

Thesis
on
**HYBRID SEARCH METHOD FOR
OPTIMIZATION IN REAL PARAMETER SPACE**

for fulfillment of the requirements
of
DOCTOR OF PHILOSOPHY

Submitted By

Nirbhow Jap Singh
Regd. No.: 950904009

Under the supervision of

Dr. J.S. Dhillon

and

Dr. D.P. Kothari



**Electrical and Instrumentation Engineering Department
Thapar University, Patiala**

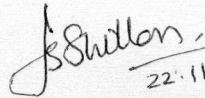
(Declared as Deemed-to-be-University u/s 3 of the UGC Act., 1956)

Punjab (India)-147004

June, 2017

Thesis Certificate

This is to certify that the thesis titled **Hybrid Search Method for Optimization in Real Parameter Space**, submitted by **NirbhawJap Singh**, to the Thapar University, Patiala for the award of the degree of **Doctor of Philosophy**, is a bona fide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.


22.11.2017

Dr. J.S. Dhillon

Professor

Electrical and Instrumentation
Engineering Department, Sant
Longowal Institute of Engineering
and Technology, Longowal-148106,
Punjab, India



Dr. D.P. Kothari

DEAN(R & D), JD College of
Engineering & Management
Nagpur-441501, Maharashtra,
India

Place:

Date:

ACKNOWLEDGMENT

First of all I thank the Almighty, for giving me the grace and health to complete my research work. I would like to express my deepest gratitude to my guides and mentors **Dr. J.S. Dhillon**, and **Dr. D.P. Kothari** for giving me an opportunity to complete my thesis under their able supervision. They have always been a pillar of support and constant source of inspiration. This thesis would never have been possible without their kind support and valuable guidance.

I feel highly obliged to **Mrs. Amrit Kaur Dhillon** wife of Dr. J. S. Dhillon for her being so kind, co-operative and generous during the whole period of research work.

I sincerely thanks my all colleagues especially, **Ms. Manbir kaur**, **Dr. Nitin Narang**, **Dr. Shakti Singh**, **Dr. Mukesh Singh** and **Dr. Vikram** for encouraging and supporting me throughout. I would like to thank my Ph.D. doctoral committee members **Dr. S.K. Jain** and **Dr. M.K. Sharma** for their time and valuable suggestion during my research period.

I am also extremely grateful and thankful to my father **S. Ranjit Singh** and mother **Mrs. Surinder Kaur**, my brother **Mr. Amrinder Singh**, relatives and friends for their infinite patience, love and unending support. At last but not the least, I would like to thank my wife **Jasmeet Kaur**, daughters **Roohani Jot Kaur**, **Harnoor Kaur** and son **Sahib Asis Singh** for their unconditional support, inspiration and love.

Mr. Nirbhow Jap Singh
Regn. No. 950904009
Thapar University, Patiala

ABSTRACT

Many real world problems need to optimize available resources to meet the desired objectives. It is an essential part for the better performance of a system. Optimization helps the planners to have suitable efficient options appropriate for their various applications and needs. Although, the optimization process has been characterized in distinct scenarios, yet the objective is always to obtain a set of decision variables in the feasible search space that maximizes beneficial objectives of the system. A practical optimization problem involves simultaneous optimization of several incommensurable and often competing objectives. In such a case, there is no single optimal solution, but rather a set of possible solutions. These solutions are optimal in the wider sense that no other solutions in the search space are superior to them, when considering all the objectives together.

Broadly, multi-objective optimization techniques are grouped under two major categories: non-interactive and interactive. In the non-interactive method, a global performance function of the objectives is identified and optimized with respect to the constraints. On the other hand, in the interactive method, a local preference function or trade-off among objectives is identified by interacting with decision makers (DMs) and the solution process proceeds gradually towards a global satisfactory solution. The solution procedure for these problems involves two parts: Firstly, a setup to generate feasible solution and secondly, a tool to select compromise solution that best suits the conflicting objectives of the problem.

Nowadays, the use of optimization algorithms is an essential part of the multiobjective solution approach, as these are responsible to generate potential solution of considered problem in feasible search space. The Evolutionary Algorithm (EA) is an effective tool to explore the problem's search area. Among the available optimization techniques, the

EAs are widely accepted. The initial development phase of EA's was dominated by Evolutionary Programming (EP), due to its single operator based mechanism. Later on, Particle Swarm Optimization (PSO) algorithm has received wide attention from the research community due to better exploration and exploitation capabilities. Recently, the Differential Evolution (DE) algorithm has gotten popular among the research community, due to its well balanced diversification and intensification process. Owing to improve the performance of a standard algorithm, hybridization is the possible alternative. It means that although, an appropriate standard algorithm results in better performance and still its performance can be further improved if standard algorithm is combined with specific heuristics or operators that incorporate domain knowledge. Furthermore, hybrid algorithms emphasize on the complementary advantage of population-based search (exploration) and their refinement (exploitative) by local search techniques. The exploration provides a consistent estimate of the global optimum, while the exploitative part concentrates on the search effort around the best solution. The combined method extracts the virtues of the parent approaches and tries to materialize efficient solution approach. Depending on the introduction of local search technique along with population based method in the search process, the hybrid algorithms are classified as before/after/interleaved procedure or mixture of these.

On the other hand, the application of non-linear dynamics has recently drawn attention in the field of optimization. One of the applications is the use of chaos map to select algorithm-dependent parameters replacing uniform random variables. The empirical studies have shown that chaotic sequences have a high level of mixing capability and thus it is expected that their use to replace fixed procedures, may result in a better balance of the exploration and exploitation. So, in the context of improving performance of EP, PSO and DE, a useful diversity is ensured by combining deterministic chaotic sequence with them.

Among the wide variety of optimization applications, power system scheduling is a realistic active application of the optimization process. The power dispatch problem from the field of power system optimization presents a challenging task to EAs. The dispatch problems are categorized as single objective, such as Economic Load Dispatch (ELD),

Minimum Emission Dispatch (MED) and Multi-objective Load Dispatch (MOLD). The ELD problem aims to minimize the power generation cost, whereas the MED is directed to minimize the gaseous pollutant emission. The MOLD intends to minimize the two conflicting objectives power generation cost and pollutant emission simultaneously. Apart, from these objectives the system associated constraints of ramp-rate limits, avoiding prohibited operating zone, valve point loading effect and multiple fuel source options are also considered for practical realization. Therefore, in the above stated power system dispatch problem, the motive is to obtain a generation schedule in the feasible search space that minimizes associated objectives, while satisfying the system operational constraints.

After careful study of the recent reported work in literature, it has been observed that: no single optimization technique suits best for various kinds of real life problems, direct search methods are better at local exploitation of solution but are sensitive to initial guess and step size. The random search methods EP, PSO and DE have proven their effectiveness as a better exploration, search process, but still they have limitations. The performance issues of these techniques are improved by hybridizing these techniques with local search techniques or adding heuristics. Another alternative incorporated is the application of chaotic sequence to modify the evolution and selection operators used in EP, PSO and DE algorithms. Apart from this, the multi-objective problem handling procedures and constraint handling techniques is another direction of research.

In the light of aforementioned gaps, the main contributions of this research are: The evolutionary programming algorithm is coordinated along with Powell's pattern search and chaotic sequence and a technique chaotic evolutionary programming and Powell's search is proposed. Further, the predator prey optimization (PPO) performance is improved using concepts of collaborative behavior, psychology along with local search techniques. The proposed algorithms are termed as synergic predator prey optimization (SPPO), adaptive predator prey optimization (APPO) and fly and walk predator prey optimization (FWPO). Lastly, the unification of differential evolution algorithm with Powell's search and chaotic sequence is proposed as chaotic differential evolution and Powell's search (CDEPS). The formulated hybrid algorithms, namely: CEPPS, SPPO, APPO, FWPO and CDEPS have been validated using the standard test problems. The

nature of functions selected for experimentation is uni-modal, multi-modal, separable, non-separable, continuous and discontinuous. In order to validate the formulated algorithms on real world problem, the economic dispatch problems for the multi fuel system, Taiwan power system consisting of 40 generators and Korean power system having 140 generators are implemented. Finally, multi-objective load dispatch has been considered for performance validation.

The limitations of weighting method to handle conflicting objective of multi-objective problem is elevated by using it as an interactive approach. In addition, surrogate worth trade off method (SWT) is incorporated in the research work to handle multi-objective problem, in interactive and non-interactive manner. The constraints being an essential constituent of real world power system problems, the constraints are handled either by using the concept of slack generator or by incorporation of power distribution.

In this thesis, the research work is organized in seven chapters. In the first chapter, the importance of optimization, eminent milestones in the development of optimization procedures is presented. The review of approaches to solve multi-objective optimization problems has been reviewed. The power system dispatch problem being a realistic application of optimization, the significant contributions of various authors related to dispatch problems and its relevant topics have been briefly reviewed in this chapter. The requisite theoretical, mathematical, and computational backgrounds of different local and global optimization techniques are reviewed which are utilized in the present study to determine the solution of optimization problems.

Second chapter, introduces a synergic predator-prey optimization (SPPO) algorithm to solve economic load dispatch problem for thermal units with practical aspects. The basic PPO model comprises prey and predator as essential components. In the SPPO, the decision making of prey is bifurcated into corroborative and impeded parts. It comprises four behaviors namely inertial, cognitive, collective swarm intelligence, and prey's individual and neighborhood concern of predator. In order to, improve the quality of prey swarm, which influence the convergence rate, opposition based initialization is used. In order to, verify robustness of proposed algorithm general benchmark problems and small, medium, and large power generation test power systems are simulated.

The chapter three, incorporates chaotic evolutionary programming and pattern search (CEPPS) as solution procedure, In CEPPS, the chaotic mutation operator is used in the evolution phase. Similarly, the selection operation is also a chaotic sequence guided. In order to, ensure exploitation, Powell's pattern search operator is included, which operates under stochastic process. The surrogate worth trade off approach is utilized to select a compromised solution from the Pareto front, which suits both the considered objectives in the feasible search space. The generalized test functions and realistic MOLD problem is considered to investigate the performance of the proposed solution approach. The performance comparison of CEPPS and chaotic evolutionary programming, applying on generalized benchmark test functions, shows that CEPPS has better search capabilities. However, while implementing dispatch problems, the CEPPS has resulted in premature convergence.

Fourth chapter, presents adaptive predator-prey optimization (APPO) to solve MOLD problem with objectives of operating cost and pollutant emission. In APPO, the fear factor from predator is based on the cognitive and social behavior of prey, to ensure continuous mobility (magnitude and direction) of prey, resulting in better diversification of solutions in the domain throughout the search process. In this chapter, the multi-objective optimization problem is handled by weighting method, whereby the weight pattern assigned to the objectives has been undertaken as decision variables. This results in non-inferior solutions at each swarm move. In order to select a best-compromised solution, fuzzy theory is applied. The performance of the proposed algorithm is investigated on power system test problems. The proposed method provides better results in terms of fuel cost and pollutant emission. In addition, the better satisfaction level of both the conflicting objectives, well distributed Pareto front, acceptable solution in a single trial run and insensitivity to parameter variations is observed in comparison to other existing methods reported in literature.

