2
	h		+	+	+	+	=	+	
F_{16}	mean	1.00	0.96	0.22	0.96	0.60	0.89	0.62	0.41
	std	0.24	0.22	0.11	0.21	0.21	0.23	0.34	0.14
	rank	8	7	1	6	3	5	4	2
	h		0	–	=	–	=	–	
F_{17}	mean	12.15	10.68	14.48	10.23	12.22	13.59	18.18	11.00
	std	3.23	5.32	1.67	0.06	1.46	1.80	6.44	3.28
	rank	4	2	7	1	5	6	8	3
	h		0	+	–	=	+	+	
F_{18}	mean	26.41	28.08	14.21	30.71	32.46	23.15	31.80	15.60
	std	8.72	9.72	1.96	3.90	9.02	6.62	12.47	2.33
	rank	4	5	1	6	8	3	7	2
	h		0	–	+	+	–	+	
F_{19}	mean	0.59	0.63	0.56	0.23	1.18	0.68	1.25	0.50
	std	0.15	0.18	0.13	0.11	0.67	0.22	0.84	0.14
	rank	4	5	3	1	7	6	8	2
	h		0	=	–	+	+	+	

F_{20}	mean	2.22	2.94	2.97	3.00	2.96	2.65	3.38	2.52
	std	0.52	0.71	0.47	0.27	0.45	0.80	0.56	0.48
	rank	1	4	6	7	5	3	8	2
	h		+	+	+	+	+	+	
F_{21}	mean	1.85E+02	3.41E+02	3.35E+02	2.08E+02	3.94E+02	4.00E+02	3.84E+02	3.75E+02
	std	9.11E+01	9.84E+01	9.77E+01	8.59E+01	4.20E+01	5.57E-14	6.13E+01	7.10E+01
	rank	1	4	3	2	7	8	6	5
	h		+	+	+	+	+	+	
F_{22}	mean	3.81E+01	1.89E+02	5.34E+02	1.91E+01	2.42E+02	2.67E+02	3.53E+02	1.22E+02
	std	2.65E+01	1.05E+02	1.88E+02	1.37E+01	1.74E+02	1.82E+02	1.59E+02	7.57E+01
	rank	2	4	8	1	5	6	7	3
	h		+	+	-	+	+	+	
F_{23}	mean	6.22E+02	7.52E+02	7.55E+02	9.27E+02	9.90E+02	7.62E+02	1.33E+03	5.15E+02
	std	2.73E+02	2.90E+02	1.96E+02	1.81E+02	3.78E+02	3.47E+02	3.48E+02	1.97E+02
	rank	2	3	4	6	7	5	8	1
	h		+	+	+	+	+	+	
F_{24}	mean	1.23E+02	2.11E+02	1.28E+02	1.31E+02	2.15E+02	2.08E+02	2.18E+02	2.03E+02
	std	2.20E+01	5.03E+00	3.42E+01	9.89E+00	2.04E+01	6.23E+00	4.26E+00	1.91E+01
	rank	1	6	2	3	7	5	8	4
	h		+	=	+	+	+	+	
F_{25}	mean	1.72E+02	2.07E+02	1.86E+02	1.83E+02	2.14E+02	2.07E+02	2.09E+02	2.05E+02
	std	4.18E+01	1.97E+01	3.85E+01	2.86E+01	1.75E+01	5.37E+00	2.16E+01	1.44E+01
	rank	1	6	3	2	8	5	7	4
	h		+	+	+	+	+	+	
F_{26}	mean	1.12E+02	1.75E+02	1.14E+02	1.21E+02	1.60E+02	1.84E+02	1.72E+02	1.89E+02
	std	6.34E+00	6.99E+01	1.78E+01	5.51E+00	3.01E+01	8.37E+01	5.37E+01	5.13E+01
	rank	1	6	2	3	4	7	5	8
	h		+	=	+	+	+	+	
F_{27}	mean	3.28E+02	4.24E+02	3.42E+02	3.53E+02	4.40E+02	3.28E+02	4.47E+02	3.70E+02
	std	2.43E+01	1.14E+02	4.64E+01	2.07E+01	7.26E+01	3.90E+01	9.12E+01	3.22E+01
	rank	2	6	3	4	7	1	8	5
	h		+	+	+	+	=	+	
F_{28}	mean	1.97E+02	2.84E+02	3.03E+02	2.08E+02	6.08E+02	3.40E+02	4.31E+02	3.26E+02
	std	9.89E+01	5.43E+01	6.88E+01	7.43E+01	2.05E+02	8.35E+01	2.02E+02	1.25E+02
	rank	1	3	4	2	8	6	7	5
	h		+	+	+	+	+	+	
Average rank		2.46	4.82	4.29	4.04	6.21	4.25	6.82	2.82
Final rank		1	6	5	3	7	4	8	2
+/-			18/8/2	18/7/3	20/4/4	25/2/1	18/6/4	26/1/1	
Algorithms		EPSO	PSO	LIPS	CLPSO	HPSO-TVAC	FDR-PSO	sHPSO	f_k -PSO

B. Results for 30 dimensional problems

The performances of all algorithms on 30 dimensional problems are similar to their performances on 10 dimensional problems. For 30 dimensional unimodal prob-

lems, except on function F_3 , the proposed EPSO algorithm performs the best on 4 out of 5 unimodal functions as in previous 10D experiments. Unlike previous 10D experiments, the performance of f_k -PSO algorithm is decreased and FDR-PSO algorithm offers the best performance on function F_3 .

All algorithms perform fairly on 30D problems, except PSO, HPSO-TVAC, and sHPSO algorithms. Out of 13 multimodal problems, the proposed EPSO algorithm achieves the best performance on the three multimodal functions F_6 , F_{12} , and F_{13} . Similarly, FDR-PSO offers the best on functions F_7 , F_9 , and F_{10} while CLPSO yields the best results on F_{11} , F_{14} , and F_{17} . For the remaining four functions, f_k -PSO obtains the best results on functions F_8 and F_{18} and LIPS does on functions F_{15} and F_{16} . However, overall the proposed EPSO algorithm ranks within top 1 to 3 for most problems.

For the 30D expanded functions, the proposed EPSO algorithm performs the best and the second best on the expanded function F_{19} and F_{20} respectively. The EPSO algorithm also offers consistent and robust performance and ranks within top 1 to 3 on the 30 dimensional composition functions. In terms of the average ranks values over all the problems, EPSO ranks first while f_k -PSO is the second best, followed by FDR-PSO, CLPSO, LIPS, PSO, HPSO-TVAC, and sHPSO algorithms. In addition, the statistical analysis results (+/0/-)/(better/the same/worse) affirmed the findings that EPSO outperforms its individual PSO variants and the sHPSO algorithm. Therefore, overall the proposed EPSO algorithm performs significantly better on all types of CEC2013 benchmark problems.

For 30 dimensional problems, the convergence performance of EPSO and other PSO algorithms is also analysed using the mean performance of 51 runs for each function and shown in Figures 4-7, 4-8 and 4-9 for some unimodal, multimodal, expanded and composition functions. It can be observed from the figures that EPSO has a superior convergence performance compared to other PSO variants.

Table 4-5 Comparison of EPSO vs other PSOs on CEC2013 30 dimensional problems

F	Criteria	EPSO	PSO	LIPS	CLPSO	HPSO-TVAC	FDR-PSO	sHPSO	f_k -PSO
F_I	mean	0.00E+00	5.35E-13	4.32E-13	2.18E-13	3.24E-12	4.46E-13	1.68E-07	0.00E+00
	std	0.00E+00	1.63E-13	4.05E-13	4.46E-14	2.92E-12	1.01E-13	1.18E-06	0.00E+00
	rank	1	6	4	3	7	5	8	1
	h		+	+	=	+	+	+	

F_2	mean	1.91E+05	1.27E+07	7.51E+06	1.61E+07	8.78E+05	2.91E+05	8.70E+05	1.59E+06
	std	7.66E+04	7.60E+06	1.16E+07	3.89E+06	3.42E+05	1.30E+05	3.68E+05	8.03E+05
	rank	1	7	6	8	4	2	3	5
	h		+	+	+	+	+	+	
F_3	mean	1.20E+08	8.87E+07	5.10E+08	5.01E+08	7.97E+08	2.02E+07	4.64E+08	2.40E+08
	std	2.14E+08	1.10E+08	6.39E+08	2.94E+08	1.01E+09	2.87E+07	5.20E+08	3.71E+08
	rank	3	2	7	6	8	1	5	4
	h		−	+	+	+	−	+	
F_4	mean	2.28E+02	3.78E+03	7.35E+04	3.19E+04	5.54E+03	1.51E+03	3.41E+03	4.78E+02
	std	1.30E+02	9.60E+02	1.66E+04	5.55E+03	2.43E+03	7.04E+02	3.34E+03	1.96E+02
	rank	1	5	8	7	6	3	4	2
	h		+	+	+	+	+	+	
F_5	mean	0.00E+00	5.86E-13	2.72E-09	3.57E-13	8.29E-11	4.10E-13	2.83E-08	0.00E+00
	std	0.00E+00	1.23E-13	7.47E-09	5.09E-14	2.29E-10	1.02E-13	2.02E-07	0.00E+00
	rank	1	5	7	3	6	4	8	1
	h		+	+	+	+	+	+	
F_6	mean	11.18	79.13	89.35	25.27	34.88	21.97	33.76	29.90
	std	10.80	36.50	49.97	4.97	26.98	23.22	27.17	17.60
	rank	1	7	8	3	6	2	5	4
	h		+	+	+	+	+	+	
F_7	mean	37.77	39.13	76.77	81.35	129.93	26.76	136.66	63.90
	std	13.97	12.53	19.82	11.46	42.76	14.26	49.71	30.90
	rank	2	3	5	6	7	1	8	4
	h		=	+	+	+	−	+	
F_8	mean	20.95	20.94	20.95	20.94	20.93	20.93	20.97	20.90
	std	0.05	0.06	0.05	0.06	0.07	0.06	0.08	0.06
	rank	7	4	6	5	3	2	8	1
	h		=	=	=	=	=	=	
F_9	Mean	22.16	21.80	23.18	26.72	31.36	17.89	32.63	18.50
	Std	2.77	3.01	2.41	1.62	2.27	2.99	3.71	2.69
	rank	4	3	5	6	7	1	8	2
	h		=	+	+	+	−	+	
F_{10}	mean	0.19	0.18	0.19	4.33	0.64	0.13	0.16	0.23
	std	0.15	0.12	0.21	1.09	0.40	0.08	0.12	0.13
	rank	5	3	4	8	7	1	2	6
	h		=	=	+	+	−	=	
F_{11}	mean	0.28	20.21	63.60	8.81E-14	10.48	28.19	87.69	23.60
	std	0.53	5.15	18.64	2.86E-14	4.82	7.88	33.13	8.76
	rank	2	4	7	1	3	6	8	5
	h		+	+	−	+	+	+	
F_{12}	mean	50.20	77.67	63.39	114.58	312.71	51.55	262.13	56.40
	std	13.57	37.08	13.99	16.97	64.95	15.25	92.25	15.10
	rank	1	5	4	6	8	2	7	3
	h		+	+	+	+	=	+	
F_{13}	mean	1.16E+02	1.53E+02	1.43E+02	1.68E+02	3.81E+02	1.20E+02	2.83E+02	1.23E+02
	std	2.99E+01	3.47E+01	3.24E+01	1.63E+01	6.13E+01	2.23E+01	6.14E+01	2.19E+01
	rank	1	5	4	6	8	2	7	3
	h		+	+	+	+	=	+	

F_{14}	mean	4.37E+01	9.18E+02	2.10E+03	4.45E+00	8.95E+02	1.07E+03	1.94E+03	7.04E+02
	std	2.44E+01	2.72E+02	3.26E+02	2.42E+00	2.12E+02	2.56E+02	4.96E+02	2.38E+02
	rank	2	5	8	1	4	6	7	3
	h		+	+	-	+	+	+	
F_{15}	mean	3.44E+03	6.59E+03	3.00E+03	4.38E+03	4.07E+03	3.77E+03	4.38E+03	3.42E+03
	std	5.74E+02	8.54E+02	4.59E+02	4.18E+02	8.19E+02	8.91E+02	5.58E+02	5.16E+02
	rank	3	8	1	6	5	4	7	2
	h		+	-	+	+	+	+	
F_{16}	mean	1.81	2.27	0.49	1.73	1.40	2.04	1.47	0.85
	std	0.34	0.36	0.16	0.29	0.43	0.38	0.59	0.22
	rank	6	8	1	5	3	7	4	2
	h		+	-	=	-	+	-	
F_{17}	mean	40.37	48.22	89.19	31.46	39.82	68.86	129.11	52.60
	std	5.20	17.93	21.08	0.30	5.39	9.20	49.94	7.11
	rank	3	4	7	1	2	6	8	5
	h		+	+	-	=	+	+	
F_{18}	mean	82.65	214.78	88.24	198.79	323.77	146.55	209.69	68.10
	std	32.65	41.06	15.90	11.87	53.23	62.11	66.81	9.68
	rank	2	7	3	5	8	4	6	1
	h		+	+	+	+	+	+	
F_{19}	mean	1.96	3.25	7.26	0.73	5.87	2.98	13.84	3.12
	std	0.45	1.00	2.37	0.23	1.97	0.77	6.98	0.98
	rank	2	5	7	1	6	3	8	4
	h		+	+	-	+	+	+	
F_{20}	mean	10.71	14.08	12.26	13.80	13.36	14.51	14.72	12.00
	std	0.75	1.39	0.76	0.69	1.02	1.35	0.50	0.93
	rank	1	6	3	5	4	7	8	2
	h		+	+	+	+	+	+	
F_{21}	mean	2.63E+02	2.81E+02	2.16E+02	2.71E+02	3.11E+02	3.18E+02	3.20E+02	3.11E+02
	std	4.80E+01	7.13E+01	6.25E+01	4.14E+01	8.95E+01	9.19E+01	9.59E+01	7.92E+01
	rank	2	4	1	3	6	7	8	5
	h		+	+	+	+	+	+	
F_{22}	mean	1.64E+02	9.36E+02	2.85E+03	1.21E+02	7.01E+02	1.14E+03	2.09E+03	8.59E+02
	std	4.98E+01	3.02E+02	4.87E+02	1.75E+01	2.45E+02	2.51E+02	5.93E+02	3.10E+02
	rank	2	5	8	1	3	6	7	4
	h		+	+	-	+	+	+	
F_{23}	mean	3.98E+03	6.43E+03	4.03E+03	5.19E+03	4.98E+03	4.07E+03	6.04E+03	3.57E+03
	std	7.34E+02	1.33E+03	6.27E+02	3.71E+02	7.45E+02	7.91E+02	9.07E+02	5.90E+02
	rank	2	8	3	6	5	4	7	1
	h		+	=	+	+	+	+	
F_{24}	mean	2.48E+02	2.65E+02	2.65E+02	2.76E+02	2.96E+02	2.41E+02	3.01E+02	2.48E+02
	std	1.10E+01	9.78E+00	2.52E+01	4.62E+00	9.94E+00	1.54E+01	1.18E+01	8.11E+00
	rank	3	4	5	6	7	1	8	2
	h		+	+	+	+	-	+	

F_{25}	mean	2.81E+02	2.81E+02	3.02E+02	2.94E+02	2.96E+02	2.74E+02	3.28E+02	2.49E+02
	std	1.23E+01	8.19E+00	9.11E+00	4.73E+00	9.91E+00	1.94E+01	1.34E+01	7.82E+00
	rank	3	4	7	5	6	2	8	1
	h		=	+	+	+	=	+	
F_{26}	mean	2.00E+02	3.16E+02	2.38E+02	2.01E+02	2.38E+02	3.09E+02	3.33E+02	2.95E+02
	std	8.52E-03	6.88E+01	6.41E+01	4.27E-01	7.36E+01	6.18E+01	8.33E+01	7.06E+01
	rank	1	7	3	2	4	6	8	5
	h		+	+	+	+	+	+	
F_{27}	mean	7.11E+02	8.66E+02	9.45E+02	6.13E+02	1.19E+03	7.59E+02	1.15E+03	7.76E+02
	std	1.93E+02	9.38E+01	6.14E+01	2.64E+02	8.75E+01	1.42E+02	9.16E+01	7.11E+01
	rank	2	5	6	1	8	3	7	4
	h		+	+	-	+	+	+	
F_{28}	mean	2.80E+02	3.91E+02	3.37E+02	3.00E+02	3.08E+03	3.48E+02	1.15E+03	4.01E+02
	std	6.01E+01	3.16E+02	1.99E+02	4.12E-04	7.96E+02	2.68E+02	9.64E+02	3.48E+02
	rank	1	5	3	2	8	4	7	6
	h		+	+	+	+	+	+	
Average rank		2.32	5.14	5.04	4.21	5.68	3.64	6.75	3.14
Final rank		1	6	5	4	7	3	8	2
+/-/-			22/5/1	24/3/2	19/3/6	25/2/1	19/4/5	25/2/1	
Algorithms		EPSO	PSO	LIPS	CLPSO	HPSO-TVAC	FDR-PSO	sHPSO	f_k -PSO

(a) F_2 (b) F_4

Figure 4-7 Convergence characteristic graphs of CEC2013 30 dimensional unimodal functions:

(a) F_2 : Rotated High Conditioned Elliptic Function and (b) F_4 : Rotated Discus Function

Figure 4-8 Convergence characteristic graphs of CEC2013 30 dimensional multimodal functions: (a) F_6 : Rotated Rosenbrock's Function and (b) F_{14} : Schwefel's Function

Figure 4-9 Convergence characteristic graphs of CEC2013 30 dimensional expanded functions: (a) F_{20} : Expanded Scaffer's F_6 Function and composition function: (b) F_{28} : Schwefel's Function

4.3.3 Algorithm Computational Complexity

The computational complexity of the proposed EPSO algorithm is calculated as defined in the CEC2013 benchmark competition [211] and presented in this section. In addition, the algorithm complexity of its individual PSO variants is also calculated and compared with the algorithm complexity of EPSO algorithm. All the experiments are conducted on the following system:

- OS: Windows 7
- CPU: Intel(R) Xeon (R) CPU E3-1226 v3@3.30GHz
- RAM: 8GB
- Language: MatLab 2015a

Table 4-6 shows the computation complexity of EPSO algorithm on 10 and 30 dimensions. In Table 4-6, T_0 is the time computed by running the following codes from [211]:

```
for i=1:1000000
x= 0.55 + (double) i; x=x + x; x=x./2; x=x*x;
x=sqrt(x); x=log(x); x=exp(x); y=x/x;
end
```

T_1 is the time to execute 200,000 evaluations of benchmark function F_{14} by itself with D dimensions. T_2 is the execution time of EPSO algorithm on the same function F_{14} with the same number of function evaluation 200,000 in D dimensions. The time $\hat{T}_2$ is the mean values of T_2 over five runs. As seen in Table 4-6, both T_1 , $\hat{T}_2$ and $(\hat{T}_2 - T_1)/T_0$ scaled linearly with the number of dimensions.