The chapter five, presents a fuzzy surrogate worth trade-off approach to decide the preferred solution among the non-dominated solutions' set. The uniformity of non-dominated solutions' set is maintained exploiting a quality measure approach. This chapter describes a novel foraging activity paradigm to generate non-dominated solutions. The presented

approach comprises of fly and walk behavior of preys. The direction of turn heuristic of prey handles system's equality constraint. The search algorithm balances exploration and exploitation and handles system constraints autonomously. The performance of the proposed algorithm is investigated using generalized benchmark functions and multi-fuel, medium and large power system MTPLD problems. The experimental results show that the proposed approach is robust, depends on least parameters and retains Pareto quality in independent trials while generating non-dominated solutions in a trial run.

Sixth chapter, utilizes a chaotic differential evolution and Powell's pattern search algorithm (CDEPS) is proposed to solve MOLD problem. The chaotic differential evolution method is responsible for the diversification and Powell's pattern search is dedicated to exploitation. Further, the performance of two CDEPS variants based on Gauss map and Tent map is investigated. Another objective of this chapter is to introduce SWT as a non-interactive approach to solve MOLD problem. The performance analysis is done using generalized benchmark test functions and complex MTPLD problems. The Wilcoxon's test is used to analyze the experimental results. The exhaustive analysis shows that the Tent map based CDEPS has better ability to generate quality generation schedule with faster convergence rate.

Finally, the chapter seven, presents a brief summary and conclusions of all the chapters and the recommendation for further research.

List of Publications

Publication (SCI) from research work (chapter wise)

1. **(Chapter 2)**

Singh, N.J., Dhillon, J.S. and Kothari, D.P., 2016. Synergic predator-prey optimization for economic thermal power dispatch problem. Elsevier, **Applied Soft Computing**, 43, pp.298-311 (impact factor 2.857).

2. **(Chapter 5)**

Singh, N.J., Dhillon, J.S. and Kothari, D.P. Surrogate worth trade off Multi-objective thermal power load dispatch. Submitted to Elsevier, **Energy**, 138, pp.1112-1123. (impact factor 4.292).

3. **(Chapter 6)**

Singh, N.J., Dhillon, J.S. and Kothari, D.P., 2017. Multi-objective thermal power load dispatch using chaotic differential evolutionary algorithm and Powell's method. Springer, **Soft Computing**, pp.1-16, (impact factor 1.630).

Publication (SCI) under review (chapter wise)

1. **(Chapter 4)**

Singh, N.J., Dhillon, J.S. and Kothari, D.P. Multi-objective thermal power load dispatch using adaptive predator prey optimization. submitted to Elsevier, **Applied Soft Computing**, (submitted after minor revision) (impact factor 2.857).

2. **(Chapter 6)**

Singh, N.J., Dhillon, J.S. and Kothari, D.P. Non-interactive approach to solve multi-objective thermal power dispatch problem using composite search algorithm. Submitted to Elsevier, **Applied Soft Computing**, (submitted after minor revision) (impact factor 2.857).

Contents

Acknowledgements	i
Abstract	iii
Publications	ix
List of Tables	xiv
List of Figures	xvii
List of Abbreviations	xx
List of Symbols	xxii
1 INTRODUCTION	1
1.1 OPTIMIZATION PROBLEM	4
1.1.1 Single Objective Optimization Problem	5
1.1.2 Multiobjective Optimization Problem	7
1.2 OPTIMIZATION TECHNIQUES	9
1.2.1 Conventional Techniques	9
1.2.2 Random Search Techniques	10
1.2.3 Hybrid Search Techniques	18
1.3 MULTIOBJECTIVE OPTIMIZATION	23
1.3.1 Multi objective Evolutionary Algorithm	28
1.3.2 Metrics for Multiobjective Evolutionary Algorithm	28
1.4 ECONOMIC LOAD DISPATCH	29
1.5 MULTIOBJECTIVE LOAD DISPATCH	34
1.6 SCOPE OF WORK	39
1.7 OBJECTIVES	40
1.8 OUTLINE OF THESIS	40
2 SYNERGIC PREDATOR PREY OPTIMIZATION FOR ECONOMIC LOAD DISPATCH	44
2.1 INTRODUCTION	44
2.2 ECONOMIC LOAD DISPATCH PROBLEM	47
2.3 CONSTRAINT HANDLING	49
2.3.1 Equality Constraints Handling	49
2.3.2 Ramp Rate Limit Constraint Handling	50
2.3.3 Prohibited Operating Zone Constraint Handling	50
2.4 SYNERGIC PREDATOR PREY OPTIMIZATION	51
2.4.1 Representation of Prey Particles	52
2.4.2 Initialization of Prey Swarm	52
2.4.3 Opposition Based Prey Swarm Initialization	52

Contents Continued...

2.4.4	Prey Swarm Velocity Initiation	53
2.4.5	Predator Velocity and Position Initialization	53
2.4.6	Generation of Feasible Solution	54
2.4.7	Updating Prey Positions	54
2.5	EXPERIMENTAL RESULTS AND ANALYSIS	58
2.5.1	Test Systems	58
2.5.2	Generalized Test Functions	60
2.5.3	Test Power System-1	62
2.5.4	Test Power System-2	64
2.5.5	Test Power System-3	65
2.5.6	Statistical Performance Analysis of SPPO Algorithm	67
2.6	CONCLUSIONS	71
3	MULTIOBJECTIVE LOAD DISPATCH USING CHAOTIC EVOLUTIONARY PROGRAMMING AND POWELL'S METHOD	72
3.1	INTRODUCTION	72
3.2	MULTI-OBJECTIVE LOAD DISPATCH	75
3.3	PROPOSED HYBRID SEARCH ALGORITHM	77
3.3.1	Chaotic Evolutionary Programming	77
3.3.2	Powell's Pattern Search	78
3.4	SOLUTION METHODOLOGY	79
3.4.1	Solution Representation and Initialization	79
3.4.2	Non-inferior Solution Set Reformation	80
3.4.3	Surrogate Worth Trade-off Decision Maker	80
3.4.4	Population Individuals Update	82
3.5	SIMULATION CASES AND RESULTS	84
3.5.1	Test Systems	84
3.5.2	Generalized Test Functions	84
3.5.3	Test Power System-1	87
3.5.4	Test Power System-2	88
3.5.5	Test Power System-3	92
3.5.6	Robustness Analysis	93
3.6	CONCLUSION	94
4	MULTIOBJECTIVE LOAD DISPATCH USING ADAPTIVE PREDATOR PREY OPTIMIZATION	95
4.1	INTRODUCTION	95
4.2	ADAPTIVE PREDATOR PREY OPTIMIZATION	99
4.2.1	Discrete Velocity Variation	100
4.2.2	Adaptive Velocity Limit	101
4.3	WEIGHTING SUM METHOD	101
4.4	FUZZY DECISION MAKING	102

4.5	SOLUTION METHODOLOGY	103
4.5.1	Representation of Swarm	104
4.5.2	Initialization of Swarm and Predator Position	104
4.5.3	Heuristic Random Search to Generate Feasible Solution	105
4.5.4	Constraint Handling	105
4.5.5	Objective Function	107
4.5.6	Position Update of Prey and Predator	108
4.6	SIMULATION CASES AND RESULTS	108
4.6.1	Test Power Systems	108
4.6.2	Parameter Setting	110
4.6.3	Test Power System-1	110
4.6.4	Test Power System-2	112
4.6.5	Test Power System-3	116
4.6.6	Robustness Analysis	117
4.6.7	Parameter sensitivity analysis	120
4.7	CONCLUSIONS	120
5	SURROGATE WORTH TRADE-OFF METHOD FOR MULTIOBJECTIVE LOAD DISPATCH	122
5.1	INTRODUCTION	122
5.2	NATURAL FORAGING BEHAVIOR BASED PPO	125
5.2.1	Prey Position Update: Fly, walk and Direction of turn model	125
5.2.2	Predator Position Modification	128
5.3	Proposed Solution Approach	128
5.3.1	Solution representation and initialization	128
5.3.2	FWPO Components Update	130
5.3.3	Pareto Uniformity Measure	130
5.4	SIMULATION CASES AND RESULTS	131
5.4.1	Parameter Setting	131
5.4.2	Generalized Test Functions	131
5.4.3	Test Power Systems-1	133
5.4.4	Test Power System-2	136
5.4.5	Test Power System-3	138
5.4.6	Quality Measure Analysis	140
5.4.7	Sensitivity Analysis of FWPO Algorithm	140
5.5	CONCLUSION	142
6	MULTIOBJECTIVE LOAD DISPATCH USING CHAOTIC DIFFERENTIAL EVOLUTION AND POWELL'S METHOD	144
6.1	INTRODUCTION	144
6.2	PROPOSED CDEPS ALGORITHM	147
6.2.1	Chaotic Differential Evolution	147
6.3	SOLUTION METHODOLOGY	149
6.3.1	Non-Interactive Approach	149
6.3.2	Interactive Solution Approach	151
6.4	SIMULATION CASES AND RESULTS	153
6.4.1	Generalized Test Functions	153
6.4.2	Test Power System-1	157
6.4.3	Test Power System-2	159
6.4.4	Test Power System-3	162

6.4.5	Convergence Behavior Analysis	164
6.4.6	Robustness Analysis	165
6.4.7	Wilcoxon's Signed Rank Test	167
6.5	CONCLUSION	167
7	CONCLUSIONS	169
7.1	INTRODUCTION	169
7.2	SIGNIFICANT CONTRIBUTIONS	169
7.3	SUGGESTIONS FOR FURTHER RESEARCH	174
	REFERENCES	176
	Appendices	201
A	RANDOM SEARCH TECHNIQUES	202
A.1	PARTICLE SWARM OPTIMIZATION (PSO)	202
A.2	PREDATOR PREY OPTIMIZATION (PPO)	203
A.3	ANTI PREDATOR PREY OPTIMIZATION (APSO)	204
A.4	DIFFERENTIAL EVOLUTION (DE)	205
A.5	EVOLUTIONARY PROGRAMMING (EP)	206
B	DIRECT SEARCH METHODS	207
B.1	POWELL'S METHOD	207
B.2	HOOKES JEEVE'S METHOD	207
C	GENERALIZED BENCHMARK FUNCTIONS	210
D	TEST POWER SYSTEMS	213
D.1	SYSTEM DATA FOR TEST POWER SYSTEM-1	213
D.2	SYSTEM DATA FOR TEST POWER SYSTEM-2	217
D.3	SYSTEM DATA FOR TEST POWER SYSTEM-3	219

List of Tables

Table no	Caption	Page
1.1	Prominent evolutionary algorithms	11
1.2	Metrics for multiobjective algorithm performance	29
2.1	Generalized bench mark functions considered for analysis of SPPO	60
2.2	Electrical power test systems undertaken for performance analysis	61
2.3	SPPO parameter setting	61
2.4	Analysis of generalized test functions under investigation (10-D) [380]	62
2.5	Test power system-1, ELD cost comparison ($P_D=2700MW$)	63
2.6	Test power system-1, ELD Power distribution plan by SPPO ($P_D=2700$ MW)	63
2.7	Test power system-1, ELD Power distribution plan by SPPO ($P_D = 2700MW$)	64
2.8	Test power system-2, ELD cost comparison ($P_D = 10500MW$)	66
2.9	Test power system 2, ELD power distribution plan by SPPO ($P_D = 10500MW$)	66
2.10	Test system-3, ELD cost comparison	67
2.11	Test system-3, Optimal power generation schedule (MW) using SPPO	68
2.12	Analysis of the parameter sensitivity (Test power System-3, Case-1)	69
2.13	Test power system-3, case-1, Descriptive statistics	69
2.14	Student's T-test for sample mean ($\alpha = 0.05$)	69
2.15	One-sample Wilcoxon signed rank test ($\alpha = 0.05$)	69
3.1	Performance analysis of generalized benchmark test functions	85
3.2	Test power system-1, Comparison of ELD ($P_D = 2700MW$)	87
3.3	Test power system-1, Cost and emission for ELD and MED respectively	88
3.4	Test power system-1, MOLD solution comparison ($P_D = 2700MW$)	89
3.5	Test power system-1, MOLD solution comparison ($P_D = 2700MW$)	89
3.6	Test power system 1, MOLD Power distribution ($P_D = 2700MW$)	89
3.7	Test power system-1, MOLD Power distribution ($P_D=2700$)	89
3.8	Test power system-2, Optimal solution (cost, emission) comparison ($P_D =$ $10500MW$)	90
3.9	Test power system-2, Power generation (in MW) schedule by CEPPS ($P_D =$ $10500MW$)	91
3.10	Test power system 3, ELD solution comparison ($P_D = 49342MW$)	91
3.11	Test power system 3, Solution comparison ($P_D = 49342MW$)	92
3.12	Test power system 3, MOLD solution comparison ($P_D = 49342MW$)	92
3.13	Test power system-3, Optimal power allocation using CEPPS-2 (in MW)	93
3.14	Test power system-1, Descriptive statistics	93
4.1	Test power system-1, comparison of cost ($P_D = 2700MW$)	110
4.2	Test power system-1, Cost and emission for ELD and MED respectively	111
4.3	Test power system-1, MOLD solution comparison ($P_D = 2700MW$)	111
4.4	Test power system-1, MOLD solution comparison ($P_D = 2700MW$)	111
4.5	Test power system 1, MOLD Power distribution ($P_D = 2700MW$)	113

List of Tables Continued...

Table no	Caption	Page
4.6	Test power system-1, MOLD Power distribution ($P_D=2700$)	114
4.7	Test power system-2, Optimal solution (cost , emission) comparison ($P_D=10500$ MW)	115
4.8	Test system-3, MOLD power schedule by APPO ($P_D = 10500MW$)	115
4.9	Test power system 3, ELD solution comparison ($P_D = 49342MW$)	117
4.10	Test power system-3, MOLD solution comparison ($P_D=49342$ MW)	117
4.11	Test system 3, MOLD Power distribution by APPO ($P_D = 49342MW$)	118
4.12	Descriptive statistics Test power system-3	119
4.13	Analysis of the parameter sensitivity Test power system-2	119
5.1	Optimal values of generalized test functions using FWPO	133
5.2	Test power system-1, comparison of cost ($P_D = 2700$)	134
5.3	Test power system-1, Cost and emission for ELD and MED respectively	134
5.4	Test power system-1, MOLD solution comparison ($P_D = 2700MW$)	135
5.5	Test power system-1, MOLD solution comparison ($P_D = 2700MW$)	135
5.6	Test power system-1, MOLD solution comparison ($P_D = 2700MW$)	135
5.7	Test power system 1, MOLD Power distribution ($P_D = 2700MW$)	136
5.8	Test power system-1, MOLD Power distribution ($P_D=2700$)	136
5.9	Test power system-2, Optimal solution comparison ($P_D=10500$ MW)	137
5.10	Test power system-2, Power schedule plan for ELD, MED and MOLD ($P_D=10500$ MW)	139
5.11	Test power system-3, Optimal cost comparison ($P_D=49342$ MW)	139
5.12	Test power system-3, MOLD compromised solution comparison	139
5.13	Test power system-3, Optimal power schedule plan ($P_D=49342$ MW)	141
5.14	Swarm size (N_P) vs (cost and emission) analysis by FWPO for Test system-3	141
6.1	Performance analysis of generalized benchmark test functions	156
6.2	Test power system-1, comparison of cost ($P_D = 2700MW$)	157
6.3	Test power system-1, Cost and emission for ELD and MED respectively	158
6.4	Test power system-1, MOLD solution comparison ($P_D = 2700MW$)	159
6.5	Test power system-1, MOLD solution comparison ($P_D = 2700MW$)	159
6.6	Test power system-2, Optimal solution comparison ($P_D=10500$ MW)	160
6.7	Test power system-2, Comparison of MOLD solution using interactive and non-interactive approach ($P_D = 10500MW$)	161
6.8	Test system-3, MOLD power schedule by APPO ($P_D = 10500MW$)	162
6.9	Test power system 3, ELD solution comparison ($P_D = 49342MW$)	163
6.10	Test power system 3, Extremum of cost and emission objective functions	164
6.11	Test power system 3, MOLD solution comparison ($P_D = 49342MW$)	164
6.12	Test system 3, Optimal generation schedule for MOLD ($P_D=49342MW$)	165

List of Tables Continued...

Table no	Caption	Page
6.13	Test power system-3, Descriptive statistics	167
6.14	Test power system 3, Wilcoxon's signed rank test outcomes	167
D.1	Power generator's cost coefficients	214
D.2	Test power system-1, Fuel option and corresponding power generator limits	215
D.3	Test power system-1, Prohibited operating zone (POZ)	215
D.4	Gaseous pollutant's emission coefficient	216
D.5	B-matrix for multi-fuel system ($\times 10^{-4}$) [330]	216
D.6	Power generation cost coefficient of forty generators	217
D.7	Gaseous pollutant's emission coefficient of forty generators	218
D.8	Power generation cost coefficients of Korean power system	219
D.9	Gaseous pollutant's emission coefficient of Korean system	221
D.10	Prohibited operating zones data for Korean power system's thermal gener- ating units	224

List of Figures

Figure no	Caption	Page
1.1	Local vs global solution	6
1.2	Yearly publication distribution in reputed journals	12
1.3	Hybrid publication distribution in reputed journals	20
1.4	Classifications of Multiobjective solution procedures	22
1.5	Multiobjective evolutionary algorithm classification	27
2.1	Synergic predator prey optimization	59
2.2	Generalized benchmark problems convergence analysis	62
2.3	Convergence characteristics of Test power system-1, case-2	65
2.4	Convergence characteristics comparison test power system-3	67
2.5	Frequency distribution of test power system-3	68
3.1	Membership function behaviour	81
3.2	Graphical representation of surrogate worth trade-off (μ_D) behaviour	82
3.3	Flow chart of CEPPS based solution approach	83
3.4	Generalized benchmark test functions convergence comparison	86
4.1	Flow chart of APPO based solution approach	109
4.2	Test power system 1, case 2, Convergence characteristics comparison	112
4.3	Test power system-1, case-2, Frequency plot of 100 trials	113
4.4	Test power system 1, case 5, Convergence characteristics comparison	113
4.5	Test power system-1, Case 5 Frequency plot of 100 trials	114
4.6	Test power system-2, Pareto front	116
4.7	Test power system-3, Convergence characteristics comparison	118
4.8	Test power system-3, Frequency plot of 30 trials	119
5.1	Natural foraging behavior: <i>fly and walk</i>	125
5.2	Flow chart of FWPPPO based solution approach	129
5.3	Convergence analysis of PPO using generalized benchmark problems	132
5.4	Convergence analysis of FWPO using generalized benchmark problems	132
5.5	Effect of fly velocity limits variation on fitness value of Schwefel 2.26 function	133
5.6	Test system-2, Pareto front using FWPO and PPO	138
5.7	Test system-3, FWPO Pareto front's uniformity analysis	140
5.8	Test system-3, Pareto-frontier by applying FWPO	140
5.9	Performance comparison of FWPO algorithm	142
6.1	Scaling function behavior	149
6.2	Non-interactive solution approach using CDEPS approach	152
6.3	Interactive solution approach using CDEPS approach	154
6.4	Generalized benchmark test functions convergence comparison	155
6.5	Test power system-2 Pareto-front using interactive SWT approach	161
6.6	Test power system-3, Pareto-front using interactive SWT approach	164

List of Figures Continued...

Figure no	Caption	Page
6.7	Convergence behavior, Test power system-3, case-1 and 2	166
6.8	Robustness analysis test, Test power system-3, case-1 and 2	166

List of Abbreviations

Notation	Description
ABC	Artificial Bee Colony
ACO	Ant Colony Optimization
AIS	Artificial Immune System
APPO	Adaptive Predator Prey Optimization
APSO	Anti Predator Swarm Optimization
BA	Bat Algorithm
BBPSO	Barebones Particle Swarm Optimization
BCO	Bee Colony Optimization
BFO	Bacterial Foraging Optimization
BSA-ES	Binary successive approximation base evolutionary search
BSO	Brain Storm Optimization
CCPSO	Composite Particle Swarm Optimization
CDE	Chaotic Differential Evolution
CDEPS	Chaotic Differential Evolution and Powell's method
CEP	Chaotic Evolutionary Programming
CEP	Classical Evolutionary Programming
CEPPS	Chaotic Evolutionary Programming Pattern Search
CQGSO	Continuous Quick Group Search Optimizer
CSO	Cat Swarm Optimization
CSS	Charged System Search
DE	Differential Evolution
DE-SQP	Differential Evolution Sequential Quadratic Programming
DHS	Differential Harmony Search
DM	Decision maker
EA	Evolutionary Algorithm
ELD	Economic Load Dispatch
ELHN	Enhanced Augmented Hopfield Neural Network
EMO	Electromagnetish Optimization
EP	Evolutionary Programming
ES	Evolutionary Strategy
FA	Firefly Algorithm
FEP	Fast Evolutionary Programming
FP	Flower Pollination
FWPO	Fly and Walk Predator Prey Optimization
GA	Genetic Algorithm
GP	Genetic Programming
GSA	Gravitational Search Algorithm
GSO	Group Search Optimizer
HS	Harmony Search
IFEP	Improved Fast Evolutionary Programming
IGA-MU	Improved Genetic Algorithm
IPSO	Improved Particle Swarm Optimization
IWD	Intelligent Water Drop