The computation complexity comparisons of EPSO algorithm and its individual PSO variants are shown in Table 4-7 for 10D and Table 4-8 for 30D. The same procedure and the same system are used to compute the computation complexity of the individual PSO variants of the proposed EPSO algorithm. It can be observed from the tables that the computational cost of EPSO algorithm is lower than LISP and FDR-PSO algorithms and higher than PSO, CLPSO, and HPSO-TVAC.

However, as shown in the performance comparison, Table 4-4 and Table 4-6, the performance of EPSO is significantly better than PSO, HPSO-TVAC and CLPSO algorithm. Moreover, overall the proposed EPSO algorithm performs consistently well on various types of problems. Therefore, the computation cost of EPSO algorithm is

reasonable and affordable compared to the computational cost of its individual PSO variants.

Table 4-6 Computational Complexity of EPSO algorithm

	T_0	T_1	$\hat{T}_2$	$(\hat{T}_2 - T_1)/T_0$
10D	0.1881	0.5549	4.0275	18.4615
30D	0.1881	0.9682	6.3019	28.3557

Table 4-7 Comparison of Computational Complexity of EPSO vs its PSO variants on 10 D

10 D	T_0	T_1	$\hat{T}_2$	$(\hat{T}_2 - T_1)/T_0$
EPSO	0.1881	0.5549	4.0275	18.4615
PSO	0.1881	0.5549	2.7271	11.5481
LIPS	0.1881	0.5549	6.0542	29.2360
FDR	0.1881	0.5549	4.9655	23.4482
CLPSO	0.1881	0.5549	2.8495	12.1988
HPSO-TVAC	0.1881	0.5549	0.5948	0.2121

Table 4-8 Comparison of Computational Complexity of EPSO vs its PSO variants on 30 D

30 D	T_0	T_1	$\hat{T}_2$	$(\hat{T}_2 - T_1)/T_0$
EPSO	0.1881	0.9682	6.3019	28.3557
PSO	0.1881	0.9682	3.8857	15.5104
LIPS	0.1881	0.9682	7.4823	34.6310
FDR	0.1881	0.9682	8.3246	39.1090
CLPSO	0.1881	0.9682	3.8636	15.3929
HPSO-TVAC	0.1881	0.9682	1.2681	1.5944

4.3.4 Different search behaviours of EPSO on different benchmark problems

In this section, the different search behaviour profiles of EPSO algorithm on different benchmark problems are discussed together with their 3D maps for 2D functions. The 3D maps of the benchmark problems are obtained from [195] and [211]. The probability values (pk) of the PSO strategies in EPSO are plotted against the number of generations to show the search behaviour profiles of the proposed EPSO algorithm.

The average behaviour profiles of 30 trial runs for CEC2005 benchmark problems are shown in Figures 4-10 and 4-11. For the recent CEC2013 benchmark problems, the average behaviour profiles of 51 trial runs are illustrated in Figures 4-12 and 4-13.

In the proposed EPSO algorithm, the traditional PSO, HPSO-TVAC, FDR-PSO, LIPS and CLPSO strategies are employed through the self-adaptive evaluation strategy. The traditional PSO is used to ensure fast converge speed. HPSO-TVAC re-initializes the particles whenever they are stagnated in the search space and uses time-varying acceleration coefficients to enhance the global search in the early stage of the optimization phase and encourage local search at the end of the process. Thus, HPSO-TVAC in EPSO prevents premature convergence. In addition, FDR-PSO is used to enhance local search capability. Searching multiple local optima in multimodal optimization is achieved by using LIPS in EPSO. Furthermore, CLPSO-*gbest* is deployed to enhance the performance on solving multimodal optimization problems.

In Figures 4-10 and 4-11, the search behaviour profiles of EPSO algorithm on CEC2005 unimodal, multimodal, expanded and hybrid composition functions are shown alongside with their 3-D maps for 2D functions. The figures indicate that the behaviour of EPSO is varying from one problem to another. In simple unimodal sphere function, as shown in Figure 4-10 (a), EPSO promotes HPSO-TVAC algorithm in the early stage of the optimization process. Towards the global optimum, LIPSO, FDR-PSO, and CLPSO-*gbest* variants are assigned alternately to find the better solutions. After the population is converged, all algorithms became equal probability.

In the multimodal Shifted Rosenbrock's function shown in Figure 4-10 (b), EPSO assigned HPSO-TVAC algorithm at the beginning of optimization process and switched to PSO at the end of the process to find the near-optimum solution/the optimum. On the other hand, in the multimodal Shifted Rotated Ackley's function with the multiple local optima, LIPS, FDR-PSO, and CLPSO-*gbest* algorithms took the lead roles in finding the optimal solution as shown in Figure 4-10 (c). Similarly, in the Shifted Rotated Rastrigin's problem with multiple local optima, FDR-PSO and LIPS have higher probability values in the early stage as shown in Figure 4-11 (a). After getting the good solution, HPSO-TVAC fine-tuned to find better solutions. Finally, PSO drove the particle to converge toward the near-optimum/optimum solution.

The search behaviour profiles of EPSO on the Shifted Expanded Griewank's plus Rosenbrock's problem and hybrid composition problem with multiple local optima are illustrated in Figures 4-11 (b) and (c). The figures showed that the search behaviour profiles are similar to unimodal as well as multimodal optimization.

The search behaviour profiles of EPSO on CEC2013 benchmark problems are presented in Figure 4-12 and Figure 4-13. The search behaviour plots for the new unimodal different powers function and for two multimodal problems are shown in Figures 4-12 (a), (b) and (c) respectively. The search behaviour plots for the new multimodal, expanded and composition functions are illustrated in Figures 4-13 (a), (b) and (c), respectively. The figures show the search behaviour profiles are similar to that of CEC2005 benchmark problems.

It can be observed from the search behaviour profiles that EPSO employs various PSO strategies throughout the optimization process. During the optimization process, EPSO promotes its PSO strategies based on their merits to take the leading role in finding better solutions. Different PSO strategies are promoted for different types of problems. Therefore, the proposed EPSO algorithm is able to perform well consistently on different types of benchmark problems from both CEC2005 and recent CEC2013 competitions.

(a) F_1 : Shifted Sphere Function

(b) F_6 : Shifted Rosenbrock's Function (c) F_8 : Shifted Rotated Ackley's Function with Global Optimum on Bounds

Figure 4-10 Search behaviour plots for CEC2005 benchmark functions: (a) unimodal function F_1 and two multimodal functions: (b) F_6 and (c) F_8

(a) F_{10} : Shifted Rotated Rastrigin's Function

(b) F_{13} : Shifted Expanded Griewank's plus Rosenbrock's Function ($F_8 F_2$)

(c) F_{15} : Hybrid Composition Function

Figure 4-11 Search behaviour plots for CEC2005 benchmark functions: (a) multimodal function F_{10} , (b) shifted expanded function F_{13} and (c) hybrid composition function F_{15}

(a) F_5 : Different Powers Function

(b) F_8 : Rotated Ackley's Function (c) F_{11} : Rotated Rastrigin's Function

Figure 4-12 Search behaviour plots for CEC2013 benchmark functions: (a) unimodal function (F_5) and multimodal functions: (b) F_8 and (c) F_{11}

(a) F_{17} : Lunacek bi-Rastrigin Function

(b) F_{19} : Rotated Expanded Griewank's plus Rosenbrock's Function

(c) F_{28} : Composition Function

Figure 4-13 Search behaviour plots for CEC2013 benchmark functions: (a) multimodal function F_{17} , (b) rotated expanded function F_{19} and (c) composition function F_{28}

4.3.5 Discussion

The performance of ensemble PSO is investigated using different classes of problems such as unimodal, multimodal, expanded and hybrid composition problems and compared with its individual PSO strategy, recent OLPSO algorithm and heterogeneous PSO algorithm. The experimental results show the ensemble PSO performs consistently better than its individual PSO strategy and other recent PSO algorithms. In addition, the EPSO also performs well consistently (either best or second best) regardless of problem nature and size. It can be observed in the convergence characteristic graphs of test functions shown in Appendix 3. This verified that ensemble PSO benefits

from each individual PSO algorithm and enhances the universality and robustness performance.

4.4 Conclusion

In this Chapter, ensemble PSO (EPSO) is proposed in which a pool of PSO algorithms is constructed with *inertia weight* PSO, LIPS, HPSO-TVAC, FDR-PSO and CLPSO with *gbest*. Then, the suitable PSO strategy is gradually selected in self-adaptive manner to guide the particle in the current stage of the search process. In addition, CLPSO is assigned to guide the particles in small subpopulation to preserve the population diversity. In such a way, the diversity is maintained throughout the search process and the EPSO encourages the particles to converge to the global optimum without being trapped into a local optimum. The performance of EPSO algorithm is evaluated using CEC2005 and CEC2013 benchmark functions. The experimental results shows that the proposed EPSO algorithms performs well consistently on CEC2005 benchmark function as well as recent CEC2013 benchmark set and outperforms its individual PSO variants as well as competitive PSO algorithms.

5 HYBRID SOLUTION TO UC PROBLEM

5.1 Introduction

As mentioned in Section 2.2, the unit commitment (UC) problem involves two sub-problems that can be solved separately. Two or more deterministic or metaheuristic methods can be used to handle two sub-problems separately. On the other hand, deterministic method can be used to solve the unit-scheduling sub-problem while metaheuristic method is applied to solve the economic dispatch sub-problem. In UC, unit-scheduling is the combinatorial optimization problem with integer variables where deterministic methods based on numerical optimization technique are suitable. ED sub-problem is a non-linear, non-convex optimization problem with intense equality and inequality constraints. Therefore, finding the global optimum of ED sub-problem using conventional deterministic methods is difficult. However, metaheuristic methods that are intelligent general-purposed optimization methods are suitable for solving ED sub-problem.

Therefore, employing a deterministic method to solve the unit-scheduling sub-problem and an metaheuristic method to solve the ED sub-problem is a promising potential solution to the UC to overcome high dimensionality issue and to improve the solution quality. Deterministic methods are fast in finding the solution yet may miss the optimal solution. Metaheuristic methods are flexible yet computationally expensive.

However, by combining these two methods, they can benefit from each other and enable us to solve the large-scale power systems. Motivated by this, in this Chapter, a new hybrid solution of deterministic method to solve unit-scheduling sub-problem and metaheuristic method to handle ED sub-problem is proposed. In the proposed hybrid method, the time-ahead priority listing (TPL) method is introduced as a deterministic method to solve unit-scheduling sub-problem whereas heterogeneous comprehensive learning PSO (HCLPSO) presented in Chapter 3.3 is proposed as a metaheuristic method to solve the ED sub-problem.

5.2 Hybrid Solution to UC

5.2.1 UC Problem Formulation and Problem Statement

The economic operation of a power system is mainly effected by how UC problem is formulated. A power system is comprised of five major components: generation, transmission, distribution, system load demand and network. In this Thesis, unit commitment scheduling problem from generation section is studied. The following assumptions are made before UC problem is formulated [212]:

- The network in the power system is capable of transmitting generated power to demand side without losses or network failure.
- The load demand is fixed for scheduling period.
- The operating cost of a generating unit is assumed to be composed of start-up cost and production cost.
- Shut-down cost of a generating unit is considered as zero.

Then, as mentioned in Chapter 2.2.1, UC problem is formulated as a minimization problem to obtain minimum operation cost while satisfying the system and unit constraints.

5.2.2 Unit-scheduling

Unit-scheduling sub-problem determines scheduling of generating units over a scheduling period. The committed units must meet system demand and spinning reserve requirements, subject to a variety of constraints. The sub-problem is considered as a combinatorial optimization problem with the binary 0/1 (on/off) variables. In order to handle this sub-problem, a new priority listing method that is based on simple numerical optimization technique is developed. Before a new priority listing method is presented, simple priority listing method will be briefly introduced and explained in this section.

5.2.2.1 Priority Listing (PL) Method

Priority listing (PL) method is the simple solution to UC problem [169]. Priority list is created based on the parameters of each unit. Units with higher capacity will have a higher priority. Thus, a unit with the highest maximum power capacity will be located on the top of the priority list and the rest units will be located in the ascending order of their maximum output power toward the bottom of the list. If the units have the same maximum power capacity, the units will start up in the descending order of heat rate. Thus, the unit with lower heat rate (HR) will have higher priority. The heat rate HR is calculated according to the formula:

$$HR = \frac{F_i(P_{i,t})}{P_{i,t}} \quad (5.1)$$

The units are committed based on the priority list until the load demand and the spinning reserve constraints are satisfied at each time interval.

5.2.2.2 Time-ahead Priority Listing Method (TPL)

The proposed PL method is composed of two steps. The first step is turning on the generators to meet the load demand and spinning reserve requirements. The next step is identifying unsuitable generators and turning them off to produce the optimal schedule with a minimum production cost. In the first step, for each scheduling hour t , current maximum available power generation of the power system (P_t^{max}) and available power capacity (P_t^{ava}) are calculated as follows:

$$P_t^{max} = \sum_{i=1}^N P_i^{max} U_{i,t} \quad (5.2)$$

$$P_t^{ava} = P_t^{max} - (P_{D,t} + P_{R,t}) \quad (5.3)$$

If load demand and spinning reserve requirements are not satisfied, i.e. P_t^{ava} is less than zero, we look ahead the number of consecutive hours ($T_t^{P_t^{max} < (P_{D,t} + P_{R,t})}$) at which P_t^{max} does not meet $(P_{D,t} + P_{R,t})$. Then, the generators with their minimum up time (T_i^{up}) close to the number of consecutive hours ($T_t^{P_t^{max} < (P_{D,t} + P_{R,t})}$) will be turned on until satisfying load demand and spinning reserve requirements at hour t . By considering ahead the time $T_t^{P_t^{max} < (P_{D,t} + P_{R,t})}$ and turning on the generators with their minimum up time close to the number of consecutive hours ($T_t^{P_t^{max} < (P_{D,t} + P_{R,t})}$), we can prevent unnecessary starting up and shutting down generation units during the scheduling period.