List of Abbreviations Continued

Notation	Description
IWO	Invasive Weed Optimization
KGMO	Kinetic Gas Molecule Optimizer
KHA	Krill herd optimization
LMP SO	Modulated Particle Swarm Optimization
MABS	Modulated Artificial Bee Colony
MCDM	Multicriterion decision making
MDE	Modified Differential Evolution
MED	Minimum emission dispatch
MFEP	Mean gaussian and cauchy Evolutionary Programming
MODE	Multiobjective Differential Evolution
MOEA	Multiobjective Evolutionary Algorithm
MOLD	Multiobjective load dispatch
MOP	Multiobjective optimization problem
MPSO	Multiobjective Particle Swarm Optimization
NPGA	Nitche Pareto Genetic Algorithm
NSGA	Non-dominated sorting Genetic Algorithm
PDE	Pareto Differential Evolution
PF	Pareto front
POZ	Prohibited Operating Zone
PPO	Predator Prey Optimization
PS	Pareto Set
PSO	Particle Swarm Optimization
QTLBO	Quasi Oppositional Teacher Learner Based Optimization
RCCRO	Real Coded Chemical Reaction Optimization
RCGA	Real Coded Genetic Algorithm
RFD	River Formation Dynamics
RPD	Recognition Primed Model
SA	Simulated Annealing
SBOA	Seeker Based Optimization Algorithm
SFLA	Shuffled Frog Leaping Algorithm
SO	Spiral Optimization
SOOP	Single Objective Optimization Problem
SPEA	Strength Pareto Evolutionary algorithm
SPPO	Synergic Predator Prey Optimization
SQP	Sequential Quadratic Programming
SWT	Surrogate worth trade-off
TLBO	Teacher Learner Based Optimization

List of Symbols

Notation	Description
$A(P)$	augmented objective function
B_{00}, B_{j0} and B_{ij}	transmission loss coefficients
DR_j	down ramp limit of j^{th} generator
F^*	optimal value of function $F(X)$
IT^{max}	maximum number of iterations
M	number of conflicting objectives
N_P	population size
N_g	number of committed generators
N_{zj}	number of prohibited operating zone of j^{th} generator
PP_j	predator's position in j^{th} dimension
PP_j^{max}	upper position bound of predator
PP_j^{min}	lower position bound of predator
P_s	slack generator
P_D	forecasted load demand
P_L	network transmission loss
P_i^B	i^{th} prey's previous best position
P_i^W	i^{th} prey's previous worst position
P_i^t	i^{th} prey's position at t^{th} iteration
P_i^{NB}	i^{th} prey's previous not best position
$P_{j,i}^L$	lower range of i^{th} prohibited zone of j^{th} generator
$P_{j,i}^U$	upper range of i^{th} prohibited zone of j^{th} generator
P_{jk}^{max}	committed generator's upper limit for k^{th} fuel option
P_{jk}^{min}	committed generator's lower limit for k^{th} fuel option
P_j	real power generated by j^{th} generator
P_j^0	previous generated power of j^{th} generator
P_s	power of slack generator
P_s^{max}	upper power generation limit of slack generator
P_s^{min}	lower power generation limit of slack generator
Pv_j	predator's position in j^{th} dimension
Pv_j^{max}	upper velocity bound of predator
Pv_j^{min}	lower velocity bound of predator
UR_j	upper ramp limit of j^{th} generator
X	Search space X
x^*	optimal point x^*
X^*	optimal vector X^*
β	Euclidean distance between prey and predator
ϵ	small number (termination criteria)
η_i	Strategy parameter associated with i^{th} individual
γ	probabilistic fear controlled two state operator
μ	Membership value
μ_A	Aggregation function value
μ_D	surrogate worth trade-off value
μ_c	Satisficing function value

List of Symbols Continued....

Notation	Description
μ_s	scaling function value
a	maximum amplitude of predator effect over prey
$a1_{jk}, \dots, e1_{jk}$	cost coefficients
$a2_{jk}, \dots, e2_{jk}$	emission coefficient
b	maximum distance of predator effect over prey
$c_{11}, c_{12}, c_{12} \& c_{22}$	acceleration coefficients
f_k^{max}	maxima of k^{th} objective function
f_k^{min}	minima of k^{th} objective function
g^B	global best position
g^W	global worst position
g^{NB}	global not best position
n	number of variables
p	is number of equality constraints
q	number of inequality constraints
r_k	penalty factor
v_i^t	i^{th} prey's velocity at t^{th} iteration
v_j^{max}	maximum velocity limit of j^{th} particle
v_j^{min}	minimum velocity limit of j^{th} particle
w	inertia weight
w_k	k^{th} normalized weight
x_i^{max}	upper bound on decision variable
x_i^{min}	lower bound on decision variable
x_i	i^{th} decision variable x
z_i^*	Ideal point
z_i^{nad}	Nadir point

CHAPTER 1

INTRODUCTION

Optimization is a procedure to search a best, feasible solution with respect to an objective. The objective can be cost of operation and amount of harmful gases, while meeting a set of system constraints. For decades, the optimization problems either unconstrained or constrained are frequently encountered in various fields of economics, management, science and engineering. Therefore, many engineers and researchers devote to this field rigorously and have proposed plenty of prominent algorithms. These techniques are classified as: directed search methods, gradient methods and random search techniques. However, it is observed that there does not exist a universal solution procedure to solve all kinds of problems [78]. The system constraints as well as resource limitations such as calculation complexity or memory requirements and/ or execution time requirements further enhance the complexity.

The real life problems are often associated with multiple objectives. In complex engineering optimization problems there often exists several non-commensurable criteria which must be considered. The situation is formulated as a multi-criterion or multiobjective optimization problem in which the goal of the engineer is to maximize or minimize not a single objective function but several objective functions simultaneously. The purpose of multiobjective problems in the mathematical programming framework is to optimize the different objective functions simultaneously. Owing to the conflicting nature of objectives, an optimal solution that simultaneously maximizes/ minimizes all the criteria is not attainable. Instead, there are several solutions, called efficient solutions that have the property, that no improvement in any objective is possible without sacrificing one or more

of other objectives. Such possible solutions are known as non-inferior or Pareto optimal solutions. Many real-world optimization applications involve the optimization of multiple non-commensurable and often competing objectives simultaneously. The solutions procedure of these multiobjective optimization problems is a two step process and can be classified as [147]:

- Priori preference articulation where multiple objectives are aggregated into a scalar function with adequate weights. It transforms the multiobjective problem into a single-objective problem prior to optimization.
- Progressive preference articulation, i.e., decision and optimization are intertwined where partial preference information is provided, upon which the optimization occurs.
- Posteriori preference articulation where a family of trade-off solutions is found before a decision is made to determine the best solution.

The set of all the Pareto optimal points is called the *Pareto set* (PS) and the set of all the Pareto optimal objective vectors is the *Pareto front* (PF). In many real-life applications of multi objective optimization, an approximation to the PF is required by a decision maker for selecting a final preferred solution. Most multiobjective optimization problems may have many or even infinite Pareto optimal vectors. It is very time-consuming, if not impossible, to obtain the complete Pareto front. On the other hand, the decision maker may not be interested in having an unduly large number of Pareto optimal vectors due to memory and execution time constraints. Therefore, many multiobjective optimization algorithms opt a manageable number of Pareto optimal vectors that are best representatives of the entire PF. Some researchers have also made an effort to approximate the Pareto front by using a mathematical model [469].

Among the complex engineering optimization problems, from the power system engineering domain, a crucial issue is optimal use of available resources to generate economical power. The heavy dependency on electricity in daily life has resulted in concerns of planning operation and control of the interconnected power system. Among these, an important problem is the economic operation of power system, which means that at

every step in the planning, scheduling and operation of generators, the system must be optimal, leading to absolute economy. The transmission losses also play an important role in the economic operation of the power system. The historical background of optimal planning of electric power resources is reported as far back as the early 1920's, where the concern was economical allocation of generation to available power generators [13]. Today's *Economic Load Dispatch* (ELD) is an important optimization task in power system operation, which aims to allocate generation among the committed units, to divide the power economically, while satisfying various system constraints. Practical, ELD problem has complex and nonlinear characteristics with many equality and inequality constraints. In practice, because of physical operation limitations such as faults in the machines themselves or the associated auxiliaries, such as boiler, feed pumps *etc.*, units can have prohibited operating regions that must be avoided in practical applications. Further, due to acceleration restrictions of generating units, the practical power generators have ramp rate limits constraints. In addition, the generating units may have multi-valve steam turbines with number of steam admission valves that are opened sequentially to obtain ever-increasing output of the units and the valve point effects produce a ripple-like heat rate curve. Moreover, the generating units that have multi-fuel source, lead to the problem of determining the most economic fuel utilization [163].

Since, the area of interest for optimizing the power generation system is expanded variety of other objectives like security, reliability and environmental effects are to be considered along with economical objectives [26]. These considerations can be of conflicting nature and requires decision making for the optimal operation of the power system. In a large number of real life problems also, decision making is a problem while handling multiple goals. The achievement of these goals is expressed in terms of quantitative performance criteria, some of which can be selected as the optimization objectives. The situation is formulated as *multiobjective optimization problem* (MOP) in which the goal is to maximize/ minimize several objectives at the same time.

Many approaches and objectives are proposed for the solution of MOP [92]. These methods are categorized in two groups: Non-interactive and interactive methods [58]. In non-interactive methods a global preference function of the objectives is identified and

optimized with respect to the constraints. In the interactive methods, a local preference function or trade-off among objectives is identified by interacting with the decision maker and the solution process gradually moves towards the globally satisfactory solution. Mostly, interactive methods are used to find non-inferior solutions. Qualitatively a non-inferior solution of a MOP is one where any improvement of one objective function can be achieved only at the expense of another. The most widely used methods of generating such non-inferior solutions are weighting method. Solution methodologies for solving the interactive MOP differ in two major ways [93]: The procedure used to generate non-inferior solutions, The ways used to interact with the *Decision maker* (DM) and the type of information made available to the DM. In almost all decision making problems, there are several criteria for judging the possible alternatives. The main concern of DM is to fulfill conflicting goals while satisfying the constraints of the optimization problem formulated for the given system.

1.1 OPTIMIZATION PROBLEM

The optimization problem aims to find solutions which are optimal or near-optimal with respect to desired goals. Usually, such kind of problems cannot be solved in one step, but a solution process is to be followed. Often, the solution process involves hierarchical steps, namely: recognizing and defining problems, constructing and solving models, and evaluating and implementing solutions. Combinatorial optimization problems are concerned with the efficient allocation of limited resources to meet desired objectives. The decision variables take values from bounded, discrete sets and additional constraints on basic resources. Usually, a problem has many possible alternatives and the predetermined goal/objective determines a best alternative as a solution. The situation is different for continuous optimization problems, which are concerned with the optimal setting of parameters or continuous decision variables. Here, no limited number of alternatives exist, but optimal values for continuous variables have to be determined. In real world problems there is involvement of more than one decision variables and at least one objective function.