In the second step, we find the generators, which satisfy two criteria: (1) the unit must be on and (2) $P_t^{ava} > P_i^{max}$. In reverse to the first step, the number of consecutive hour ($T_t^{P_t^{max} \geq (P_{D,t} + P_{R,t})}$) at which current $P_t^{max} \geq P_{D,t} + P_{R,t}$ is considered ahead this time. Then, among the generators which satisfied two criteria, the generator with the lowest minimum downtime (T_i^{down}) are turned off if $T_t^{P_t^{max} \geq (P_{D,t} + P_{R,t})}$ is greater than its minimum down time. If there were the generators with the same minimum down time, the one with the highest P_i^{max} will be selected to be turned off.

The minimum up (T_i^{up}) and down time (T_i^{down}) constraints are considered before turning on and turning off the generators. In the calculation of the start-up cost ($ST_{i,t}$), the start-up cost is the cold start-up cost SC_i^{cold} when the generator is turned off for more than the summation of minimum down time (T_i^{down}) and cold start hours (T_i^{cold}). Otherwise, the start-up cost is the hot start-up cost (SC_i^{hot}). The procedure of the proposed time-ahead PL method is described systematically as follows:

Step 1. FOR $t = 1: T$

Step 2. Calculate P_t^{max} and P_t^{ava}

Step 3. WHILE ($P_t^{max} < (P_{D,t} + P_{R,t})$ at t)

- 1) Calculate the number of consecutive hours $T_t^{P_t^{max} < (P_{D,t} + P_{R,t})}$
- 2) Find $A = (\text{Generators with Minimum Up time } (T_i^{up}) \text{ near to } (T_t^{P_t^{max} < (P_{D,t} + P_{R,t})}))$
- 3) Turn on the generators in A if satisfies $T_i^{off} > T_i^{down}$

END WHILE

Step 4. Find $B = (\text{Generators are ON \& } (P_t^{max} < P_t^{ava}))$

Step 5. WHILE (NOT EMPTY B)

- 1) $l = \text{Minimum (Minimum Down time } (T_i^{down}) \text{ of Generators in } B)$
- 2) $C = (\text{Generators in } B \text{ with Minimum Down time} = l)$
- 3) $D = (\text{Generators in } C \text{ with highest maximum power } (P_i^{max}))$
- 4) Find the number of consecutive hours $(T_t^{P_t^{max} \geq (P_{D,t} + P_{R,t})})$
- 5) IF $T_t^{P_t^{max} \geq (P_{D,t} + P_{R,t})} > 1$

Turn off the generators in D if satisfies $T_i^{on} > T_i^{up}$

END IF

END WHILE

END FOR

5.2.3 Economic Dispatch

The ED sub-problem is an essential nonlinear optimization problem in UC with continuous variables. Once hour-to-hour scheduling is generated, ED allocates the load demand among the committed units with a minimum production cost while satisfying the physical and operational constraints of each unit. Therefore, using an efficient and fast optimization method for solving the ED sub-problem can improve the quality of the UC solution and thereby the performance of the overall UC problem. In this section, heterogeneous comprehensive learning PSO (HCLPSO) method proposed in Chapter 3.3 is used to tackle the ED sub-problem. The implementation of HCLPSO for ED sub-problem is as follows:

- Step 1. The particles in the population are randomly initialized within the generating limits of each committed unit. For D number of committed units, the i^{th} particle is represented as $P_i = [P_i^1, P_i^2, \dots, P_i^D]$. For the d^{th} dimension,

$$P_i^d = P_{min}^d + rand * (P_{max}^d - P_{min}^d), rand \in (0,1) \quad (5.3)$$

Step 2. For each particle in the swarm, the fitness function called evaluation function is identified as follows:

$$\text{Objective: Minimize } f(P_i) = \sum_{i=1}^D F_i(P_i) \quad (5.4)$$

$$\text{Subject to } \sum_{i=1}^D P_i = P_D \quad (5.5)$$

$$\sum_{i=1}^D P_{i,max} = P_{D,t} + P_{R,t} \quad (5.6)$$

$$P_{ij,min} \leq P_{ij} \leq P_{ij,max} \quad (5.7)$$

Step 3. Calculate the fitness value of each particle and initialize $pbests$, $pbest_{fi(d)}^d$ and $gbest$.

Step 4. Update the velocity of each particle in the population using equation (3.1) for subpopulation 1 (exploration-enhanced population) and equation (3.3) for subpopulation 2 (exploitation-enhanced population).

Step 5. If $(V_i^d > V_{max}^d)$, then $V_i^d = V_{max}^d$ and if $(V_i^d < V_{min}^d)$, then $V_i^d = V_{min}^d$.

Step 6. Update position of each individual particle by equation (2.2).

Step 7. Satisfying the constraint (5.7) if $(P_i^d > P_{max}^d)$, then $P_i^d = P_{max}^d$ and if $(P_i^d < P_{min}^d)$, then $P_i^d = P_{min}^d$.

Step 8. Evaluate the fitness of each particle using equation (5.4) and infeasible solutions that violate the constraints (5.5) and (5.6) are handled by epsilon constraint handling method [213].

Step 9. Update $pbests$, $pbest_{fi(d)}^d$ and $gbest$.

Step 10. Go to step 4 until the stopping criterion is satisfied.

5.3 Experimental Setup and Results

The unit characteristics and the load demand are given in Tables 5-1 [214] and 5-2 [186]. In all simulations, the spinning reserve requirement is assumed to be 10% of the load demand and total scheduling period is 24 hours. The simulations are run 50 trials for 10-, 20-, 40-, 60-, 80-, and 100-bus systems. The performance of the proposed hybrid model is evaluated with other hybrid methods and compared in terms of average production cost and execution time averaged over 50 runs. Expansion of the number of units is based on the 10-bus system. For the 20-bus system, the base 10 units are duplicated and the load demand is multiplied by 2. The other unit systems are expanded in the same manner. The settings for other PSOs and HCLPSO are as follows:

Dimension, D = number of committed units

Population size = $5 * D$

Maximum number of function evaluation = $5000 * D$

Table 5-1 Data Specifications of 10-bus Test System

	Unit 1	Unit 2	Unit 3	Unit 4	Unit 5	Unit 6	Unit 7	Unit 8	Unit 9	Unit 10
$P_{\max}$	455	455	130	130	162	80	85	55	55	55
$P_{\min}$	150	150	20	20	25	20	25	10	10	10
a	1000	970	700	680	450	370	480	660	665	670
b	16.19	17.26	16.60	16.50	19.70	22.26	27.74	25.92	27.27	27.79
c	0.00048	0.00031	0.002	0.00211	0.00398	0.00712	0.00079	0.00413	0.00222	0.00173
T_{up}	8	8	5	5	6	3	3	1	1	1
T_{down}	8	8	5	5	6	3	3	1	1	1
ST_h	4500	5000	550	560	900	170	260	30	30	30
ST_c	9000	10000	1100	1120	1800	340	520	60	60	60
CSH	5	5	4	4	4	2	2	0	0	0
Initial Status	8	8	-5	-5	-6	-3	-3	-1	-1	-1

Table 5-2 Hourly Load Demand

hr	Demand	hr	Demand	hr	Demand	hr	Demand	hr	Demand	hr	Demand
1	700	5	1000	9	1300	13	1400	17	1000	21	1300
2	750	6	1100	10	1400	14	1300	18	1100	22	1100
3	850	7	1150	11	1450	15	1200	19	1200	23	900
4	950	8	1200	12	1500	16	1050	20	1400	24	800

5.3.1 Comparison of proposed time-ahead PL method and Other PL methods

In this Chapter, the proposed TPL method is compared with the simple priority listing (PL) [169], extended priority listing (EPL) [168], stochastic priority listing (SPL) [215] and methodological priority listing (MPL) [216]. EPL and SPL simulation results are obtained from [168] and [215]. For a fair comparison, MPL, PL and the proposed time-ahead PL methods are performed using the same power system as in [168] and [215].

The performance of each PL method is evaluated using their average production cost and execution time obtained in 50 trial runs. The comparisons of the experimental results are presented in Tables 5-3 and 5-4, respectively. As shown in Table 5-3, the proposed PL method provides better solutions than other compared PL methods and consistently performs well on both small and large power systems. In terms of execution time (sec), EPL method obtains the solutions faster than proposed PL method. However, proposed PL method finds the best solutions rapidly on large-scale power systems for the problem sizes of 80 and 100.

Table 5-3 Comparison of Time-ahead PL method vs other PL methods

unit systems	EPL		SPL		MPL		PL		TPL	
	\$	sec	\$	sec	\$	sec	\$	sec	\$	sec
10	563977	0.72	564950	7.24	601523	7.41	566561	3.98	563373	4.12
20	1124369	2.97	1123938	16.32	1246022	16.11	1222428	11.34	1121331	7.55
40	2246508	11.9	2248645	46.32	2521399	36.50	2492200	25.94	2236850	14.42
60	3366210	23	3371178	113.85	3773488	61.05	3753608	42.66	3353575	23.91
80	4489322	44.4	4492909	215.77	5068776	97.30	5011279	68.18	4474988	34.30
100	5608440	64.5	5615530	374.03	6041572	106.20	5707747	57.11	5596410	49.37

Table 5-4 Production Cost Difference of Time-ahead PL vs other PL methods

unit systems	Production Cost Difference (\$)			
	EPL vs. TPL	SPL vs. TPL	MPL vs. TPL	PL vs. TPL
10	6.04E+02	1.58E+03	3.81E+04	3.19E+03
20	3.04E+03	2.61E+03	1.25E+05	1.01E+05
40	9.66E+03	1.18E+04	2.85E+05	2.55E+05
60	1.26E+04	1.76E+04	4.20E+05	4.00E+05
80	1.43E+04	1.79E+04	5.94E+05	5.36E+05
100	1.20E+04	1.91E+04	4.45E+05	1.11E+05

5.3.2 Comparison of PSOs vs. HCLPSO on ED sub-problem

Before applying the proposed hybrid solution to UC, the performance of HCLPSO method on ED sub-problem is evaluated with other PSO algorithms of PSO [13], CLPSO [78], FIPS [31] and HPSO-TVAC [57]. Simple PL method is used to handle the UC sub-problem when comparing the performances of HCLPSO and other PSOs on ED sub-problem. The hybrid models are compared in terms of average production cost and execution time in Tables 5-5 and 5-6, respectively. In terms of the execution time, PL+ PSO hybrid model finds the solutions faster than PL-HCLPSO.

However, PL-HCLPSO hybrid model achieves better solutions and outperforms all other hybrid models in terms of minimum production cost. This proves that unlike other PSOs, HCLPSO is able to jump out of the local minimum and achieve a better performance.

Table 5-5 Comparison of (PL-HCLPSO) hybrid model vs other hybrid models

Unit Systems	PL-PSO		PL-CLPSO		PL-HPSO-TVAC		PL-HCLPSO	
	\$	sec	\$	sec	\$	sec	\$	sec
10	560373.67	4.19	558320.68	7.37	557964.97	18.59	558691.24	9.79
20	1155830.28	11.00	1154510.28	18.23	1154391.81	98.45	1131147.05	21.45
40	2232133.00	25.46	2230157.77	42.39	2232724.17	396.26	2209829.61	46.33
60	3279889.57	41.27	3280796.54	68.93	3281721.41	937.17	3253645.04	74.84
80	4318242.65	66.35	4318787.19	103.01	4315948.04	1718.47	4293368.37	113.13
100	4870580.98	59.95	4873975.25	88.78	4873959.08	1980.00	4868845.64	102.95

Table 5-6 Production Cost Difference of PL+HCLPSO vs other hybrid models

Unit Systems	Production Cost Difference (\$)		
	PL-PSO	PL-CLPSO	PL-HPSO-TVAC
10	1.68E+03	-3.71E+02	-7.26E+02
20	2.47E+04	2.34E+04	2.32E+04
40	2.23E+04	2.03E+04	2.29E+04
60	2.62E+04	2.72E+04	2.81E+04
80	2.49E+04	2.54E+04	2.26E+04
100	1.74E+03	5.13E+03	5.11E+03

5.3.3 Comparison of TPL-HCLPSO vs. hybrid PSO models

The proposed hybrid model (TPL-HCLPSO) is implemented and evaluated with other three hybrid models: new PL-PSO, PL-PSO and PL-HCLPSO. Firstly, the proposed TPL method is merged together with PSO and applied to solve the UC. Com-

pared to PL-PSO and PL-HCLPSO hybrid models, TPL-PSO model offers faster execution time as well as better quality solutions.

In terms of execution time presented in Table 5-7, TPL-PSO can find the solutions rapidly even if the problem size is large. However, the execution time doubles after hybridizing the proposed TPL with HCLPSO algorithm. Thus, it can be observed that the newly proposed PL method executes faster, but longer execution time is due to the complexity of the HCLPSO algorithm.

In terms of production cost, TPL-PSO hybrid model fails to offer low minimum production cost. Among four hybrid models, the combination of the proposed TPL method and HCLPSO obtains the lowest minimum production cost. As shown in Table 5-8, the production cost of the proposed hybrid model is lower than other hybrid models as much as \$ 300 - \$ 300000 per day. Hence, the proposed new hybrid model is able to find better solutions and improves the overall performance on UC despite the doubling of execution time.

Table 5-7 Comparison of new hybrid solution vs other hybrid solutions

unit systems	TPL-PSO		TPL-HCLPSO		PL-PSO		PL-HCLPSO	
	\$	sec	\$	sec	\$	sec	\$	sec
10	556541.60	4.88	554384.40	9.87	560373.67	4.19	558691.24	9.79
20	1056752.32	7.48	1056021.38	18.22	1155830.28	11.00	1131147.05	21.45
40	2031855.14	16.77	2030983.67	33.97	2232133.00	25.46	2209829.61	46.33
60	2983200.68	26.25	2981784.45	54.78	3279889.57	41.27	3253645.04	74.84
80	3932344.75	36.09	3930617.68	82.51	4318242.65	66.35	4293368.37	113.13
100	4858214.23	52.21	4857926.05	100.92	4870580.98	59.95	4868845.64	102.95

Table 5-8 Production Cost Difference between new hybrid solution and other hybrid solutions

unit systems	Production Cost Difference (\$)		
	TPL-PSO vs. TPL-HCLPSO	PL-PSO vs. TPL-HCLPSO	PL-HCLPSO vs TPL -HCLPSO
10	2.16E+03	5.99E+03	4.31E+03
20	7.31E+02	9.98E+04	7.51E+04
40	8.71E+02	2.01E+05	1.79E+05
60	1.42E+03	2.98E+05	2.72E+05
80	1.73E+03	3.88E+05	3.63E+05
100	2.88E+02	1.27E+04	1.09E+04

5.3.4 Comparison of proposed (TPL-HCLPSO) and other optimization methods

In this section, the proposed hybrid solution combining time-ahead priority listing method and HCLPSO is compared with other deterministic (ELR [217] and RM [218]) and metaheuristic (GA[214] and ES-PSO [102]) methods in terms of minimum production cost and execution time. In terms of minimum production cost, the proposed hybrid approach of TPL-HCLPSO obtains the lowest minimum production cost for all small and large-scale power systems. In terms of execution time, the relaxation method RM performs the best, however, the algorithm fails to offer minimum production cost.

Table 5-9 Comparison of TPL-HCLPSO vs other methods in terms of production cost

unit sys- tems	Total Production Cost (\$)				
	TPL-HCLPSO	GA	RM	ES-PSO	ELR
10	554,384	565,825	563,977	563,942	563,977
20	1,056,021	1,126,243	1,123,990	1,123,474	1,123,297
40	2,030,984	2,251,911	2,243,676	2,243,517	2,244,237
60	2,981,784	3,376,625	3,361,589	3,364,719	3,363,491
80	3,930,618	4,504,933	4,481,833	4,483,992	4,485,633
100	4,857,926	5,627,437	5,599,761	5,603,815	5,605,678

Table 5-10 Comparison of TPL-HCLPSO vs other methods in terms of execution time

unit sys- tems	Execution Time (sec)				
	TPL-HCLPSO	GA	RM	ES-PSO	ELR
10	9.87	221	1.38	12	4
20	18.22	733	2.48	19	16
40	33.97	2697	5.37	42	52
60	54.78	5840	9.61	56	113
80	82.51	10,036	14.19	113	209
100	100.92	15,733	18.9	162	345

5.4 Conclusion

In this Chapter, a new hybrid solution is proposed to solve the UC. Firstly a new priority listing method is developed to solve the unit-scheduling sub-problem and compared with other PL methods. Then the deterministic new PL method is combined with the metaheuristic heterogeneous comprehensive learning PSO (HCLPSO) method to form the hybrid model to tackle the UC problem. The proposed hybrid solution is tested using the 10-, 20-, 40-, 60-, 80-, and 100-bus test systems and compared with

other hybrid models of PL-PSO, TPL-PSO and PL-HCLPSO. The proposed hybrid model outperforms other hybrid models on all small and large power systems and provides the lowest minimum production cost.

6 BINARY HCLPSO To UC WITH RENEWABLE RESOURCES

6.1 Introduction

All over the world, the sizes of power plants are increasing to meet the enormous global power demand. However, with the concern over the global climate change, renewable energy resources have been integrated to the power system as a means of reducing greenhouse gas emission.

When discussing integration of renewable resources, wind and solar renewable energy are the focus more than other resources as both are usually integrated in relatively large quantities on the both supply and demand sides of the power system. However, their variable and uncertain nature can induce significant fluctuation on the supply side. Using advanced metering infrastructure on demand side can cause significant uncertainty and unpredictable interruption to the power system. Due to such uncertainty and unpredictable nature, unit commitment (UC) problem in power generation planning is challenging in a power system to balance energy generation and demand for power system security and reliability requirement. Therefore, in this Chapter, UC problem is formulated with renewable resources (solar and wind) and studied considering minimizing the power production cost as well as reducing CO₂ emission.

Due to uncertainties in load demand and renewable resources, UC problem with renewable resources are difficult to handle by traditional optimization methods.

On the other hand, CI techniques that can adapt and facilitate the intelligent behaviour in complex and uncertain environments are suitable for solving such kinds of the problems under uncertainty. Therefore, in this Chapter, one of the variants of CI technique, binary HCLPSO algorithm, is proposed to find the optimal solution to UC problem with renewable resources. Then, UC problem under uncertainty is investigated with/without renewable resources in order to reduce greenhouse gas emission.

6.2 Binary HCLPSO to UC problem with Renewable Resources

6.2.1 UC Problem Formulation

Due to unpredictable and intermittent features of solar and wind energies, the actual output power from solar and wind farms differs from the forecasted output power. The uncertainty in solar and wind energies can be represented with probability distribution function of forecasted error. Since it is impossible to cover for all the scenarios in the continuous state, a discrete probability set [136] is used to represent uncertainty in load demand, solar and wind energy resources. The UC problem is reformulated as a cost-emission optimization problem to reduce the production cost as well as emission cost as follows:

Objective Function: UC Cost-Emission Optimization

$$PC = \sum_{s \in S} \rho_{wind} \rho_{solar} \rho_D \sum_{i=1}^N \sum_{t=1}^T [W_c (F_i(P_{i,t}^s) + ST_{i,t}(1 - U_{(i-1),t})) + W_e(\psi_i \mathcal{E}C_i(P_{i,t}^s))] \quad (6.1)$$

where, S is a discrete set of scenarios and s represents each scenario. The parameter s_0 is the penalty factor and ‘ ρ ’ represents the corresponding probability of load demand, solar and wind resources. Start-up cost ($ST_{i,t}$) are calculated as described in Section 2.2.1. Fuel and Emission costs are calculated according to:

$$\text{Fuel Cost: } F_i(P_{i,t}^s) = a_i + b_i P_{i,t}^s + c_i (P_{i,t}^s)^2 \quad (6.2)$$

$$\text{Emission Cost: } \mathcal{E}C_i(P_{i,t}^s) = \alpha_i + \beta_i P_{i,t}^s + \gamma_i (P_{i,t}^s)^2 \quad (6.3)$$

where, $P_{i,t}^s$ is the output of unit i at time t considering the scenario s . The parameter a , b , c , α , β and γ are fuel cost and emission coefficients. W_c and W_e are weight factors and either $W = 0$ or 1 value is used to increase the system flexibility. ψ_i is the emission penalty factor calculated by:

$$\psi_i = \frac{F_i(P_i^{max})}{\mathcal{E}C_i(P_i^{max})} \quad (6.4)$$

Constraints: The constraints of UC cost-emission optimization problem are:

System Constraints:

- 1) *Power balance:* The power generated from thermal generator, solar and wind units must satisfy the system demand:

$$\sum_{i=1}^N P_{i,t}^s + P_{wind,t}^s + P_{solar,t}^s = P_{D,t}^s \quad (6.5)$$

- 2) *Spinning Reserve Constraint:* the spinning reserve is required to maintain the security and reliability of the power system:

$$\sum_{i=1}^N P_{i,t}^{max} + P_{wind,t}^s + P_{solar,t}^s \geq P_{D,t}^s + P_{R,t} \quad (6.6)$$

Unit Constraints:

Unit constraints of UC cost-emission optimization problem are the same as the constraints of UC problem without renewable resources presented in Section 2.2.1.

6.2.2 Binary HCLPSO Approach to UC Problem

In UC problem, the number of on/off binary combinations grows exponentially with increasing number of generating units. Hence, the problem becomes more and more complex for large scale power industry. In addition, the uncertainty nature of renewable energy and power demand added fuel to the fire. Finally, the conventional optimization techniques have been pushed to their limits and a new technique is urgently required to address the emerging problem. Compared to the conventional optimization methods, intelligent optimization techniques are more suitable to handle the uncertainty

due to their capability of facilitating intelligent behaviour in complex and changing environments. Therefore, a binary version of HCLPSO algorithm is proposed in this Chapter, based on binary PSO (BPSO) presented by Kennedy and Eberhart in [98].

In binary HCLPSO (B-HCLPSO), a particle's velocity are updated using the same velocity update equations (3.1) and (3.3) of HCLPSO algorithm proposed in Chapter 3.3. The position update equation (2.2) is replaced by the following equation:

$$\text{if } rand() < s(V_i^d), \text{ then } X_i^d = 1 \quad (6.7)$$

$$\text{else } X_i^d = 0$$

where, $rand()$ is a quasirandom number selected from a uniform distribution in the range of $[0.0, 1.0]$. $s(V_i^d)$ is the sigmoid function which is defined as follows:

$$s(V_i^d) = \frac{1}{1 + \exp(-V_i^d)} \quad (6.8)$$

In binary HCLPSO, the position X and the previous best position $pbest$ can take only 0 or 1 integer values. According to equation (6.8), the velocity V_{max} determines the probability the position X_i^d will take on 0 or 1 values. With the high V_{max} value, 1 will be on most of dimension and with low V_{max} , the individual is more likely to choose 0 in most of dimensions. Thus, in binary particle swarm, V_{max} function is similar to the mutation rate in GA and small V_{max} allows high mutation rate. In binary HCLPSO, $V_{max} = \pm 4$ is used to have high mutation rate. The same c , c_1 , c_2 and w values of HCLPSO algorithm are used for the binary version of HCLPSO algorithm.

In binary HCLPSO approach for UC problem under uncertainty, each particle represents the potential solution to the UC problem. Firstly, the particles are randomly initialized in the search space. The position and velocity of each particle are defined as vectors $X = (X_1, X_2, \dots, X_D)$ and $V = (V_1, V_2, \dots, V_D)$. If there are N number of generators to generate the power to meet the load demand during scheduling period T , the dimension of each individual is $N*T$ and each individual represents the on/off status of generating units for the period T as shown in Figure 6-1. If the schedule is $[U_{1,t}, U_{2,t}, U_{3,t}, \dots, U_{N,t}, P_{solar,t}, P_{wind,t}]$ at time t , for all scenarios ($s \in S$), the load demand $(D_t^s - P_{solar}^t - P_{wind}^t)$ is dispatched using lambda iteration method. At each

scenario s , the load demand (D_t^s), the solar ($P_{solar,t}^s$) and wind ($P_{wind,t}^s$) power is calculated with known discrete probability from Table 6-1 according to [136]. The flow chart of UC cost-emission optimization by binary HCLPSO is presented in Figure 6-2.

Figure 6-1 The population structure of binary HCLPSO algorithm for UC problem

Figure 6-2 Flow chart of UC cost-emission optimization using binary HCLPSO algorithm

6.2.3 Experimental Setup

The same 10-bus power system, as described in Chapter 5, is considered for simulation of UC problem with renewable resources. The characteristics of each unit are as described in Table 5-1 and the hourly load demand of the power system is the same as in Table 5-2. The discrete probability set [136] are presented in Table 6-1 and the emission coefficients [219] are listed in Table 6-2. The wind speed and solar data used to estimate forecasting wind and solar power are presented in Figure 6-2 and Figure 6-3 [136]. The parameter settings of the binary HCLPSO algorithm are as follows:

Population Size = 20, Number of generations = 1000

Dimension = number of generation units * scheduling period ($T=24$ hours)

In order to validate the performance of binary HCLPSO algorithm, binary HCLPSO algorithm is tested on simple UC problem without considering uncertainty and compared with the hybrid approach of binary and real coded PSO [104], improved binary PSO algorithm [105], integer-coded GA [220] and RM [218] algorithms. The parameter settings of the binary HCLPSO are defined as the same as in other hybrid approaches. The comparison of the results of the proposed binary HCLPSO and other optimizations methods are shown in Table 6-3. The solutions to UC problem are compared with respect to total power production cost without considering emission and renewable resources.

Table 6-1 A discrete distribution set of solar, wind and demand [136]

Wind			Solar			Demand		
Probability	Expected (x %)	Expected (x %)	Probability	Expected (x %)	Expected (x %)	Probability	Expected (x %)	Expected (x %)
ρ_{wind}	(1~11) h	(12~24) h	ρ_{solar}	(1~11) h	(12~24) h	ρ_D	(1~11) h	(12~24) h
0.50	100%	100%	0.70	100%	100%	0.60	100%	100%
0.15	99%	98%	0.15	98.5%	97.5%	0.15	98.5%	98%
0.15	101%	102%	0.15	101.5%	102.5%	0.15	102%	103%
0.10	97.5%	95%				0.05	98%	97%
0.10	102.5%	105%				0.05	103%	104%
*Expected value x % means x % of the forecasted data of solar, wind and load demand								

Table 6-2 Emission Coefficients for 10-bus power system [219]

Unit	α (ton h ⁻¹)	β (ton MW ⁻¹ h ⁻¹)	γ (ton MW ⁻² h ⁻¹)
1	103.3908	-2.4444	0.0312
2	103.3908	-2.4444	0.0312
3	300.391	-4.0695	0.0509
4	300.391	-4.0695	0.0509
5	320.0006	-3.8132	0.0344
6	320.0006	-3.8132	0.0344
7	330.0056	-3.9023	0.0465
8	330.0056	-3.9023	0.0465
9	350.0056	-3.9524	0.0465
10	360.0012	-3.9864	0.047

Figure 6-3 Irradiance Solar Data obtained from NREL solar prospector [136]

Figure 6-4 Wind Speed Data obtained from NREL wind prospector [136]

Table 6-3 Comparison of Binary HCLPSO vs. other solutions to UC problem

Method	Total Production Cost (\$)		
	Best	Average	Worst
ICGA [220]	566,404	-	-
HPSO [104]	563,942	564,772	565,785
IBPSO [105]	563,977	564,155	565,312
RM [218]	563,977	-	-
BHCLPSO	563,189	564,110	566,075

Compared to improved binary PSO, hybrid PSO and integer coded GA, binary HCLPSO performs the best in solving UC problem of 10-bus power system. Then, UC problem under uncertainty is investigated for two cases: under uncertainty in load demand without renewable resources and under uncertainty in load demand, solar and wind data. Considering the uncertainty, the results are shown in Table 6-4 for the UC problem under uncertainty in load demand and in Table 6-5 for UC problem under uncertainty in demand as well as renewable resources. It can be observed that the total production cost and emission cost are lower once solar and wind energies are integrated. In addition, the complexity of the problem increases by 15 times once the renewable energies are integrated. This case study shows that the proposed binary HCLPSO could find the feasible solution although complexity increases multiple times.

Table 6-4 Scheduling and Power Dispatch of UC problem without Solar and Wind

	1	2	3	4	5	6	7	8	9	10	Solar	Wind	Demand
Hr 1	455	245	0	0	0	0	0	0	0	0	0	0	700
Hr 2	455	295	0	0	0	0	0	0	0	0	0	0	750
Hr 3	455	265	0	130	0	0	0	0	0	0	0	0	850
Hr 4	455	355	0	130	0	0	0	0	10	0	0	0	950
Hr 5	455	395	0	130	0	0	0	10	10	0	0	0	1000
Hr 6	455	455	0	130	0	0	25	25	10	0	0	0	1100
Hr 7	455	455	0	130	0	65	25	10	10	0	0	0	1150
Hr 8	455	440	130	130	0	20	25	0	0	0	0	0	1200
Hr 9	455	455	130	130	85	20	25	0	0	0	0	0	1300
Hr 10	455	455	130	130	162	33	25	10	0	0	0	0	1400
Hr 11	455	455	130	130	162	73	25	10	0	10	0	0	1450
Hr 12	455	455	130	130	162	80	25	43	10	10	0	0	1500
Hr 13	455	455	130	130	162	33	25	10	0	0	0	0	1400
Hr 14	455	455	130	130	95	0	25	10	0	0	0	0	1300
Hr 15	455	435	130	130	25	0	25	0	0	0	0	0	1200
Hr 16	455	310	130	130	25	0	0	0	0	0	0	0	1050
Hr 17	455	260	130	130	25	0	0	0	0	0	0	0	1000

Hr 18	455	340	130	130	25	20	0	0	0	0	0	0	1100
Hr 19	455	405	130	130	25	20	25	0	10	0	0	0	1200
Hr 20	455	455	130	130	155	20	25	10	10	10	0	0	1400
Hr 21	455	455	130	130	85	20	25	0	0	0	0	0	1300
Hr 22	455	360	130	130	25	0	0	0	0	0	0	0	1100
Hr 23	455	420	0	0	25	0	0	0	0	0	0	0	900
Hr 24	455	320	0	0	25	0	0	0	0	0	0	0	800
Power Production Cost = \$559448, Emission Cost= \$679480, Number of Scenarios = 5													

Table 6-5 Scheduling and Power Dispatch of UC problem with Solar and Wind

	1	2	3	4	5	6	7	8	9	10	Solar	Wind	Demand
Hr 1	455	245	0	0	0	0	0	0	0	0	0	10.54	700
Hr 2	455	295	0	0	0	0	0	0	0	0	0	22.27	750
Hr 3	455	370	0	0	25	0	0	0	0	0	0	25.5	850
Hr 4	455	455	0	0	40	0	0	0	0	0	0	25.5	950
Hr 5	455	390	0	130	25	0	0	0	0	0	0	25.5	1000
Hr 6	455	360	130	130	25	0	0	0	0	0	0	25.5	1100
Hr 7	455	410	130	130	25	0	0	0	0	0	0.09	25.5	1150
Hr 8	455	455	130	130	30	0	0	0	0	0	17.46	25.5	1200
Hr 9	455	455	130	130	95	0	25	10	0	0	31.45	25.5	1300
Hr 10	455	455	130	130	162	33	25	10	0	0	36.01	25.5	1400
Hr 11	455	455	130	130	162	73	25	10	10	0	38.06	25.5	1450
Hr 12	455	455	130	130	162	80	25	43	10	10	35.93	25.5	1500
Hr 13	455	455	130	130	162	23	25	10	10	0	36.78	25.5	1400
Hr 14	455	455	130	130	95	0	25	10	0	0	31.59	24.82	1300
Hr 15	455	435	130	130	25	0	25	0	0	0	9.7	20.74	1200
Hr 16	455	310	130	130	25	0	0	0	0	0	12.92	14.62	1050
Hr 17	455	260	130	130	25	0	0	0	0	0	0	25.5	1000
Hr 18	455	360	130	130	25	0	0	0	0	0	0	19.04	1100
Hr 19	455	450	130	130	25	0	0	10	0	0	0	25.5	1200
Hr 20	455	455	130	130	162	23	25	10	10	0	0	18.02	1400
Hr 21	455	455	130	130	85	20	25	0	0	0	0	25.5	1300
Hr 22	455	445	130	0	25	20	25	0	0	0	0	21.42	1100
Hr 23	455	420	0	0	25	0	0	0	0	0	0	0	900
Hr 24	455	320	0	0	25	0	0	0	0	0	0	2.55	800
Power Production Cost=\$553926.29, Emission Cost=\$668655.61, Number of Scenarios = 75													

6.3 Conclusion

In this Chapter, UC problem is formulated as cost-emission optimization problem and binary HCLPSO is proposed to solve the UC problem under uncertainty considering the concern over reducing production cost as well as emission cost. Before applying binary HCLPSO algorithm to UC problem under uncertainty, the performance

of binary HCLPSO is evaluated by comparing with other optimization techniques without considering uncertainty and emission cost. The experiment is conducted on 10-bus power system and binary HCLPSO obtains superior performance compared to binary PSO, integer coded GA and other hybrid optimization methods. Then binary HCLPSO is employed to solve UC problem under uncertainty in load demand and renewable energies of solar and wind resources. The results showed that binary HCLPSO consistently performs well on solving UC problem with/without considering uncertainty in renewable resources.