1.1.1 Single Objective Optimization Problem

The optimization problem with more than one variable and a single objective function is termed as scalar optimization problem [462]. Considering a decision vector $X = [x_1, x_2, \dots, x_n]$, a single objective problem is defined as:

Mathematically,

$$\text{Minimize/Maximize} \quad f(X) \quad (1.1)$$

subject to:

$$\text{Inequality constraints} \quad g_j(X) \leq 0 \quad (j = 1, 2, \dots, p) \quad (1.2)$$

$$\text{Equality constraints} \quad h_k(X) = 0 \quad (k = 1, 2, \dots, q) \quad (1.3)$$

$$\text{Decision boundaries} \quad x_i^{\min} \leq x_i \leq x_i^{\max} \quad (i = 1, 2, \dots, n) \quad (1.4)$$

where n is the number of decision variables, p is number of inequality constraints and q is number of equality constraints. x_i^{\min} is the lower bound on decision variable, x_i^{\max} is the upper bound on decision variable .

It is desired to obtain a solution X^* such that the above stated objective is either maximized or minimized in such a way that the stated constraints are satisfied. The associated common definitions are as follows:

- Feasible solution: It is a decision vector X that satisfies all the constraints imposed by the problem on it. A set of all feasible solutions defines the feasible region of the problem.
- Local solutions: A local solution is a solution of problem, that there does not exist any solution in its neighborhood, with better objective function value. However, it may not be a best solution of the problem.
- Global solutions: It is a solution of the problem, such that there exists no other feasible solution having better objective function value.

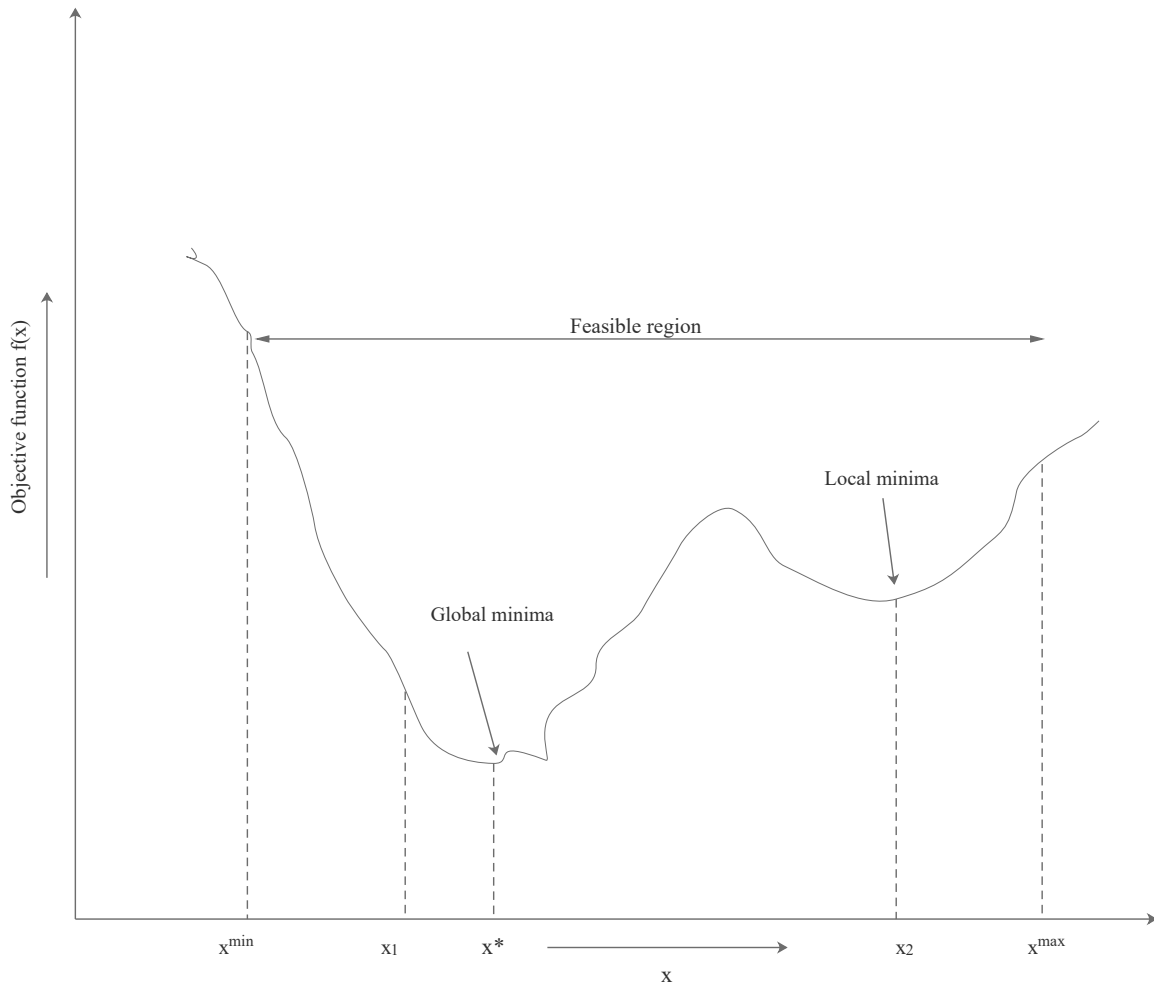


Figure 1.1: Local vs global solution

- **Continuous function:** A function $f(x)$ is said to be continuous at a point x_1 in its domain, if the limit of $f(x)$ as x approaches c through the domain of $f(x)$ exists and is equal to $f(c)$. Mathematically,

$$\lim_{x \rightarrow c} f(x) = f(c) \quad (1.5)$$

- **Unimodal function:** The function $f(x)$ is said to be unimodal on $[x^{min}, x^{max}]$ if for any two points x_1 and x_2 , such that $x^{min} < x_1 < x_2 < x^{max}$,

$$f(x_1) \geq f(x^*) \leq f(x_2) \quad (1.6)$$

- **Multi-modal function:** A multi-modal function has multiple local minima, in its search space, but there is only one global minima.

1.1.2 Multiobjective Optimization Problem

The MOP always have more than one non-commensurable objective to be minimized or maximized simultaneously [461]. These problems are also called as *multi criterion decision making (MCDM)* problem. The solution of the MOP problem must satisfy Vilfredo Pareto's concept of Pareto efficiency, which states that "It is impossible to make any one individual better off without making at least one individual worse off". Therefore, the solution procedures often results in a set of solutions, instead of a single solution. Mathematically, a MOP with M objectives is defined as:

Minimize

$$F(X) = [f_1(X), f_2(X), \dots, f_M(X)]^T \quad (1.7)$$

subject to:

Inequality constraints

$$g_j(X) \leq 0 \quad (j = 1, 2, \dots, p) \quad (1.8)$$

Equality constraints

$$h_k(X) = 0 \quad (k = 1, 2, \dots, q) \quad (1.9)$$

Decision boundaries

$$x_i^{\min} \leq x_i \leq x_i^{\max} \quad (i = 1, 2, \dots, n) \quad (1.10)$$

The multiobjective programming is an act of multi facet decision making in a problem, where more than one conflicting objective are to be satisfied simultaneously. The terminology of significance related to MOP is as follows:

- Pareto-optimality: A point, $x^* \in X$, is Pareto-optimal iff there does not exist another point, $x \in X$, such that $F(x) \prec F(X^*)$ for at least one objective function.

$$\{x \in X | \exists x^* \in X : \forall i f_i(x^*) \leq f_i(x)\}$$

- Weak Pareto Optimality: A solution $x^* \in X$ is a weak Pareto optimal if there does not exist another solution $x_i \in X$, such that $F(x) < F(x^*)$.

$$\{x^* \in X | \nexists x \in X : \forall i f_i(x) < f_i(x^*)\}$$

- Pareto optimal set: The Pareto optimal set is P^* is defined as:

$$P^* = \{x \in X | \exists y \in X : \forall i f_i(y) \preceq f_i(x)\}$$

- Pareto front: For a Pareto Optimal set FP^* , the Pareto front is defined as:

$$FP^* = \{F(x) = (f_1(x), \dots, f_k(x)) | x \in P^*\}$$

- Ideal point: It represents the lower bounds of the Pareto front and it is represented as z_i^* such that

$$z_i^* = \{z \in Z | \forall_i \min(z_i)\}$$

- Nadir point: It represents the upper bound of Pareto front and it is represented as z_i^{nad} such that

$$z_i^{nad} = \{z \in Z | \forall_i \max(z_i)\}$$

- Decision maker: As stated above, in MOP there are always multiple Pareto optimal solutions, but from an implementation point of view only one optimal solution is to be selected. This aspect of MOP requires a *decision maker* (DM) that can provide preference information to choose the best solution in a particular instance [282]. It is incorporated with the help of a Preference function, which can balance conflicting objectives, by favorably finding trade-off between objectives.

- Preferences: It refers to a decision maker's opinion in concern to solutions in the objective space.

- Utility function: In terms of multiobjective optimization, an individual utility function is defined for each objective function and represents the relative importance of the objective.

- Global criterion: A global criterion is a scalar function that mathematically combines multiple objective functions, it does not necessarily involve utility or preferences [147].
- Compromised solution: It is an alternative to the idea of Pareto optimality and efficiency, which yields a single point solution. It entails minimizing the difference between the potential optimal point and utopia point [476]
- Utopia point: A point $F^* \in z^k$ is a utopia point [147], *iff*

$$F_i^* = \min(x) F_i(x) | x \in X; \forall i$$

1.2 OPTIMIZATION TECHNIQUES

Nowadays, optimization is an indispensable part of decision making in relation to design, planning and operation problems. The optimization always involves certain decision variables, problem constraints and objective functions. The objective function for a problem is selected as per expectations from the system. Optimization is an act to select a set of decision variables, such that the desired objectives are either minimized or maximized, subject to system constraints. The selected set of decision variable is called an optimal solution of the problem. The procedure adopted to obtain an optimal solution is called optimization algorithm. An optimization algorithm proceeds iteratively to satisfy the objectives, until termination criteria is satisfied. The optimization problems are classified in several ways:

1.2.1 Conventional Techniques

Conventional optimization methods are also addressed as deterministic optimization techniques. In this category, the solution procedures are classified as: gradient techniques, direct search methods and random techniques. The pros and cons of gradient techniques and directed search methods are summarized as follows:

Gradient methods: These methods require 1st and 2nd order derivative of function under investigation. Since, the objective function in most of the engineering design problems is not differentiable, these methods cannot be applied directly. Hence, the gradient

based search methods cannot be applied to the problems where the objective function is discontinuous. On the contrary, in problems where the derivative information of the objective function is easily available, gradient search methods are very efficient. Most gradient based search methods work by searching along several search directions iteratively. The algorithms vary according to the way the search directions are defined. The most generally used gradient based optimization methods include Newton's method [7], Marquardt's method, conjugate gradient method [201] and steepest descent method [5]. The convergence behavior of these algorithms depends upon the initially selected starting point and the nature of the objective function.

Directed search methods: These methods possess excellent simplicity, flexibility and reliability. The models are useful when no analytical solution is available for an optimization problem, because there is always a possibility that the numerical solution may get stuck in local optima. The directed search methods explore the search area, considering a single variable at a time, hence these methods require a large number of function evaluations to find a solution. Moreover, the algorithm may prematurely converge to an unsatisfactory solution, especially in the case of functions with highly non-linear interactions among variables, lacks finite termination on quadratic objectives [97]. In few words, a carefully selected directed search method is an effective tool for solving many nonlinear optimization problems.

1.2.2 Random Search Techniques

A random search technique is an algorithm that uses some kind of randomness or probability in its operational mechanism. These may be called as a Monte Carlo method or a stochastic algorithm. The term meta-heuristic is also commonly associated with a random search algorithms [300]. Many nature inspired algorithms such as *Genetic Algorithm* (GA), *Evolutionary Programming* (EP), *Evolutionary Strategy* (ES), *Particle swarm optimization* (PSO) and *Genetic Programming* (GP), *differential evolution* (DE) have already gained wide acceptance in solving optimization problems [346]. The era of random techniques (population based techniques) started in 1989, with the introduction of GA followed by EP and ES. Since then, the repository of random search techniques has

Table 1.1: Prominent evolutionary algorithms

Algorithm	Year	Principle	References
Simulated annealing (SA)	1983	Thermodynamic effects	[18]
Genetic algorithm (GA)	1989	Survival of fittest	[23]
Evolutionary programming (EP)	1993	Survival of fittest	[38]
Particle swarm optimization (PSO)	1995	Bird foraging behavior	[54]
Ant colony optimization (ACO)	1996	Ant behavior	[62]
Differential evolution (DE)	1997	Survival of fittest	[471]
Harmony search (HS)	2001	Music improvisation	[101]
Electromagnetism optimization (EMO)	2003	Attraction repulsion mechanism	[123]
Shuffled frog leaping (SFLA)	2004	Frog movement	[196]
Artificial immune systems (AIS)	2005	Immune system	[36]
Bee colony optimization (BCO)	2006	Bee swarm	[187]
Cat swarm optimization (CSO)	2006	Cat behavior	[192]
Artificial bee colony (ABC)	2007	Foraging activity	[223]
Firefly (FA)	2007	Flashing light pattern	[479]
Biogeography-Based Optimization (BBPSO)	2008	Biogeography	[263]
Gravitational search (GSA)	2009	Law of gravity	[292]
Bat algorithm (BA)	2010	Echolocation	[480]
Brain Storm optimization (BSO)	2011	Human brain storming	[350]
Flower pollination (FP)	2011	Pollination process	[478]
Bacterial foraging (BFO)	2012	Foraging activity	[365]
Krill herd (KHA)	2012	Herd foraging activity	[357]
Invasive weed optimization (IWO)	2014	Biological invasion	[477]

been increasing continuously till date with the introduction of state of the art concepts. A majority of contributions are nature inspired/ physical or chemical phenomenon based. The information related to these techniques in the past decade is summarized in Table: 1.1.

In general, the nature remains the main source of inspiration for the research community, as a result the majority of the recently proposed algorithms are nature inspired. Further, the nature-inspired algorithms are classified as bio-inspired and swarm intelligence based algorithms [378]. The swarm intelligence based methodology mimics the collective behavior mechanism of insects like ants, bees *etc.* or other social groups like bird flocks, fish schools *etc.* The foraging behavior to locate the better foraging areas, based on cognitive and social experience is the underneath working concept. The famous algorithms in this class are PSO, *Ant Colony Optimization (ACO)*, *Bacterial Foraging Activity (BFO)*, *Krill Herd Algorithm (KHA)*. The bio-inspired algorithms, follows the fundamental aspects of the neo-Darwinian evolutionary process [113]. These stochastic approaches follow the concepts of natural evolution. These methods simultaneously search with a population of candidate solutions and associate an objective score as a fitness value

for each one. The algorithms then select among the population to favor those solutions that are more fit. The next generation (new population/ offspring's) consists of replicates of the better fitness solutions that have been genetically mutated and crossed over in a biological metaphor. The decision variables are perturbed in such a way that they inherit characters of their parents, as well as change in random ways. The EP, ES and GA are famous techniques in this category that have dominated in the previous decade. The physical or chemical phenomenon base algorithm involves certain physical or chemical laws such as electrical charges, gravitational laws, musical concepts *etc.* [378]. In short, a variety of new population based techniques are proposed as a successful optimization solution procedure. A few techniques, which have attracted sight of the research community are tabulated in Table 1.1. The objective of these techniques is to provide a better quality solution procedure, with faster convergence rate and lesser resource utilization. Among these techniques the EP, PSO and DE have dominated in the past decade. The yearly publication data of duration 2000-2016, is presented in Fig. 1.2

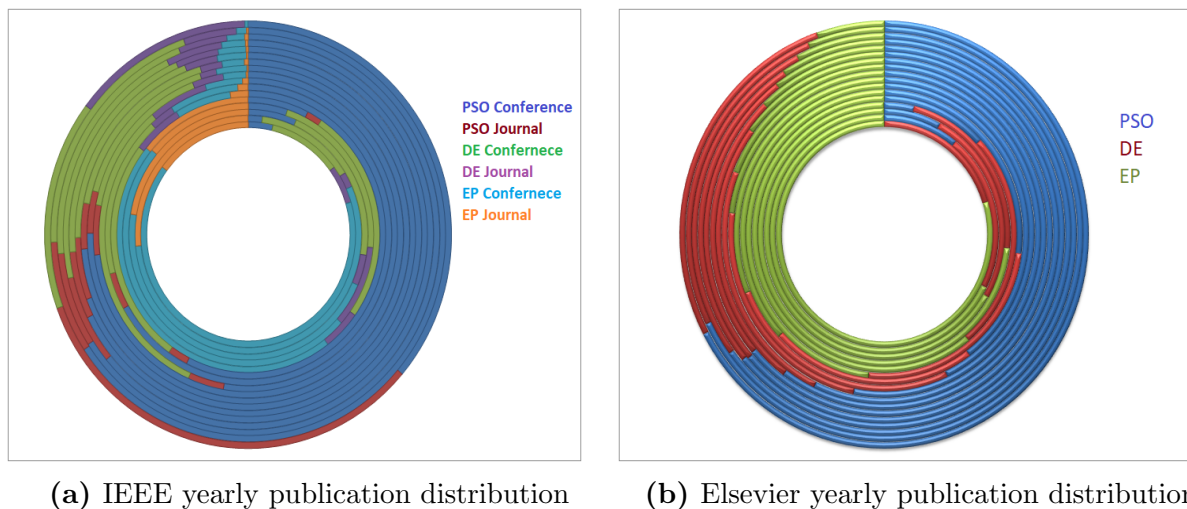


Figure 1.2: Yearly publication distribution in reputed journals

In 1965, the EP was initially proposed as an approach to artificial intelligence [465] and later on successfully applied to many real world optimization problems. The mutation and selection operators are the two main constituents of EP. The mutation operator is responsible for generation of new solutions (offspring), whereas selection operation is employed to decide parents for the next generation. The EP solution procedure is presented in algorithm. 1.1:

Algorithm 1.1 Evolutionary programming algorithm

Initialize individual's population

- 1: **while** stopping criteria unmet **do**
 - 2: Evaluate fitness score of each individual based on the objective function
 - 3: Create offspring of each individual
 - 4: Evaluate fitness score of each offspring based on the objective function
 - 5: Conduct pairwise comparison over the union of parents and offspring
 - 6: Select better individuals for the next generation
 - 7: **end while**
-

The main disadvantage of the EP is that it is slow to converge to a good near-optimum in solving some of the multi-modal optimization problems [68]. As mutation is a key search operator that generates new offspring of the current parents. A new mutation operator based on Cauchy random numbers is proposed to outperforms the classical EP [91]. Further, to improve the performance of the EP the heuristic methods are suggested for initialization of population [64]. In addition, the EP is integrated with fuzzy theory resulting in algorithms such as fuzzy controlled EP [73], fuzzy EP [124], fuzzy adaptive EP [136] *etc.*

The applications of EP involve mobile robot path planning [412], optimal power flow [418], optimal allocation of flexible AC transmission systems [420], recurrent neural network design [45], flexible job shop scheduling [419], fuzzy rule base designing [95] *etc.* In the field of electrical power system EP is widely applied to various problems. It is used to solve to *economic dispatch* (ELD) problem [181] *etc.* An EP [72] and its hybrid [65] based approach has been suggested to handle constrained optimization problems. Further, the EP is extended with an interactive fuzzy satisfying approach to solve the multiobjective scheduling problem [139]. The EP has been used without non domination sorting procedure to solve the multiobjective problem [291]. A multiobjective EP and its applications to power generation expansion planning has been presented [285]. Multiobjective EP has been used to solve combined economic emission load dispatch problem [265]. Recently, in 2016 the EP has been successfully used in diverse applications such as optimal reactive power dispatch [450], impedance spectroscopy analysis [440], optical network problem [455], partial inverse modeling [454], wheeling problem [108] *etc.*

In 1995, the PSO algorithm was initially introduced by Kennedy and Eberhart [54].

The basic idea behind PSO is to simulate a swarm of bees, fishes, or birds looking for food. Apart from searching for their targets independently and stochastically, these species also exchange information with each other so that all of them can eventually move toward the target from different directions. Since the foraging paths are different for each particle, a wide area of design space is explored and this increases the chance of finding the global optimum solution. The PSO is developed through simulation of a simplified social system, and has been found to be robust in solving nonlinear optimization problems. The PSO generates high-quality solutions quickly and has stable convergence characteristic than other stochastic methods [126]. In addition, the PSO algorithm in comparison to other evolutionary algorithms can easily be programmed and is independent of encoding processes. In PSO, a number of the particles are placed in the search space of problem or function, and each evaluates the objective function at its current location. Each particle, then determines its movement through the search space by combining some aspect of the history of its own current and best-fitness locations with those of one or more members of the swarm, with some random perturbations. The next iteration takes place after all particles have been moved. Eventually the swarm as a whole, like a flock of birds collectively foraging for food, is likely to move close to an optimum of the fitness function. The functioning of the PSO search procedure is explained with the help of Algorithm 1.2:

Algorithm 1.2 Particle swarm optimization algorithm

Initialize particle's population array

- 1: **while** stopping criteria unmet **do**
 - 2: Evaluate fitness score of each particle based on the objective function
 - 3: Update particle's individual experience on basis of fitness value
 - 4: Update global best experience of swarm
 - 5: Update velocity and position of particles
 - 6: **end while**
-

However, the particle's velocity update is parameter dependent that can affect the result significantly. Improper setting of these parameters might decrease the search efficiency or even result in divergence of the search process [100]. Although the PSO algorithm has a coordinated searching scheme, this does not give it an apparent advantage over other evolutionary algorithms in finding the global optimum solution, especially for those highly-noised multi-modal problems. PSO behavior has been enormously analyzed

for its parameter dependency, parameter selection, stability *etc.* [303, 210]. A standard has been designed to be a straightforward extension of the original PSO algorithm while taking into account more recent developments that can be expected to improve performance on standard measures [215]. In order to improve the performance of PSO various modifications have been suggested such as the introduction of an inertia weight [85], constriction factor [111], fully informed particle swarm [148], static topologies [148], dynamic topologies [168]. Moreover, PSO variants such as binary PSO [71], adaptive PSO [256], regulated discrete PSO [107], angle modulation PSO [175] have been presented. A survey on parameter selection for PSO has been presented [382]. Further, PSO is hybridized with mutation operator [79], evolutionary methods [117], breeding [106], Cauchy mutation [119], ACO [140], DE [103], GP [178] *etc.* In addition, a velocity independent PSO model called *bare bones PSO* (BBPSO) [128], BBPSO with jumps [283] have been proposed. A wide variety of survey papers related to PSO's overview, methods, analysis, taxonomy such as [293] *etc.* have appeared in literature, that present better view of PSO conception.