7 CONCLUSION AND FUTURE WORKS

7.1 Conclusions

PSO is a population-based swarm intelligence algorithm in which each particle searches for the global optimum over the search space by following the previous experiences of the particles and the best experience of the whole swarm. As the particles are guided by the entire swarm's best experience, PSO can quickly converge to the optimum. Unfortunately, the best experience of the entire swarm may lead the particles to get stuck into the local optimum if the global optimum were located near a local optimum. In this way, PSO could suffer diversity loss during optimization process due to overexploitation. As a result, the premature convergence could occur and the global optimum could not be found. Therefore, heterogeneous comprehensive learning PSO (HCLPSO) algorithm is proposed in Chapter 3 to balance exploration and exploitation process of PSO algorithm. In the proposed HCLPSO algorithm, the entire population is divided into two heterogeneous subpopulations and each subpopulation group is assigned to perform exploration and exploitation process separately. In order to prevent diversity loss, the particles from the exploration subpopulation are not allowed to interact with the particles from the exploitation subpopulation. Moreover, time-varying inertia weight and time varying acceleration coefficients are used to bias the exploration and

exploitation balance at the early and late stages of the search process. The performance of the proposed HCLPSO is evaluated using CEC2005 benchmark problems and compared with current state-of-the-art PSO algorithms. The non-parametric statistical results showed that HCLPSO consistently performs well and overall outperforms other PSO algorithms.

According to no free lunch (NFL) theorem, an optimization algorithm cannot be expected to outperform other optimization algorithms on solving different kinds of optimization problems. An algorithm may outperform another on a problem, but may not perform well on other problems. For instance, some PSO algorithms perform well on unimodal optimization problems whereas other PSO algorithms provide better performance on multimodal optimization problems. Therefore, in order to enhance the universality and robustness of PSO performance, ensemble of PSO algorithm (EPSO) is proposed in Chapter 4. In the proposed EPSO algorithm, different PSO algorithms such as original PSO, LIPS, HPSO-TVAC, FDR-PSO and CLPSO are ensemble together and the self-adaptive mechanism is used to select the suitable PSO strategy to guide the particles in the current stage of the search process. The performance of EPSO algorithm is evaluated using unimodal, multimodal, expanded extended and hybrid composite optimization problems. The experimental results showed that the proposed EPSO algorithm outperforms its individual PSO strategies and recent state-of-the-art PSO algorithms. Overall, the EPSO performs consistently well on all four different types of optimization problems.

As mentioned before, the developed PSO algorithms consistently outperforms other state-of-the-art PSO algorithms in many standard benchmark problems. To evaluate the performance and applicability of the proposed algorithms, a real world UC problem is selected as a case study. Unit commitment problem is the economic operation of a power system, which determines the on/off scheduling of the generating units and the optimal amount of power of the committed units while satisfying the system and unit constraints. When integrating a power system with renewable resources, unit commitment problem faces challenges in balancing the power generation and load demand of the power system. Deterministic, metaheuristic and hybrid of deterministic and metaheuristic optimization methods have been proposed to solve the UC problem in the literature.

Firstly, the hybrid solution of deterministic and metaheuristic optimization is proposed to solve the UC problem without renewable resources. UC problem is decomposed into two sub-problems: unit-scheduling sub-problem (determining the feasible on/off unit schedule) and economic dispatch (determining the optimal amount of power to be generated by the committed units). As a deterministic method, the time-ahead priority listing method is proposed to solve the unit-scheduling sub-problem and HCLPSO algorithm proposed in Chapter 2 is used to tackle the economic dispatch sub-problem. The performance of the proposed hybrid solution (TPL+HCLPSO) is evaluated with the deterministic and metaheuristic optimization methods as well as other hybrid solutions. The comparison results demonstrated that the proposed hybrid solution consistently perform better than other compared optimization techniques in solving UC problems of 10-,20-,40-,60-,80- and 100-bus power systems.

In Chapter 6, UC problem is formulated with renewable resources (solar and wind) under uncertainty nature to 1) reflect the complexity real world applications and 2) reduce production cost and CO₂ emission cost. As it is impossible to consider the continuous state of forecasted error, a discrete probability set of 75 scenarios is represented for the uncertainty in load demand, solar and wind data. Binary HCLPSO algorithm is proposed to find the optimal scheduling of the UC problem with renewable resources. Optimizing scheduling of UC problem considering the uncertainty takes long execution time. Therefore, binary HCLPSO algorithm is employed to solve the UC problem of the 10-bus power system integrated with renewable resources of solar and wind energies. Before applying to binary HCLPSO algorithm to UC problem, the performance of BHCLPSO is compared with other binary PSO, integer-coded GA and the hybrid of binary and real-coded PSO algorithm. The binary HCLPSO consistently performs well on solving the UC problem with/without considering uncertainty in renewable resources.

7.2 Future Works

7.2.1 Futures works for PSO algorithm

PSO algorithm is highly efficient in finding the global optimum for small-scale problem. However, the algorithm suffers from the curse of dimensionality when solv-

ing large-scale optimization problems. The problem here is how to balance exploration in the vast search space and exploitation in the good regions with the limited number of particles. For future research, the followings are proposed for solving large-scale optimization problems.

As motioned earlier, the curse of dimensionality is one of the biggest challenges in large-scale optimization problems. Most PSO algorithms are mainly focused on the improvement of algorithm instead of understanding the relationship between inputs and results. In other research societies, such as data science community, they try to understand the input-output relationship to reduce dimensions or tackle the problem. As a result, data scientists could able to handle the large-scale data efficiently and effectively (e.g. Netflix recommender system). Thus, this resilient feature should be beneficial to PSO algorithm in narrowing down the search regions.

PSO algorithm has generated many input and output pairs during optimization process and utilizes only the local best and/or global best solutions in most algorithms. These input and output pairs from each function evaluation contains useful information and it can be used in the descriptive analytics, correlating between inputs and outputs, the influence of each dimension to the results through various data mining techniques. Using this information, PSO algorithm could –

- Compute better guidance vectors
- Narrow the search region
- Predict the candidate hot-spot search regions

Besides, the success of PSO as a single-objective swarm optimizer motivated the researchers to extend PSO to other optimization fields such as multi-objective optimization, constrained optimization, binary optimization, and combinatorial optimization. PSO also shows achievement in the study of those fields. However, there is still a lack of in-depth research on performance of PSO on those fields and it would be worth to investigate the followings to improve the performance of PSO in those optimization fields:

- Developing theoretical aspects

- Emphasis on efficiency
- Self-adaptation of parameters
- Niching techniques
- Learning mechanisms

7.2.2 Future works on solving power system problems by PSO

UC problem in power system is a nonlinear mix-integer combinatorial optimization power problem with two types of binary and continuous variable. Therefore, UC problem has been decomposed into two sub-problems: unit-scheduling sub-problem with binary decision variables representing on/off unit schedule and ED sub-problem with the continuous variable representing the power generation of levels of the units. As a result, UC variables are represented by different coding schemes or different optimization methods were used to handle the two sub-problems separately.

However, instead of using two different coding schemes or two different optimization techniques, mixed-integer formulation of UC constrained optimization problem can be formulated into floating-point arithmetic model and PSO can be used to solve the UC problem.

Besides UC problem in power system, PSO has been applied to solve other power system problems such as reactive power and voltage control, power system reliability and security, optimal power flow, power system controller design, etc. However, the capabilities of PSO are not limited to only those applications and as future works, PSO can also be applied to other nonlinear stochastic optimization problems such as generation expansion planning, multi-objective generator scheduling maintenance problem, capacitor placement problem, electricity machinery design problem and renewable energy systems.

BIBLIOGRAPHY

- [1] A. Eiben and C. Schippers, "On evolutionary exploration and exploitation," in *Annales Societatis Mathematicae Polonae. Series 4: Fundamenta Informaticae*, 1998, pp. 35-50.
- [2] M. CREPINSEK, S.-H. LIU, and M. MERNIK, "Exploration and Exploitation in Evolutionary Algorithms: A Survey," *ACM computing surveys*, vol. 45, 2013.
- [3] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization," in *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, 1999.
- [4] *U.S. Department of Energy, The SMART GRID: An Introduction, Technical Report, 2014.* Available: [http://energy.gov/sites/prod/files/oeprod/DocumentsandMedia/DOE_SG_Book_Single_Pages\(1\).pdf](http://energy.gov/sites/prod/files/oeprod/DocumentsandMedia/DOE_SG_Book_Single_Pages(1).pdf)
- [5] A. Madureira, J. Ferreira, and Z. Vale, *Computational Intelligence for Engineering Systems: Emergent Applications*: Springer Netherlands, 2010.
- [6] S. Sumathi, L. A. Kumar, and S. P, *Computational Intelligence Paradigms for Optimization Problems Using MATLAB®/SIMULINK®*: CRC Press, 2016.
- [7] H. Haibo, "Toward a Smart Grid: Integration of computational intelligence into Power Grid," in *Neural Networks (IJCNN), The 2010 International Joint Conference on*, 2010, pp. 1-6.
- [8] G. K. Venayagamoorthy, "Potentials and promises of computational intelligence for smart grids," in *Power & Energy Society General Meeting, 2009. PES '09. IEEE*, 2009, pp. 1-6.
- [9] P. J. Werbos, "Computational Intelligence for the Smart Grid-History, Challenges, and Opportunities," *IEEE Computational Intelligence Magazine*, vol. 6, pp. 14-21, 2011.

- [10] V. K. Jain, *Computational Intelligence*: Excel Books, 2010.
- [11] M. Pant, K. Deep, A. Nagar, and J. C. Bansal, *Proceedings of the Third International Conference on Soft Computing for Problem Solving: SocProS 2013*: Springer India, 2014.
- [12] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, 1995, pp. 1942-1948.
- [13] S. Yuhui and R. Eberhart, "A modified particle swarm optimizer," in *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*, 1998, pp. 69-73.
- [14] M. Clerc, "The swarm and the queen: towards a deterministic and adaptive particle swarm optimization," in *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, 1999.
- [15] M. Clerc and J. Kennedy, "The particle swarm - explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, pp. 58-73, 2002.
- [16] R. C. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," in *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, 2000, pp. 84-88 vol.1.
- [17] J. Kennedy and R. Mendes, "Neighborhood topologies in fully informed and best-of-neighborhood particle swarms," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 36, pp. 515-519, 2006.
- [18] J. Kennedy, "Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance," in *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, 1999, pp. 1-1938 Vol. 3.
- [19] J. Kennedy, "The particle swarm: social adaptation of knowledge," in *Evolutionary Computation, 1997., IEEE International Conference on*, 1997, pp. 303-308.
- [20] J. Kennedy and R. Mendes, "Population structure and particle swarm performance," in *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on*, 2002, pp. 1671-1676.
- [21] D. Watts and S. Strogatz, "Collective dynamics of'small-world'networks," *Nature*, vol. 393, pp. 440-442, 1998.
- [22] R. MENDES, J. KENNEDY, and J. NEVES, "The fully informed particle swarm: Simpler, maybe better," *IEEE Transactions on Evolutionary Computation*, vol. 8, pp. 204-210, 2004.

- [23] K. E. Parsopoulos and M. N. Vrahatis, "Unified Particle Swarm Optimization in Dynamic Environments," in *Applications of Evolutionary Computing*, vol. 3449, F. Rothlauf, J. Branke, S. Cagnoni, D. Corne, R. Drechsler, Y. Jin, *et al.*, Eds., ed: Springer Berlin Heidelberg, 2005, pp. 590-599.
- [24] M. Clerc, "Back to random topology," *Relatório Técnico*, mar, 2007.
- [25] Q. Ni and J. Deng, "A New Logistic Dynamic Particle Swarm Optimization Algorithm Based on Random Topology," *The Scientific World Journal*, vol. 2013, 2013.
- [26] X. Li, "Niching without niching parameters: particle swarm optimization using a ring topology," *IEEE Transactions on Evolutionary Computation*, vol. 14, pp. 150-169, 2010.
- [27] P. N. Suganthan, "Particle swarm optimiser with neighbourhood operator," in *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, 1999, pp. 1-1962 Vol. 3.
- [28] M. Nasir, S. Das, D. Maity, S. Sengupta, U. Halder, and P. Suganthan, "A dynamic neighborhood learning based particle swarm optimizer for global numerical optimization," *Information Sciences*, pp. 16-36, 2012.
- [29] S. B. Akat and V. Gazi, "Particle swarm optimization with dynamic neighborhood topology: three neighborhood strategies and preliminary results," in *IEEE Swarm Intelligence Symposium*, 2008.
- [30] M. Richards and D. A. Ventura, "Dynamic sociometry in particle swarm optimization," 2003.
- [31] T. Peram, K. Veeramachaneni, and C. K. Mohan, "Fitness-distance-ratio based particle swarm optimization," in *Swarm Intelligence Symposium, 2003. SIS'03. Proceedings of the 2003 IEEE*, 2003, pp. 174-181.
- [32] Q. Ni, C. Cao, and X. Yin, "A new dynamic probabilistic Particle Swarm Optimization with dynamic random population topology," in *Evolutionary Computation (CEC), 2014 IEEE Congress on*, 2014, pp. 1321-1327.
- [33] Y.-X. Wang and Q.-L. Xiang, "Particle swarms with dynamic ring topology," in *Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on*, 2008, pp. 419-423.
- [34] J. J. Liang and P. N. Suganthan, "Dynamic multi-swarm particle swarm optimizer," in *Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE*, 2005, pp. 124-129.
- [35] A. Engelbrecht, "Heterogeneous Particle Swarm Optimization," in *Swarm Intelligence*, vol. 6234, M. Dorigo, M. Birattari, G. Di Caro, R. Doursat, A. Engelbrecht, D. Floreano, *et al.*, Eds., ed: Springer Berlin Heidelberg, 2010, pp. 191-202.

- [36] B. J. Leonard, A. P. Engelbrecht, and A. B. van Wyk, "Heterogeneous particle swarms in dynamic environments," in *Swarm Intelligence (SIS), 2011 IEEE Symposium on*, 2011, pp. 1-8.
- [37] F. V. Nepomuceno and A. P. Engelbrecht, "A self-adaptive heterogeneous pso for real-parameter optimization," in *Evolutionary Computation (CEC), 2013 IEEE Congress on*, 2013, pp. 361-368.
- [38] F. Van den Bergh and A. P. Engelbrecht, "A cooperative approach to particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, pp. 225-239, 2004.
- [39] A. El Dor, M. Clerc, and P. Siarry, "A multi-swarm PSO using charged particles in a partitioned search space for continuous optimization," *Computational Optimization and Applications*, vol. 53, pp. 271-295, 2012.
- [40] R. Brits, A. P. Engelbrecht, and F. Van den Bergh, "A niching particle swarm optimizer," in *Proceedings of the 4th Asia-Pacific conference on simulated evolution and learning*, 2002, pp. 692-696.
- [41] T. Blackwell and P. J. Bentley, "Dynamic Search With Charged Swarms," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2002, pp. 19-26.
- [42] T. Blackwell and J. Branke, "Multiswarms, exclusion, and anti-convergence in dynamic environments," *IEEE Transactions on Evolutionary Computation*, vol. 10, pp. 459-472, 2006.
- [43] Y. Shengxiang and L. Changhe, "A Clustering Particle Swarm Optimizer for Locating and Tracking Multiple Optima in Dynamic Environments," *IEEE Transactions on Evolutionary Computation*, vol. 14, pp. 959-974, 2010.
- [44] M. Clerc, *Particle swarm optimization* vol. 93: John Wiley & Sons, 2010.
- [45] S. Janson and M. Middendorf, "A hierarchical particle swarm optimizer," in *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on*, 2003, pp. 770-776.
- [46] S. Janson and M. Middendorf, "A hierarchical particle swarm optimizer for dynamic optimization problems," in *Applications of evolutionary computing*, ed: Springer, 2004, pp. 513-524.
- [47] S. Janson and M. Middendorf, "A hierarchical particle swarm optimizer and its adaptive variant," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 35, pp. 1272-1282, 2005.
- [48] P. Ghosh, H. Zafar, S. Das, and A. Abraham, "Hierarchical dynamic neighborhood based particle swarm optimization for global optimization," in *Evolutionary Computation (CEC), 2011 IEEE Congress on*, 2011, pp. 757-764.