On the basis of literature, the PSO has been applied in a wide variety of applications. An important field among PSO's applications is power system optimization. Valle *et al.* [245] has discussed in detail the basic concepts, variants and applications of PSO in power systems. A detailed study of PSO variants is highlighted and an approach called Frankstien's PSO is proposed and its advantages are discussed [276]. The other related compiled literary studies are [296]. Recently, in 2016, the PSO applications in the power system field include economic dispatch incorporating solar energy [428], FACTS setting [433], photo-voltaic multiple peaks power tracking [451], optimal power management of plug-in hybrid electric vehicles [436], maximum power point tracking in PV system [429] *etc.*

Differential evolution optimization algorithm was proposed by Storn and Price in 1995 [474]. It has a faster convergence behavior, ability to find the true global minimum, regardless of the initial system parameter values and less number of control parameters [75]. A detailed revision of basic concepts of DE, its variants and applications to constrained multiobjective problems has been discussed by Das and Suganthan [334]. The basic strategy employs the difference of two randomly selected parameter vectors as the source of

random variations for a third parameter vector. The DE's iterative search process is explained in Algorithm 1.3.

Algorithm 1.3 Differential evolution algorithm

Initialize individual's population array randomly

- 1: **while** stopping criteria unmet **do**
 - 2: Evaluate fitness score of each particle based on the objective function
 - 3: Select three individual's randomly
 - 4: Perform mutation operation to generate mutant
 - 5: Perform recombination operation to generate trial vector
 - 6: Select an offspring for the next generation out of trial and target vector
 - 7: **end while**
-

Storn and Price [75] suggested a total of ten different working strategies for DE. These strategies were derived from the five different DE mutation schemes and each mutation strategy was combined with either the exponential/binomial crossover. This yielded a total of ten DE strategies. However, similar to other EAs, DE algorithm has also experienced some challenging problems. Despite, fast and reasonable solution, sometimes it does not guarantee a global optimal solution. The improved DE has also faster convergence, but often it has probability of searching towards a local optimum or getting premature convergence. This drawback can be overcome by a larger population size, but it results in excessive memory requirements and large computational time [340]. Further, if the size of power system problem is large, DE may face problem of dimensionality [344]. The DE algorithm stagnates at local optimal even if the population is diverse [279]. The success of DE in solving a specific problem crucially depends on appropriately choosing trial vector generation strategies and their associated control parameter values. Employing a trial-and-error scheme to search for the most suitable strategy and its associated parameter settings requires high computational costs. Moreover, at different stages of evolution, different strategies coupled with different parameter settings may be required in order to achieve the best performance [290].

In order to overcome deficiencies DE with adaptive strategy [290], fuzzy adaptive DE [173], trigonometric mutation DE [125], DE with random localization [199], greedy random search based DE [162], improved self adaptive DE [290], self adaptive DE [174], self adaptive DE with control parameter settings [189], opposition-based DE [260], DE

with global and local neighborhoods [275], DE with new mutation strategy [302]. Das et al. [275] proposed two kinds of topological neighborhood models for DE in order to achieve better balance between its explorative and exploitative tendencies. Hybridization with PSO [138], ACO [228], BFO [214], and SA [339] has been proposed for performance improvement. He and Han proposes a hybrid binary DE based on *Artificial Immune System* (AIS) for tackling discrete optimization problems [220]. In addition, the attraction-repulsion concept of electromagnetism [222] to boost the mutation operation and crossover based adaptive local search [222], has been proposed to improve the performance of the classical DE. A hybridization of DE with the neighborhood search [235] and DE based mimetic approach [287] has also been proposed.

In terms of optimizing high-dimensional problems, cooperative co-evolution [49] with divide-and-conquer strategy has been proved an effective choice with DE. The involved steps are: 1) Problem decomposition: splitting the object vectors into some smaller sub-components. 2) Optimize subcomponents: evolve each subcomponent with a certain optimizer separately. 3) Cooperative combination: combine all subcomponents to form the whole system. In order to improve performance of DE for multi-modal problems, the niching techniques include crowding distance [154], neighborhood mutation [366], fitness sharing [266], restricted tournament selection [324] and speciation [171] have been proposed.

The survey papers on conceptual and performance details related to DE includes [252, 318, 225, 267]. A comparative studies related to DE include discussion on selection schemes [60], self adapting control parameters [189] *etc.* are available in the literature for curious readers.

The DE has been successfully applied to a wide variety of problems. Hybrid DE algorithm has been to solve the permutation flow-shop scheduling problem with the make-span criterion [259], scheduling problems [204], no-wait flow-shop scheduling problem [231]. Similarly, a self-adaptive differential evolution has been used for two-stage assembly scheduling problem to minimize maximum lateness with setup times [212].

In the field of power system the DE is used for optimal power scheduling [261], transient stability [242], optimal power flow [311], environmental/ economic power [328], optimal re-

active power dispatch [338], power flow with FACTS [240], emission constrained economic dispatch [310], power system security [349], large capacitor placement problem [141], automatic generation control of inter connected system [383], maximum power point tracking of PV cell [326], load frequency control of multi-source system [399] *etc.* In the current year, DE has been applied to estimate maximum power delivered by PV cells [439], optimal reactive power flow [432], combined heat and power dispatch [444], multiobjective reactive power dispatch [431], expansion planning with solar power plant [452] and wind plant [453], optimizing power consumption in wireless networks [441], economic and emission diaptch [428], squirrel cage induction generator design [446], contingency management [448], modeling of PV module [437], design of proportional integral controller for fuel cell power plant [427], power electronic circuit management [447] *etc.*

A wide variety of procedures are developed, that are applied to solve problems of diverse fields. In the past decade, as per publications account in some of the popular scientific database the major dominant methods are EP, PSO and DE. However, from the problem solving perspective none of the algorithm can solve all the problems. As per literature, there are several types of problems where a simple EA fails to obtain an acceptable solution [182]. Hybridization is the key to obtain a universal solution procedure to solve such kind of practical problems.

1.2.3 Hybrid Search Techniques

In practical optimization problems, a general purpose algorithm may fail to deliver adequate performance. In order to improve the performance of standard algorithm hybridization is the possible alternative [460]. It means that although, an appropriate standard algorithm may result in better performance, however much greater gain is achieved by combining it with specific heuristics or operators that incorporate domain knowledge [115]. In case of many difficult problems, when well studied, the best results are obtained by hybrid approaches where specific heuristics are combined with well proven strategies. The different methods when combined together in a synergistic manner with the incorporation of domain knowledge [305] greatly enhances the problem-solving capability of the plagiarized algorithm. Hybridization methodology merges global search algorithms with a

refinement procedure. Furthermore, hybrid algorithms emphasize on the complementary advantage of population-based search (exploration) and their refinement (exploitative). The exploration provides a consistent estimate of the global optimum, while the exploitative part concentrate the search effort around the best solution. The combined method extracts the virtues of the parent approaches and materialize efficient solution approach [202]. The algorithms are hybridized to achieve the following objectives [134]:

- Improve the performance of EA.
- Improve the quality of solution by EA.
- Incorporate the EA as part of a larger system.

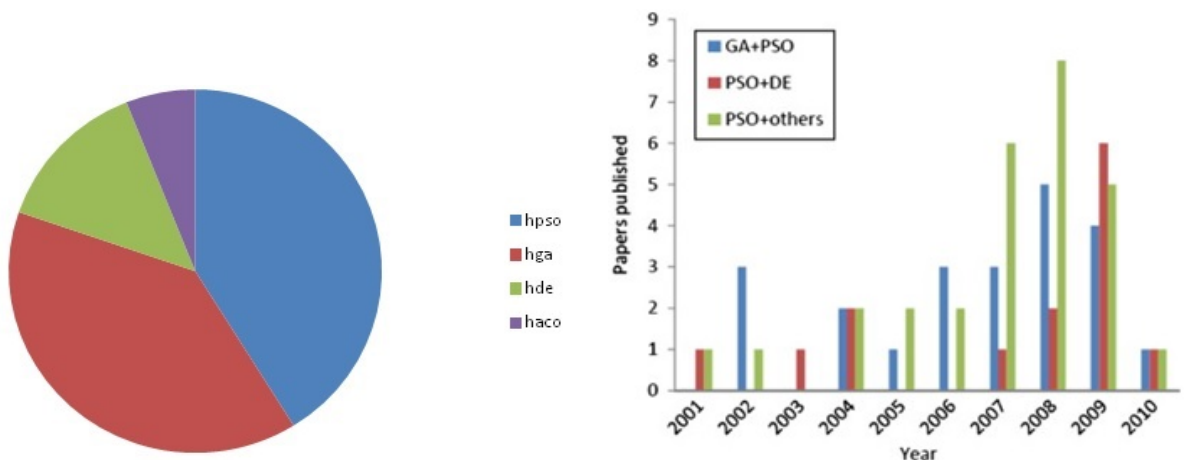
In the past decade, population based hybrid algorithms are in eye of research community. To achieve the above stated objectives, the first hybrid article appeared in *IEEE Xplore* in 1990 [457] and in science direct in 1992 [456]. The architectures of hybrid EA's are classified as follows:

- Hybridization between two EAs.
- Neural network assisted EAs.
- Fuzzy logic guided EAs.
- EA hybridized with other heuristics (local search, hill climbing, greedy search procedure *etc.*)

The refinement procedures are incorporated at different stage of the population based search. On this basis, the hybrid algorithms are classified as before/after/interleaved procedure. Further, the mode of inheritance is another way to classify the hybrid algorithms. The two common modes of inheritance are namely Lamarckian and Baldwinian inheritances. Recently, adaptive hybrids have emerged in the field of optimization. Adaption of parameters and operators is one of the important and promising area of research. Recently, manifold examples of hybrid algorithms have appeared in literature and is following an uptrend.