- [49] Y. Shi and R. Eberhart, "Parameter selection in particle swarm optimization," in *Evolutionary Programming VII*. vol. 1447, V. W. Porto, N. Saravanan, D. Waagen, and A. E. Eiben, Eds., ed: Springer Berlin Heidelberg, 1998, pp. 591-600.
- [50] Y. Shi and R. C. Eberhart, "Fuzzy adaptive particle swarm optimization," in *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on*, 2001, pp. 101-106.
- [51] L. Li, X. Bing, N. Ben, C. Yujuan, and W. Jianhuang, "The novel non-linear strategy of inertia weight in particle swarm optimization," in *Bio-Inspired Computing, 2009. BIC-TA '09. Fourth International Conference on*, 2009, pp. 1-5.
- [52] Z. Yong-ling, M. Long-hua, Z. Li-yan, and Q. Ji-xin, "Empirical study of particle swarm optimizer with an increasing inertia weight," in *Evolutionary Computation, 2003. CEC '03. The 2003 Congress on*, 2003, pp. 221-226 Vol.1.
- [53] Y.-l. Gao and Y.-h. Duan, "A New Particle Swarm Optimization Algorithm with Random Inertia Weight and Evolution Strategy," in *Computational Intelligence and Security Workshops, 2007. CISW 2007. International Conference on*, 2007, pp. 199-203.
- [54] M. Pant, T. Radha, and V. P. Singh, "Particle Swarm Optimization Using Gaussian Inertia Weight," in *Conference on Computational Intelligence and Multimedia Applications, 2007. International Conference on*, 2007, pp. 97-102.
- [55] J. Wu, X. He Xiang, W. Zhao, and R. Wang, "Exponential inertia weight particle swarm algorithm for dynamics optimization of electromechanical coupling system," in *Intelligent Computing and Intelligent Systems, 2009. ICIS 2009. IEEE International Conference on*, 2009, pp. 479-483.
- [56] A. Rezaee Jordehi and J. Jasni, "Parameter selection in particle swarm optimisation: a survey," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 25, pp. 527-542, 2013/12/01 2013.
- [57] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *IEEE Transactions on Evolutionary Computation*, vol. 8, pp. 240-255, 2004.
- [58] A. K. Renato and C. Leandro dos Santos, "Coevolutionary Particle Swarm Optimization Using Gaussian Distribution for Solving Constrained Optimization Problems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 36, pp. 1407-1416, 2006.
- [59] Z. Zhi-Hui, Z. Jun, L. Yun, and H. S. H. Chung, "Adaptive Particle Swarm Optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 39, pp. 1362-1381, 2009.

- [60] Y.-T. Juang, S.-L. Tung, and H.-C. Chiu, "Adaptive fuzzy particle swarm optimization for global optimization of multimodal functions," *Information Sciences*, vol. 181, pp. 4539-4549, 2011.
- [61] H. Mengqi, T. Wu, and J. D. Weir, "An Adaptive Particle Swarm Optimization With Multiple Adaptive Methods," *IEEE Transactions on Evolutionary Computation*, vol. 17, pp. 705-720, 2013.
- [62] P. J. Angeline, "Using selection to improve particle swarm optimization," in *IEEE World Congress on Computational Intelligence*, 1998, pp. 84-89.
- [63] C. Ying-ping, P. Wen-Chih, and J. Ming-Chung, "Particle Swarm Optimization With Recombination and Dynamic Linkage Discovery," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 37, pp. 1460-1470, 2007.
- [64] P. S. Andrews, "An Investigation into Mutation Operators for Particle Swarm Optimization," in *IEEE Congress on Evolutionary Computation, CEC 2006*, 2006, pp. 1044-1051.
- [65] W. Chengjian, H. Zhenya, Z. Yifeng, and P. Wenjiang, "Swarm directions embedded in fast evolutionary programming," in *Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on*, 2002, pp. 1278-1283.
- [66] Z. Wen-Jun and X. Xiao-Feng, "DEPSO: hybrid particle swarm with differential evolution operator," in *Systems, Man and Cybernetics, 2003. IEEE International Conference on*, 2003, pp. 3816-3821 vol.4.
- [67] N. Holden and A. A. Freitas, "A hybrid particle swarm/ant colony algorithm for the classification of hierarchical biological data," in *Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE*, 2005, pp. 100-107.
- [68] P. Shelokar, P. Siarry, V. K. Jayaraman, and B. D. Kulkarni, "Particle swarm and ant colony algorithms hybridized for improved continuous optimization," *Applied mathematics and computation*, vol. 188, pp. 129-142, 2007.
- [69] M. El-Abd, "Testing a Particle Swarm Optimization and Artificial Bee Colony Hybrid algorithm on the CEC13 benchmarks," in *Evolutionary Computation (CEC), 2013 IEEE Congress on*, 2013, pp. 2215-2220.
- [70] L. Cheng-Jian, C. Cheng-Hung, and L. Chin-Teng, "A Hybrid of Cooperative Particle Swarm Optimization and Cultural Algorithm for Neural Fuzzy Networks and Its Prediction Applications," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 39, pp. 55-68, 2009.
- [71] B.-Y. Qu, J. J. Liang, and P. N. Suganthan, "Niching particle swarm optimization with local search for multi-modal optimization," *Information Sciences*, vol. 197, pp. 131-143, 2012.

- [72] L. V. Santana-Quintero, N. Ramírez, and C. C. Coello, "A multi-objective particle swarm optimizer hybridized with scatter search," in *MICA 2006: Advances in Artificial Intelligence*, ed: Springer, 2006, pp. 294-304.
- [73] Y.-J. Wang, "Improving particle swarm optimization performance with local search for high-dimensional function optimization," *Optimization Methods & Software*, vol. 25, pp. 781-795, 2010.
- [74] B.-F. Liu, H.-M. Chen, J.-H. Chen, S.-F. Hwang, and S.-Y. Ho, "Meswarm: memetic particle swarm optimization," in *Proceedings of the 7th annual conference on Genetic and evolutionary computation*, 2005, pp. 267-268.
- [75] D. Parrott and L. Xiaodong, "Locating and tracking multiple dynamic optima by a particle swarm model using speciation," *IEEE Transactions on Evolutionary Computation*, vol. 10, pp. 440-458, 2006.
- [76] M. Daneshyari and G. G. Yen, "Cultural-Based Multiobjective Particle Swarm Optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 41, pp. 553-567, 2011.
- [77] W. N. Chen, J. Zhang, Y. Lin, N. Chen, Z. H. Zhan, H. S. H. Chung, *et al.*, "Particle Swarm Optimization With an Aging Leader and Challengers," *IEEE Transactions on Evolutionary Computation*, vol. 17, pp. 241-258, 2013.
- [78] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Transactions on Evolutionary Computation*, vol. 10, pp. 281-295, 2006.
- [79] Z. Zhi-Hui, Z. Jun, L. Yun, and S. Yu-hui, "Orthogonal Learning Particle Swarm Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 15, pp. 832-847, 2011.
- [80] L. Changhe, Y. Shengxiang, and N. Trung Thanh, "A Self-Learning Particle Swarm Optimizer for Global Optimization Problems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetic*, vol. 42, pp. 627-646, 2012.
- [81] C. Ran and J. Yaochu, "A Competitive Swarm Optimizer for Large Scale Optimization," *IEEE Transactions on Cybernetics*, vol. 45, pp. 191-204, 2015.
- [82] R. Zhigang, Z. Aimin, W. Changyun, and F. Zuren, "A Scatter Learning Particle Swarm Optimization Algorithm for Multimodal Problems," *IEEE Transactions on Cybernetics*, vol. 44, pp. 1127-1140, 2014.
- [83] M. A. M. de Oca, T. Stutzle, K. Van den Enden, and M. Dorigo, "Incremental Social Learning in Particle Swarms," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 41, pp. 368-384, 2011.
- [84] D. Chen and C. Zhao, "Particle swarm optimization with adaptive population size and its application," *Applied Soft Computing*, vol. 9, pp. 39-48, 1// 2009.
- [85] H. Sheng-Ta, S. Tsung-Ying, L. Chan-Cheng, and T. Shang-Jeng, "Efficient Population Utilization Strategy for Particle Swarm Optimizer," *IEEE*

Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, vol. 39, pp. 444-456, 2009.

- [86] L. Lanzarini, V. Leza, and A. De Giusti, "Particle Swarm Optimization with Variable Population Size," in *Artificial Intelligence and Soft Computing – ICAISC 2008*. vol. 5097, L. Rutkowski, R. Tadeusiewicz, L. Zadeh, and J. Zurada, Eds., ed: Springer Berlin Heidelberg, 2008, pp. 438-449.
- [87] J. Kennedy, "Bare bones particle swarms," in *Swarm Intelligence Symposium, 2003. SIS '03. Proceedings of the 2003 IEEE*, 2003, pp. 80-87.
- [88] R. A. Krohling and E. Mendel, "Bare Bones Particle Swarm Optimization with Gaussian or Cauchy jumps," in *Evolutionary Computation, 2009. CEC '09. IEEE Congress on*, 2009, pp. 3285-3291.
- [89] T. Blackwell, "A Study of Collapse in Bare Bones Particle Swarm Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 16, pp. 354-372, 2012.
- [90] M. Campos, R. A. Krohling, and I. Enriquez, "Bare Bones Particle Swarm Optimization With Scale Matrix Adaptation," *IEEE Transactions on Cybernetics*, vol. 44, pp. 1567-1578, 2014.
- [91] 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," *Generation, Transmission & Distribution, IET*, vol. 6, pp. 515-527, 2012.
- [92] M. Reyes-Sierra and C. C. Coello, "Multi-objective particle swarm optimizers: A survey of the state-of-the-art," *International journal of computational intelligence research*, vol. 2, pp. 287-308, 2006.
- [93] S. Mostaghim and J. Teich, "Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO)," in *Swarm Intelligence Symposium, 2003. SIS '03. Proceedings of the 2003 IEEE*, 2003, pp. 26-33.
- [94] M. Daneshyari and G. G. Yen, "Constrained Multiple-Swarm Particle Swarm Optimization Within a Cultural Framework," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 42, pp. 475-490, 2012.
- [95] A. Y. Saber and G. K. Venayagamoorthy, "Unit commitment with vehicle-to-Grid using particle swarm optimization," in *PowerTech, 2009 IEEE Bucharest*, 2009, pp. 1-8.
- [96] P. E. Onate Yumbla, J. M. Ramirez, and C. A. Coello Coello, "Optimal Power Flow Subject to Security Constraints Solved With a Particle Swarm Optimizer," *IEEE Transactions on Power Systems*, vol. 23, pp. 33-40, 2008.

- [97] S. Meie, Z. Zhi-Hui, C. Wei-Neng, G. Yue-Jiao, Z. Jun, and L. Yun, "Bi-Velocity Discrete Particle Swarm Optimization and Its Application to Multicast Routing Problem in Communication Networks," *IEEE Transactions on Industrial Electronics*, vol. 61, pp. 7141-7151, 2014.
- [98] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in *Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on*, 1997, pp. 4104-4108 vol.5.
- [99] G. Zwe-Lee, "Discrete particle swarm optimization algorithm for unit commitment," in *Power Engineering Society General Meeting, 2003, IEEE*, 2003, pp. 1-424 Vol. 1.
- [100] B. Dorronsoro and P. Bouvry, "Cellular genetic algorithms without additional parameters," *The Journal of Supercomputing*, vol. 63, pp. 816-835, 2013.
- [101] J. C. Inostroza and V. H. Hinojosa, "Short-term scheduling solved with a particle swarm optimiser," *Generation, Transmission & Distribution, IET*, vol. 5, pp. 1091-1104, 2011.
- [102] C. Po-Hung, "Two-Level Hierarchical Approach to Unit Commitment Using Expert System and Elite PSO," *IEEE Transactions on Power Systems*, vol. 27, pp. 780-789, 2012.
- [103] A. Y. Saber, T. Senjyu, A. Yona, and T. Funabashi, "Unit commitment computation by fuzzy adaptive particle swarm optimisation," *Generation, Transmission & Distribution, IET*, vol. 1, pp. 456-465, 2007.
- [104] T. O. Ting, M. V. C. Rao, and C. K. Loo, "A novel approach for unit commitment problem via an effective hybrid particle swarm optimization," *IEEE Transactions on Power Systems*, vol. 21, pp. 411-418, 2006.
- [105] X. Yuan, H. Nie, A. Su, L. Wang, and Y. Yuan, "An improved binary particle swarm optimization for unit commitment problem," *Expert Systems with Applications*, vol. 36, pp. 8049-8055, 5// 2009.
- [106] J. Yun-Won, P. Jong-Bae, J. Se-Hwan, and K. Y. Lee, "A New Quantum-Inspired Binary PSO: Application to Unit Commitment Problems for Power Systems," *IEEE Transactions on Power Systems*, vol. 25, pp. 1486-1495, 2010.
- [107] B. Zhao, C. X. Guo, B. R. Bai, and Y. J. Cao, "An improved particle swarm optimization algorithm for unit commitment," *International Journal of Electrical Power & Energy Systems*, vol. 28, pp. 482-490, 9// 2006.
- [108] K. T. Chaturvedi, M. Pandit, and L. Srivastava, "Self-Organizing Hierarchical Particle Swarm Optimization for Nonconvex Economic Dispatch," *IEEE Transactions on Power System*, vol. 23, pp. 1079-1087, 2008.

- [109] P. Jong-Bae, L. Ki-Song, S. Joong-Rin, and K. Y. Lee, "A particle swarm optimization for economic dispatch with nonsmooth cost functions," *IEEE Transactions on Power Systems*, vol. 20, pp. 34-42, 2005.
- [110] S. Jun, V. Palade, W. Xiao-Jun, F. Wei, and W. Zhenyu, "Solving the Power Economic Dispatch Problem With Generator Constraints by Random Drift Particle Swarm Optimization," *IEEE Transactions on Industrial Informatics*, vol. 10, pp. 222-232, 2014.
- [111] M. Ke, W. Hong Gang, D. Zhaoyang, and W. Kit Po, "Quantum-Inspired Particle Swarm Optimization for Valve-Point Economic Load Dispatch," *IEEE Transactions on Power Systems*, vol. 25, pp. 215-222, 2010.
- [112] T. Niknam and H. Doagou-Mojarrad, "Multiobjective economic/emission dispatch by multiobjective particle swarm optimisation," *Generation, Transmission & Distribution, IET*, vol. 6, pp. 363-377, 2012.
- [113] T. Niknam and F. Golestaneh, "Enhanced adaptive particle swarm optimisation algorithm for dynamic economic dispatch of units considering valve-point effects and ramp rates," *Generation, Transmission & Distribution, IET*, vol. 6, pp. 424-435, 2012.
- [114] B. K. Panigrahi, V. Ravikumar Pandi, and S. Das, "Adaptive particle swarm optimization approach for static and dynamic economic load dispatch," *Energy Conversion and Management*, vol. 49, pp. 1407-1415, 6// 2008.
- [115] A. I. Selvakumar and K. Thanushkodi, "A New Particle Swarm Optimization Solution to Nonconvex Economic Dispatch Problems," *IEEE Transactions on Power Systems*, vol. 22, pp. 42-51, 2007.
- [116] G. Zwe-Lee, "Particle swarm optimization to solving the economic dispatch considering the generator constraints," *IEEE Transactions on Power Systems*, vol. 18, pp. 1187-1195, 2003.
- [117] M. A. Abido, "Optimal power flow using particle swarm optimization," *International Journal of Electrical Power & Energy Systems*, vol. 24, pp. 563-571, 10// 2002.
- [118] N. Mo, Z. Y. Zou, K. W. Chan, and T. Y. G. Pong, "Transient stability constrained optimal power flow using particle swarm optimisation," *Generation, Transmission & Distribution, IET*, vol. 1, pp. 476-483, 2007.
- [119] X. Shiwei, C. Ka Wing, B. Xuefeng, and G. Zhizhong, "Enhanced particle swarm optimisation applied for transient angle and voltage constrained discrete optimal power flow with flexible AC transmission system," *Generation, Transmission & Distribution, IET*, vol. 9, pp. 61-74, 2015.

- [120] S. Ganguly, "Multi-Objective Planning for Reactive Power Compensation of Radial Distribution Networks With Unified Power Quality Conditioner Allocation Using Particle Swarm Optimization," *IEEE Transactions on Power Systems*, vol. 29, pp. 1801-1810, 2014.
- [121] L. Srivastava and H. Singh, "Hybrid multi-swarm particle swarm optimisation based multi-objective reactive power dispatch," *Generation, Transmission & Distribution, IET*, vol. 9, pp. 727-739, 2015.
- [122] H. Yoshida, K. Kawata, Y. Fukuyama, S. Takayama, and Y. Nakanishi, "A particle swarm optimization for reactive power and voltage control considering voltage security assessment," *IEEE Transactions on Power Systems*, vol. 15, pp. 1232-1239, 2000.
- [123] W. Zhang and Y. Liu, "Multi-objective reactive power and voltage control based on fuzzy optimization strategy and fuzzy adaptive particle swarm," *International Journal of Electrical Power & Energy Systems*, vol. 30, pp. 525-532, 11// 2008.
- [124] B. Zhao, C. X. Guo, and Y. J. Cao, "A multiagent-based particle swarm optimization approach for optimal reactive power dispatch," *IEEE Transactions on Power Systems*, vol. 20, pp. 1070-1078, 2005.
- [125] A. Y. Abdelaziz, F. M. Mohammed, S. F. Mekhamer, and M. A. L. Badr, "Distribution Systems Reconfiguration using a modified particle swarm optimization algorithm," *Electric Power Systems Research*, vol. 79, pp. 1521-1530, 11// 2009.
- [126] A. Ameli, S. Bahrami, F. Khazaeli, and M. R. Haghifam, "A Multiobjective Particle Swarm Optimization for Sizing and Placement of DGs from DG Owner's and Distribution Company's Viewpoints," *IEEE Transactions on Power Delivery*, vol. 29, pp. 1831-1840, 2014.
- [127] J. Xiaoling, Z. Jianguo, S. Ying, L. Kejun, and Z. Boqin, "Distribution network reconfiguration for load balancing using binary particle swarm optimization," in *Power System Technology, 2004. PowerCon 2004. 2004 International Conference on*, 2004, pp. 507-510 Vol.1.
- [128] A. T. Al-Awami, Y. L. Abdel-Magid, and M. A. Abido, "A particle-swarm-based approach of power system stability enhancement with unified power flow controller," *International Journal of Electrical Power & Energy Systems*, vol. 29, pp. 251-259, 3// 2007.
- [129] S. B. Karanki, M. K. Mishra, and B. K. Kumar, "Particle Swarm Optimization-Based Feedback Controller for Unified Power-Quality Conditioner," *IEEE Transactions on Power Delivery*, vol. 25, pp. 2814-2824, 2010.
- [130] G. Zwe-Lee, "A particle swarm optimization approach for optimum design of PID controller in AVR system," *IEEE Transactions on Energy Conversion*, vol. 19, pp. 384-391, 2004.

- [131] H. Siahkali and M. Vakilian, "Electricity generation scheduling with large-scale wind farms using particle swarm optimization," *Electric Power Systems Research*, vol. 79, pp. 826-836, 5// 2009.
- [132] S. Sheng-Yi, L. Chan-Nan, C. Rung-Fang, Gutie, x, and G. rrez-Alcaraz, "Distributed Generation Interconnection Planning: A Wind Power Case Study," *IEEE Transactions on Smart Grid*, vol. 2, pp. 181-189, 2011.
- [133] W. Shuli, L. Hai, F. Qiang, D. C. Yu, and Z. Lijun, "Economic Allocation for Energy Storage System Considering Wind Power Distribution," *IEEE Transactions on Power Systems*, vol. 30, pp. 644-652, 2015.
- [134] A. Hassan, M. Saadawi, M. Kandil, and M. Saeed, "Modified particle swarm optimisation technique for optimal design of small renewable energy system supplying a specific load at Mansoura University," *Renewable Power Generation, IET*, vol. 9, pp. 474-483, 2015.
- [135] W. Jidong and Y. Fan, "Optimal capacity allocation of standalone wind/solar/battery hybrid power system based on improved particle swarm optimisation algorithm," *Renewable Power Generation, IET*, vol. 7, pp. 443-448, 2013.
- [136] A. Y. Saber and G. K. Venayagamoorthy, "Resource Scheduling Under Uncertainty in a Smart Grid With Renewables and Plug-in Vehicles," *IEEE Systems Journal*, vol. 6, pp. 103-109, 2012.
- [137] R. V. Kulkarni and G. K. Venayagamoorthy, "Particle Swarm Optimization in Wireless-Sensor Networks: A Brief Survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 41, pp. 262-267, 2011.
- [138] Y. Jun, Z. Hesheng, L. Yun, P. Cheng, and S. Wei, "Task Allocation for Wireless Sensor Network Using Modified Binary Particle Swarm Optimization," *IEEE Sensors Journal*, vol. 14, pp. 882-892, 2014.
- [139] E. R. Dosciatti, W. Godoy, and A. Foronda, "TQ/PSO - A New Scheduler to Optimize the Time Frame with PSO in WiMAX Networks," *Latin America Transactions, IEEE (Revista IEEE America Latina)*, vol. 13, pp. 365-376, 2015.
- [140] L. Shu-Han, C. Chien-Ching, and H. Min-Hui, "Location optimisation for antennas by asynchronous particle swarm optimisation," *Communications, IET*, vol. 7, pp. 1510-1516, 2013.
- [141] Y. Rahmat-Samii, J. M. Kovitz, and H. Rajagopalan, "Nature-Inspired Optimization Techniques in Communication Antenna Designs," *Proceedings of the IEEE*, vol. 100, pp. 2132-2144, 2012.

- [142] A. A. Minasian and T. S. Bird, "Particle Swarm Optimization of Microstrip Antennas for Wireless Communication Systems," *IEEE Transactions on Antennas and Propagation*, vol. 61, pp. 6214-6217, 2013.
- [143] J. Pugh and A. Martinoli, "Inspiring and Modeling Multi-Robot Search with Particle Swarm Optimization," in *Swarm Intelligence Symposium, 2007. SIS 2007. IEEE*, 2007, pp. 332-339.
- [144] W. Jatmiko, K. Sekiyama, and T. Fukuda, "A pso-based mobile robot for odor source localization in dynamic advection-diffusion with obstacles environment: theory, simulation and measurement," *IEEE Computational Intelligence Magazine*, vol. 2, pp. 37-51, 2007.
- [145] J. Chia-Feng and C. Yu-Cheng, "Evolutionary-Group-Based Particle-Swarm-Optimized Fuzzy Controller With Application to Mobile-Robot Navigation in Unknown Environments," *IEEE Transactions on Fuzzy System*, vol. 19, pp. 379-392, 2011.
- [146] H. Bevrani, F. Habibi, P. Babahajyani, M. Watanabe, and Y. Mitani, "Intelligent Frequency Control in an AC Microgrid: Online PSO-Based Fuzzy Tuning Approach," *IEEE Transactions on Smart Grid*, vol. 3, pp. 1935-1944, 2012.
- [147] C. Hsueh-Hsien, L. Lung-Shu, C. Nanming, and L. Wei-Jen, "Particle-Swarm-Optimization-Based Nonintrusive Demand Monitoring and Load Identification in Smart Meters," *IEEE Transactions on Industry Applications*, vol. 49, pp. 2229-2236, 2013.
- [148] S. Jing Jun and L. Kay-Soon, "Photovoltaic Model Identification Using Particle Swarm Optimization With Inverse Barrier Constraint," *IEEE Transactions on Power Electronics*, vol. 27, pp. 3975-3983, 2012.
- [149] L. Chien-Hung and Y.-Y. Hsu, "Design of a Self-Tuning PI Controller for a STATCOM Using Particle Swarm Optimization," *IEEE Transactions on Industrial Electronics*, vol. 57, pp. 702-715, 2010.
- [150] C. F. Juang and C. F. Lu, "Load-frequency control by hybrid evolutionary fuzzy PI controller," *IEEE Proceeding of Generation, Transmission and Distribution*, vol. 153, pp. 196-204, 2006.
- [151] P. Ghamisi, M. S. Couceiro, F. M. L. Martins, and J. Atli Benediktsson, "Multilevel Image Segmentation Based on Fractional-Order Darwinian Particle Swarm Optimization," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, pp. 2382-2394, 2014.
- [152] C.-Y. Chen and Y. Fun, "Particle swarm optimization algorithm and its application to clustering analysis," in *Networking, Sensing and Control, 2004 IEEE International Conference on*, 2004, pp. 789-794 Vol.2.
- [153] Z. Xiangrong, W. Wenna, L. Yangyang, and L. C. Jiao, "PSO-based automatic relevance determination and feature selection system for hyperspectral image classification," *Electronics Letters*, vol. 48, pp. 1263-1265, 2012.

- [154] A. Paoli, F. Melgani, and E. Pasolli, "Clustering of Hyperspectral Images Based on Multiobjective Particle Swarm Optimization," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, pp. 4175-4188, 2009.
- [155] D. W. van der Merwe and A. P. Engelbrecht, "Data clustering using particle swarm optimization," in *Evolutionary Computation, 2003. CEC '03. The 2003 Congress on*, 2003, pp. 215-220 Vol.1.
- [156] G. Das, P. K. Pattnaik, and S. K. Padhy, "Artificial Neural Network trained by Particle Swarm Optimization for non-linear channel equalization," *Expert Systems with Applications*, vol. 41, pp. 3491-3496, 6/1/ 2014.
- [157] K. Hyunseok and C. Seongju, "High-Resolution Touch Floor System Using Particle Swarm Optimization Neural Network," *IEEE Sensors Journal*, vol. 13, pp. 2084-2093, 2013.
- [158] Y. Wei-Chang, "New Parameter-Free Simplified Swarm Optimization for Artificial Neural Network Training and its Application in the Prediction of Time Series," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, pp. 661-665, 2013.
- [159] A. Chatterjee, K. Pulasinghe, K. Watanabe, and K. Izumi, "A particle-swarm-optimized fuzzy-neural network for voice-controlled robot systems," *IEEE Transactions on Industrial Electronics*, vol. 52, pp. 1478-1489, 2005.
- [160] C. Kit Yan, C. K. F. Yiu, T. S. Dillon, S. Nordholm, and L. Sai Ho, "Enhancement of Speech Recognitions for Control Automation Using an Intelligent Particle Swarm Optimization," *IEEE Transactions on Industrial Informatics*, vol. 8, pp. 869-879, 2012.
- [161] J. I. Ababneh and M. H. Bataineh, "Linear phase FIR filter design using particle swarm optimization and genetic algorithms," *Digital Signal Processing*, vol. 18, pp. 657-668, 7// 2008.
- [162] S. Jun, F. Wei, and X. Wenbo, "A Quantum-Behaved Particle Swarm Optimization With Diversity-Guided Mutation for the Design of Two-Dimensional IIR Digital Filters," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 57, pp. 141-145, 2010.
- [163] M. Rongyong, Y. Leilei, and L. Jiang, "Improvement of particle swarm optimization and its application in the earth-rock dam instability risk calculation," in *Consumer Electronics, Communications and Networks (CECNet), 2012 2nd International Conference on*, 2012, pp. 2642-2646.
- [164] L. dos Santos Coelho and V. C. Mariani, "An Efficient Particle Swarm Optimization Approach Based on Cultural Algorithm Applied to Mechanical Design," in *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, 2006, pp. 1099-1104.

- [165] S. A. H. Soliman and A. A. H. Mantawy, *Modern Optimization Techniques with Applications in Electric Power Systems*: Springer New York, 2011.
- [166] M. Shahidehpour, H. Yamin, and Z. Li, *Market Operations in Electric Power Systems: Forecasting, Scheduling, and Risk Management*: Wiley, 2003.
- [167] R. H. Kerr, J. L. Scheidt, A. J. Fontanna, and J. K. Wiley, "Unit Commitment," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-85, pp. 417-421, 1966.
- [168] T. Senjyu, K. Shimabukuro, K. Uezato, and T. Funabashi, "A fast technique for unit commitment problem by extended priority list," *IEEE Transactions on Power Systems*, vol. 18, pp. 882-888, 2003.
- [169] F. N. Lee, "Short-term thermal unit commitment-a new method," *IEEE Transactions on Power Systems*, vol. 3, pp. 421-428, 1988.
- [170] J. J. Hargreaves and B. F. Hobbs, "Commitment and Dispatch With Uncertain Wind Generation by Dynamic Programming," *IEEE Transactions on Sustainable Energy*, vol. 3, pp. 724-734, 2012.
- [171] x, L. lvarez, x, J. pez, Go, x, *et al.*, "Commitment of Combined Cycle Plants Using a Dual Optimization–Dynamic Programming Approach," *IEEE Transactions on Power Systems*, vol. 26, pp. 728-737, 2011.
- [172] W. L. Snyder, H. D. Powell, and J. C. Rayburn, "Dynamic Programming Approach to Unit Commitment," *IEEE Transactions on Power Systems*, vol. 2, pp. 339-348, 1987.
- [173] J. Quanyuan, Z. Boran, and Z. Mingze, "Parallel augment Lagrangian relaxation method for transient stability constrained unit commitment," *IEEE Transactions on Power Systems*, vol. 28, pp. 1140-1148, 2013.
- [174] F. Zhuang and F. D. Galiana, "Towards a more rigorous and practical unit commitment by Lagrangian relaxation," *IEEE Transactions on Power Systems*, vol. 3, pp. 763-773, 1988.
- [175] A. I. Cohen and M. Yoshimura, "A Branch-and-Bound Algorithm for Unit Commitment," *IEEE Power Engineering Review*, vol. PER-3, pp. 34-35, 1983.
- [176] C. Chern-Lin and S. C. Wang, "Branch-and-bound scheduling for thermal generating units," *IEEE Transactions on Energy Conversion*, vol. 8, pp. 184-189, 1993.
- [177] L. Xiang, L. Tiejian, W. Jiahua, W. Guangqian, and W. W. G. Yeh, "Hydro Unit Commitment via Mixed Integer Linear Programming: A Case Study of the Three Gorges Project, China," *IEEE Transactions on Power Systems*, vol. 29, pp. 1232-1241, 2014.
- [178] T. S. Dillon, K. W. Edwin, H. D. Kochs, and R. J. Taud, "Integer Programming Approach to the Problem of Optimal Unit Commitment with Probabilistic

- Reserve Determination," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-97, pp. 2154-2166, 1978.
- [179] M. Piekutowski and I. A. Rose, "A Linear Programming Method for Unit Commitment Incorporated Generator Configurations, Reserve, and Flow Constraints," *IEEE Power Engineering Review*, vol. PER-5, pp. 45-46, 1985.
 - [180] L. Tao and M. Shahidehpour, "Price-based unit commitment: a case of Lagrangian relaxation versus mixed integer programming," *IEEE Transactions on Power Systems*, vol. 20, pp. 2015-2025, 2005.
 - [181] S. Li, S. M. Shahidehpour, and C. Wang, "Promoting the application of expert systems in short-term unit commitment," *IEEE Transactions on Power Systems*, vol. 8, pp. 286-292, 1993.
 - [182] S. Saneifard, N. R. Prasad, and H. A. Smolleck, "A fuzzy logic approach to unit commitment," *IEEE Transactions on Power Systems*, vol. 12, pp. 988-995, 1997.
 - [183] H. Sasaki, M. Watanabe, J. Kubokawa, N. Yorino, and R. Yokoyama, "A solution method of unit commitment by artificial neural networks," *IEEE Transactions on Power Systems*, vol. 7, pp. 974-981, 1992.
 - [184] A. H. Mantawy, Y. L. Abdel-Magid, and S. Z. Selim, "Unit commitment by tabu search," *IEEE Proceedings of Generation, Transmission and Distribution*, vol. 145, pp. 56-64, 1998.
 - [185] A. Y. Saber, T. Senjyu, T. Miyagi, N. Urasaki, and T. Funabashi, "Unit commitment by heuristics and absolutely stochastic simulated annealing," *Generation, Transmission & Distribution, IET*, vol. 1, pp. 234-243, 2007.
 - [186] K. A. Juste, H. Kita, E. Tanaka, and J. Hasegawa, "An evolutionary programming solution to the unit commitment problem," *IEEE Transactions on Power System*, vol. 14, pp. 1452-1459, 1999.
 - [187] T. Senjyu, H. Yamashiro, K. Shimabukuro, K. Uezato, and T. Funabashi, "Fast solution technique for large-scale unit commitment problem using genetic algorithm," *IEEE Proceedings of Generation, Transmission and Distribution*, vol. 150, pp. 753-760, 2003.
 - [188] D. Datta and S. Dutta, "A binary-real-coded differential evolution for unit commitment problem," *International Journal of Electrical Power & Energy Systems*, vol. 42, pp. 517-524, 11// 2012.
 - [189] S. P. Simon, N. P. Padhy, and R. S. Anand, "An ant colony system approach for unit commitment problem," *International Journal of Electrical Power & Energy Systems*, vol. 28, pp. 315-323, 6// 2006.

- [190] C. C. A. Rajan and M. R. Mohan, "An evolutionary programming-based tabu search method for solving the unit commitment problem," *IEEE Transactions on Power Systems*, vol. 19, pp. 577-585, 2004.
- [191] C. Chuan-Ping, L. Chih-Wen, and L. Chun-Chang, "Unit commitment by Lagrangian relaxation and genetic algorithms," *IEEE Transactions on Power Systems*, vol. 15, pp. 707-714, 2000.
- [192] H. H. Balci and J. F. Valenzuela, "Scheduling electric power generators using particle swarm optimization combined with the Lagrangian relaxation method," *International Journal of Applied Mathematics and Computer Science*, vol. 14, pp. 411-422, 2004.
- [193] K. E. Parsopoulos and M. N. Vrahatis, "On the computation of all global minimizers through particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, pp. 211-224, 2004.
- [194] A. P. Engelbrecht, *Fundamentals of Computational Swarm Intelligence*: Wiley, 2005.
- [195] P. Suganthan, N. Hansen, J. Liang, K. Deb, Y.-P. Chen, A. Auger, *et al.*, "Problem Definitions and Evaluation Criteria for the CEC2005 Special Session on Real-Parameter Optimization," 2005.
- [196] J. Liang, B. Qu, and P. Suganthan, "Problem definitions and evaluation criteria for the CEC2014 special session and competition on single objective real-parameter numerical optimization," *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore*, 2013.
- [197] A. Griewank and P. L. Toint, "On the Unconstrained Optimization of Partially Separable Functions," in *Nonlinear Optimization 1981*, ed: Academic Press, 1982, pp. pp. 301–312.
- [198] T. Bartz-Beielstein, *Experimental Research in Evolutionary Computation: The New Experimentalism*: Springer Berlin Heidelberg, 2006.
- [199] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, pp. 3-18, 3// 2011.
- [200] J. D. Gibbons and S. Chakraborti, *Nonparametric Statistical Inference, Fifth Edition*: CRC Press, 2010.
- [201] Y. F. Li, Z. H. Zhan, Y. Lin, and J. Zhang, "Comparisons study of APSO OLPSO and CLPSO on CEC2005 and CEC2014 test suits," in *2015 IEEE Congress on Evolutionary Computation (CEC)*, 2015, pp. 3179-3185.
- [202] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, pp. 67-82, 1997.

- [203] O. Olorunda and A. Engelbrecht, "An analysis of heterogeneous cooperative algorithms," in *Proceedings of the Eleventh conference on Congress on Evolutionary Computation*, 2009, pp. 1562-1569.
- [204] H. Wang, Z. Wu, S. Rahnamayan, H. Sun, Y. Liu, and J.-s. Pan, "Multi-strategy ensemble artificial bee colony algorithm," *Information Sciences*, vol. 279, pp. 587-603, 9/20/ 2014.
- [205] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, pp. 398-417, 2009.
- [206] R. Mallipeddi, S. Mallipeddi, and P. N. Suganthan, "Ensemble strategies with adaptive evolutionary programming," *Information Sciences*, vol. 180, pp. 1571-1581, 5/1/ 2010.
- [207] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Micro Machine and Human Science, 1995. MHS '95., Proceedings of the Sixth International Symposium on*, 1995, pp. 39-43.
- [208] K. E. Parsopoulos, *Particle Swarm Optimization and Intelligence: Advances and Applications: Advances and Applications*: IGI Global, 2010.
- [209] B. Y. Qu, P. N. Suganthan, and S. Das, "A Distance-Based Locally Informed Particle Swarm Model for Multimodal Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 17, pp. 387-402, 2013.
- [210] J. E. Baker, "Reducing bias and inefficiency in the selection algorithm," in *Proceedings of the second international conference on genetic algorithms*, 1987, pp. 14-21.
- [211] J. Liang, B. Qu, P. Suganthan, and A. G. Hernández-Díaz, "Problem definitions and evaluation criteria for the CEC2013 special session on real-parameter optimization," *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, Technical Report*, vol. 201212, 2013.
- [212] A. J. Wood, B. F. Wollenberg, and G. B. Sheblé, *Power Generation, Operation and Control*: Wiley, 2013.
- [213] B. Qu and P. Suganthan, "Constrained multi-objective optimization algorithm with an ensemble of constraint handling methods," *Engineering Optimization*, vol. 43, pp. 403-416, 2011.
- [214] S. A. Kazarlis, A. G. Bakirtzis, and V. Petridis, "A genetic algorithm solution to the unit commitment problem," *IEEE Transactions on Power Systems*, vol. 11, pp. 83-92, 1996.

- [215] T. Senjyu, T. Miyagi, A. Y. Saber, N. Urasaki, and T. Funabashi, "Emerging solution of large-scale unit commitment problem by Stochastic Priority List," *Electric Power Systems Research*, vol. 76, pp. 283-292, 3// 2006.
- [216] T. Yang and T. O. Ting, "Methodological Priority List for Unit Commitment Problem," in *Computer Science and Software Engineering, 2008 International Conference on*, 2008, pp. 176-179.
- [217] W. Ongsakul and N. Petcharaks, "Unit commitment by enhanced adaptive Lagrangian relaxation," *IEEE Transactions on Power Systems*, vol. 19, pp. 620-628, 2004.
- [218] R. Quan, J.-b. Jian, and Y.-d. Mu, "Tighter relaxation method for unit commitment based on second-order cone programming and valid inequalities," *International Journal of Electrical Power & Energy Systems*, vol. 55, pp. 82-90, 2// 2014.
- [219] A. Y. Saber and G. K. Venayagamoorthy, "Intelligent unit commitment with vehicle-to-grid —A cost-emission optimization," *Journal of Power Sources*, vol. 195, pp. 898-911, 2/1/ 2010.
- [220] I. G. Damousis, A. G. Bakirtzis, and P. S. Dokopoulos, "A solution to the unit-commitment problem using integer-coded genetic algorithm," *IEEE Transactions on Power Systems*, vol. 19, pp. 1165-1172, 2004.

AUTHOR'S PUBLICATIONS

Conference Papers

1. **N. Lynn**, P.N. Suganthan, "Comprehensive learning particle swarm optimizer with guidance vector selection", in proceedings of IEEE Swarm Intelligence Symposium (SIS), pp. 80-84, Singapore, 2013.
2. **N. Lynn**, P.N. Suganthan, "Distance-based Locally Informed Particle Swarm Optimizer with dynamic population size", in proceedings of 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems pp. 576-587, Singapore, 2014.
3. **N. Lynn**, R. Mallipeddi, P.N. Suganthan, "Differential Evolution with Two Subpopulations", in proceedings of 5th International Conference on Swarm, Evolutionary and Memetic Computing, India, 2014. (Application: unit commitment problem)
4. **N. Lynn**, R. Mallipeddi, P.N. Suganthan, "Self-adaptive Differential Evolution with Ensembles of Strategies and Sampled Parameter Values for Unit Commitment", in proceedings of 6th International Conference on Swarm, Evolutionary and Memetic Computing, India, 2015.
5. **N. Lynn**, P. N. Suganthan, "Modified Artificial Bee Colony Algorithm with Comprehensive Learning Re-initialization Strategy," in proceedings of IEEE International Conference on Systems, Man and Cybernetics, Hong Kong, 2015.

Journal Papers

1. **N. Lynn**, P. N. Suganthan, “Heterogeneous Comprehensive Learning Particle Swarm Optimization with Enhanced Exploration and Exploitation (HCLPSO),” *Swarm and Evolutionary Computation Journal*, vol. 24, pp. 11-24, 2015.
2. **N. Lynn**, P. N. Suganthan, “Ensemble Particle Swarm Optimizer”, *Applied Soft Computing Journal*, 2015 – revised and submitted.
3. **N. Lynn**, R. Mallipeddi, P. N. Suganthan, N. Srikanth “Unit Commitment using New Priority Listing Method and Heterogeneous Comprehensive Learning Particle Swarm Optimization,” *IEEE Transaction on Power Systems Journal*, 2015-submitted.
4. **N. Lynn**, P.N. Suganthan, M. Z. Ali “A Survey on Population Topologies for Evolutionary and Swarm Algorithms” – submitted
5. A. Rajasekhar, **N. Lynn**, S. Das and P.N. Suganthan, “Computing with the Collective Intelligence of Honey Bees – A Survey”, *Swarm and Evolutionary Computation Journal*, 2016 – accepted.

Appendix 1: Benchmark Functions

CEC2005 benchmark functions [195]

Function Type	Function Name	Initialization Range	Search Range	$F(x^*)$ f_bias
Unimodal Functions	F_1 : Shifted Sphere Function	$[-100,100]^D$	$[-100,100]^D$	-450
	F_2 : Shifted Schwefel's Problem 1.2	$[-100,100]^D$	$[-100,100]^D$	-450
	F_3 : Shifted Rotated High Conditioned Elliptic Function	$[-100,100]^D$	$[-100,100]^D$	-450
	F_4 : Shifted Schwefel's Problem 1.2 with Noise in Fitness	$[-100,100]^D$	$[-100,100]^D$	-450
	F_5 : Schwefel's Problem 2.6 with Global Optimum on Bounds	$[-100,100]^D$	$[-100,100]^D$	-310
Multimodal Functions	F_6 : Shifted Rosenbrock's Function	$[-100,100]^D$	$[-100,100]^D$	390
	F_7 : Shifted Rotated Griewank's Function without Bounds	$[0,600]^D$	$[-600,600]^D$	-180
	F_8 : Shifted Rotated Ackley's Function with Global Optimum on Bounds	$[-32,32]^D$	$[-32,32]^D$	-140
	F_9 : Shifted Rastrigin's Function	$[-5,5]^D$	$[-5,5]^D$	-330
	F_{10} : Shifted Rotated Rastrigin's Function	$[-5,5]^D$	$[-5,5]^D$	-330
	F_{11} : Shifted Rotated Weierstrass Function	$[-0.5, 0.5]^D$	$[-0.5, 0.5]^D$	90
	F_{12} : Schwefel's Problem 2.13	$[-100,100]^D$	$[-100,100]^D$	-460
Expanded Functions	F_{13} : Expanded Extended Griewank's plus Rosenbrock's Function(F8F2)	$[-3,1]^D$	$[-3,1]^D$	-130
	F_{14} : Shifted Rotated Expanded Scaffer's F6	$[-100,100]^D$	$[-100,100]^D$	-300
Hybrid Com- position Func- tions	F_{15} : Hybrid Composition Function	$[-5,5]^D$	$[-5,5]^D$	120
	F_{16} : Rotated Hybrid Composition Function	$[-5,5]^D$	$[-5,5]^D$	120
	F_{17} : Rotated Hybrid Composition Function with Noise in Fitness	$[-5,5]^D$	$[-5,5]^D$	120
	F_{18} : Rotated Hybrid Composition Function	$[-5,5]^D$	$[-5,5]^D$	10
	F_{19} : Rotated Hybrid Composition Function with a Narrow Basin for the Global Optimum	$[-5,5]^D$	$[-5,5]^D$	10
	F_{20} : Rotated Hybrid Composition Function with the Global Optimum on the Bounds	$[-5,5]^D$	$[-5,5]^D$	10
	F_{21} : Rotated Hybrid Composition Function	$[-5,5]^D$	$[-5,5]^D$	360
	F_{22} : Rotated Hybrid Composition Function with High Condition Number Matrix	$[-5,5]^D$	$[-5,5]^D$	360
	F_{23} : Non-Continuous Rotated Hybrid Composition Function	$[-5,5]^D$	$[-5,5]^D$	360
	F_{24} : Rotated Hybrid Composition Function	$[-5,5]^D$	$[-5,5]^D$	260
	F_{25} : Rotated Hybrid Composition Function without Bounds	$[-2,5]^D$	$[-5,5]^D$	260

CEC2014 benchmark functions [196]

Function Type	Function Name	$F^*=F(x^*)$
Unimodal Functions	F_1 : Rotated High Conditioned Elliptic Function	100
	F_2 : Rotated Bent Cigar Function	200
	F_3 : Rotated Discus Function	300
Multimodal Functions	F_4 : Shifted and Rotated Rosenbrock's Function	400
	F_5 : Shifted and Rotated Ackley's Function	500
	F_6 : Shifted and Rotated Weierstrass Function	600
	F_7 : Shifted and Rotated Griewank's Function	700
	F_8 : Shifted Rastrigin's Function	800
	F_9 : Shifted and Rotated Rastrigin's Function	900
	F_{10} : Shifted Schwefel's Function	1000
	F_{11} : Shifted and Rotated Schwefel's Function	1100
	F_{12} : Shifted and Rotated Katsuura Function	1200
	F_{13} : Shifted and Rotated HappyCat Function	1300
	F_{14} : Shifted and Rotated HGBat Function	1400
	F_{15} : Shifted and Rotated Expanded Griewank's plus Rosenbrock's Function	1500
	F_{16} : Shifted and Rotated Expanded Scaffer's F6 Function	1600
Hybrid Function 1	F_{17} : Hybrid Function 1 (N=3)	1700
	F_{18} : Hybrid Function 2 (N=3)	1800
	F_{19} : Hybrid Function 3 (N=4)	1900
	F_{20} : Hybrid Function 4 (N=4)	2000
	F_{21} : Hybrid Function 5 (N=5)	2100
	F_{22} : Hybrid Function 6 (N=5)	2200
Composition Function	F_{23} : Composition Function 1 (n=5)	2300
	F_{24} : Composition Function 2 (n=3)	2400
	F_{25} : Composition Function 3 (n=3)	2500
	F_{26} : Composition Function 4 (n=5)	2600
	F_{27} : Composition Function 5 (n=5)	2700
	F_{28} : Composition Function 6 (n=5)	2800
	F_{29} : Composition Function 7 (n=3)	2900
	F_{30} : Composition Function 8 (n=3)	3000
Search Range: [-100,100] ^p		

CEC2013 benchmark functions [211]

Function Type	Function Name	$F^*=F(x^*)$
Unimodal Functions	F_1 : Sphere Function	-1400
	F_2 : Rotated High Conditioned Elliptic Function	-1300
	F_3 : Rotated Bent Cigar Function	-1200
	F_4 : Rotated Discus Function	-1100
	F_5 : Different Powers Function	-1000
Multimodal Functions	F_6 : Rotated Rosenbrock's Function	-900
	F_7 : Rotated Schaffers F_7 Function	-800
	F_8 : Rotated Ackley's Function	-700
	F_9 : Rotated Weierstrass Function	-600
	F_{10} : Rotated Griewank's Function	-500
	F_{11} : Rastrigin's Function	-400
	F_{12} : Rotated Rastrigin's Function	-300
	F_{13} : Non-Continuous Rotated Rastrigin's Function	-200
	F_{14} : Schwefel's Function	-100
	F_{15} : Rotated Schwefel's Function	100
	F_{16} : Rotated Katsuura Function	200
	F_{17} : Lunacek Bi_Rastrigin Function	300
	F_{18} : Rotated Lunacek Bi_Rastrigin Function	400
Hybrid Composition Functions	F_{19} : Expanded Griewank's plus Rosenbrock's Function	500
	F_{20} : Expanded Scaffer's F_6 Function	600
	F_{21} : Composition Function 1 (n=5,Rotated)	700
	F_{22} : Composition Function 2 (n=3,Unrotated)	800
	F_{23} : Composition Function 3 (n=3,Rotated)	900
	F_{24} : Composition Function 4 (n=3,Rotated)	1000
	F_{25} : Composition Function 5 (n=3,Rotated)	1100
	F_{26} : Composition Function 6 (n=5,Rotated)	1200
	F_{27} : Composition Function 7 (n=5,Rotated)	1300
	F_{28} : Composition Function 3 (n=5,Rotated)	1400
Search Range: [-100,100] ^D		

Appendix 2: Convergence Characteristic Graphs of HCLPSO Algorithm for CEC2005 Benchmark Functions

(a) F_1

(b) F_2

(c) F_3

(d) F_4

(e) F_5

(f) F_6

(g) F_7

(h) F_8

(i) F_9

(j) F_{10}

(k) F_{11}

(l) F_{12}

(m) F_{13}

(n) F_{14}

(o) F_{15}

(p) F_{16}

(q) F_{17}

(r) F_{18}

(s) F_{19}

(t) F_{20}

(u) F_{21}

(v) F_{21}

(w) F_{22}

(x) F_{23}

(y) F_{25}

Appendix 3: Convergence Characteristic Graphs of EPSO Algorithm for CEC2005 Benchmark Functions

(a) F_1

(b) F_2

(c) F_3

(d) F_4

(e) F_5

(f) F_6

(g) F_7

(h) F_8

(i) F_9

(j) F_{10}

(k) F_{11}

(l) F_{12}

(m) F_{13}

(n) F_{14}

(o) F_{15}

(p) F_{16}

(q) F_{17}

(r) F_{18}

(s) F_{19}

(t) F_{20}

(u) F_{21}

(v) F_{22}

(w) F_{23}

(x) F_{24}

(y) F_{25}