In order to improve the performance of EP, it is hybridized with GA [89], SA [226], dynamic mutation [169], PSO [152], tabu search [226], SQP [110], neurofuzzy [373], GA-PSO [385], quantum theory [389], DE [308], clustering algorithm [165], tabu search [150], q-Gaussian distribution [264], learning mutation operator [248], fuzzy mutated operator [258], natural immune system [251], Self adaptive EP [144], chaotic mutation operator [241] *etc.* The application of hybrid EP in the field of power systems includes optimal reactive power dispatch [158], dynamic economic dispatch [110], optimal power flow [203, 308], reactive power planning [143], reactive power optimization [164], power system stabilizer [272], combined heat and power [120], security constrained optimal power flow [153], short term hydro thermal scheduling [250], economical output of fuel cells [166], optimal power flow with POZ constraints [363], unit commitment problem [226], non convex economic load dispatch [152], multiobjective reactive power optimization [169], multi-area unit commitment [295] *etc.*

The last decades, pie-chart distribution of major hybrid algorithms, (Fig. 1.3b) exhibit hybrid PSO as a main predominant. The first article related to PSO hybrid appeared in IEEE Xplore and science direct in 2000 and 2005 respectively. Since 2000, an enormous contribution of PSO hybrids to solve miscellanea problems is observed in various journals. The yearly distribution (Fig. 1.3) also exhibit PSO hybrids as a major contributory.



(a) Last decade's publication of major hybrid algorithm in IEEE journals
 Note:hpso-hybrid PSO, hga hybrid genetic algorithm, hde hybrid differential evolution, haco hybrid ant colony

(b) Last decade's major hybrid algorithms (courtesy [352])

Figure 1.3: Hybrid publication distribution in reputed journals

In the past decade, a wide variety of PSO hybrids have been proposed. The conventional mathematical solution approach (local search techniques) has been fused with the PSO [170] with an objective of optimality of search procedure as well as the solution. A concept of Gaussian mutation has been used by adding a small perturbation to a randomly selected velocity vector of random particles [127]. A *sequential quadratic programming* (SQP) has been used to fine tune the solutions obtained from PSO [339]. A survey on hybrids of PSO and ACO is compiled by Hendtlass *et al.* [102]. To improve the performance of PSO, it has been hybridized with ACO [333], EP [257], SQP [156, 377], mutation operators [314], fuzzy theory [319], Nelder-mead simplex method [301], linear interior method [164], tabu search [167], gradient descent method [289], SA [304], AIS [160], random walk strategy [244] *etc.* A blend of binary PSO and real PSO has been used for power scheduling problem [209]. Banks *et al.* have compiled a review on PSO hybridization, combinatorial, multi-criteria and constrained optimization, and applications [239].

The hybrid PSO has been used to solve problem of distribution estimation [132], loss minimization [167], unit commitment [209], optimal power flow [213], short term hydro power generation [249], real time energy management [323], optimal power system stabilizer design [372] *etc.*

The objective being optimality of DE's performance, DE is hybridized with other EAs and local search methods. The case studies include hybridization with SQP [377], PSO [138], GA [247], BBO [305], krill herd [404], adaptive local search [254], self tuning hybrid DE [189], self adaptive DE [174], tune free DE [299], taguchi DE [390], interior point assisted DE [337], scatter search [217], quadratic programming [194] *etc.*

Hybrid DE is successfully applied to solve problem of network reconfiguration of distribution systems [274], optimal control [80], ELD [347], engineering design [313], large scale economic dispatch [337], dynamic economic dispatch [345], hydrothermal scheduling [250], capacitor placement in large distribution system [191], ELD with multiple fuel [238], economic emission load dispatch [367], optimal reactive power [381], non convex economic dispatch [386] *etc.*

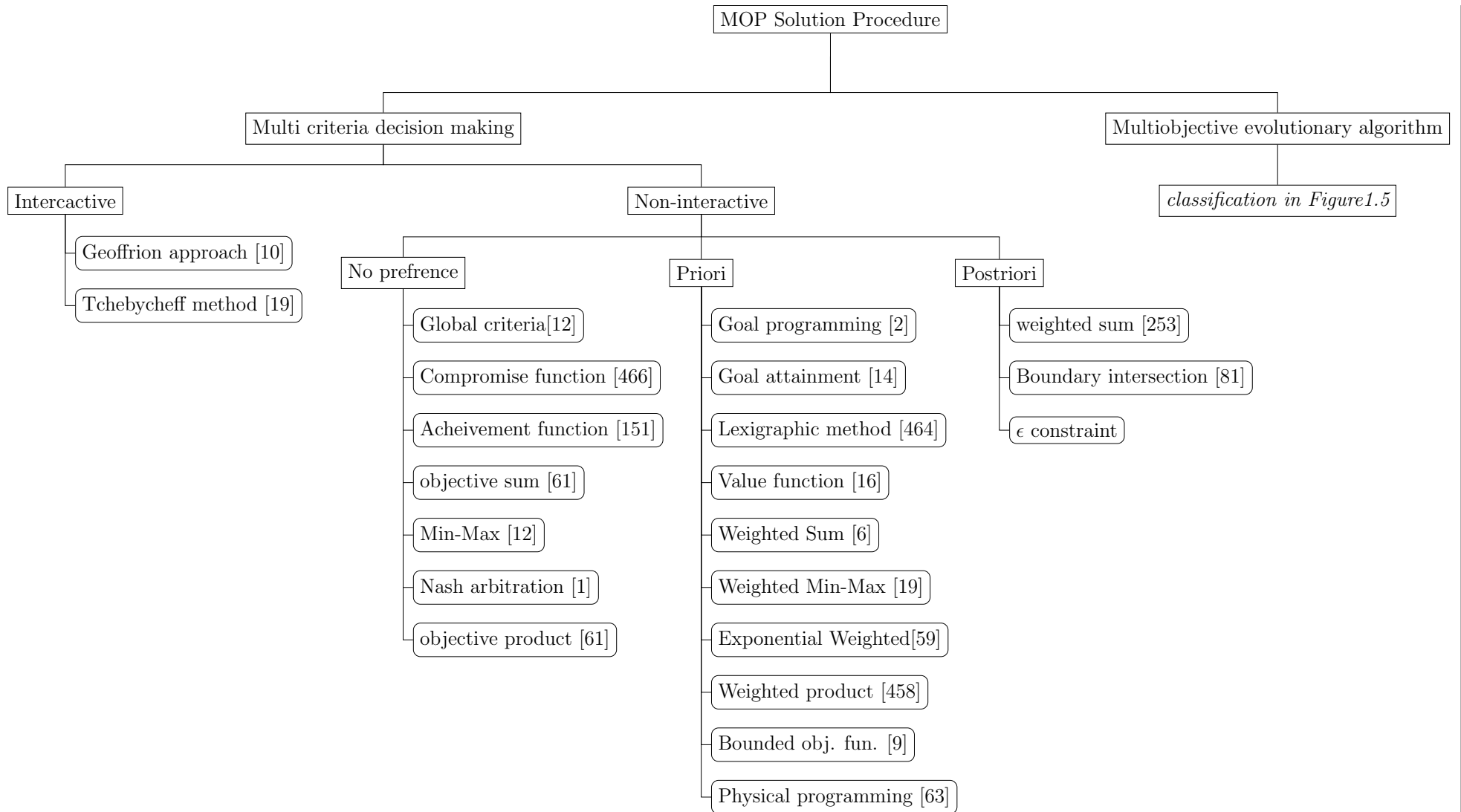


Figure 1.4: Classifications of Multiobjective solution procedures

1.3 MULTIOBJECTIVE OPTIMIZATION

The solution procedure to solve MOP primarily consists of two stages: i) find a set of Pareto optimal solutions ii) selection of the most preferred solution out of Pareto front. The approaches can be classified as *multi-criteria decision making* MCDM and evolutionary multiobjective optimization [282] as shown in Fig. 1.4. MCDM is an extremely important discipline that deals with decision making problems with multiple objectives.

Many approaches and methods have been proposed in recent years to solve multiobjective optimization problems. The DM is responsible for the selection of the best solution from among all the generated solutions. The mathematical programming techniques can be classified on the basis of decision maker's preferences incorporation to the search process. The inclusion of decision maker depends upon variety of factors such as availability of Pareto front, the nature of information available, the nature of final solution, *etc.* These methods to deal with multiobjective optimization problems are broadly grouped under two major categories: non-interactive approach and interactive approach. Each category has its own pros and cons as discussed in the ensuing sections.

1. Non-interactive approach: As Pareto optimal solutions cannot be ordered completely, an extra preference information is expected from a decision maker to select the most preferred solution for a problem involving multiple conflicting objectives [253]. The unavailability of DM and his/her preferences encourage the use of non-interactive approaches. In the non-interactive methods, a global preference function of the objectives is identified and optimized with respect to the constraints. On this basis, the procedures are classified in three ways [459]:

- (a) A priori approach: In this method the user specifies preferences, which is articulated in terms of goals or the relative importance of different objectives. Alternatively, these preferences are also called parameters. The parameters are either being set to reflect decision maker preferences or varied continuously to represent the complete Pareto front. The implementation of methods in this class requires a utility function. Based on utility function the common approaches are weighted global criterion method, weighted sum method, lexi-

cographic method *etc.* The important point of concern is that if appropriate preference is given to objectives, a satisfactory solution can be obtained without utilizing too much effort of DM. Nevertheless, in most of the cases the DM is unaware of its expectations in advance and may be disappointed at the quality of the solution obtained [253].

- (b) A posteriori approach: In many cases the decision maker is unable to express an explicit approximation of the preference function. In such cases generate first choose later approaches are better [116]. The commonly used methods are linear combinations of weights [470], normal boundary intersection [81], ϵ -constraint method [470], normal constraint method [131] *etc.* The methods categorized in this class allow DM to inspect the options before deciding the final solution. The main points of concern about this class is the method to present options (possible solutions) to the DM. One has to compromise between computational expense to extract number of points for presentation and clearer representation of the Pareto optimal set. Therefore, this class is suitable for MOP with lesser number of conflicting objectives [147].

The **weighted sum method** a common approach to MOP can be used as a posteriori approach in such a way that distinct weight vectors are used to generate different Pareto optimal solutions. It is explained as follows:

The weighted sum method is also known as a parametric approach. The MOP is converted to *scalar objective optimization problem* (SOOP) as follows:

Minimize

$$\sum_{k=1}^M w_k f_k(x) \tag{1.11}$$

subject to:

$$\sum_{k=1}^M w_k = 1, \quad w_i \geq 0 \quad (k = 1, 2, \dots, M) \tag{1.12}$$

where $f_k(x)$ is the k^{th} objective function. w_k is the weight associated with k^{th}

objective function.

The approach yields meaningful results to the decision maker only when solved many times for different values of *normalized weight* w_k . The weighting coefficients do not reflect proportionally the relative importance of objectives, but are only factors which, when varied locate points in the non-inferior set [463]. In addition, the related discussion includes a partial weighting method [21], graphical interpretation of the weighted sum method [70], a mathematical relation between weights and DM's preferences [473]. The selection of weights being crucial task for successful application of the weighted sum method, results in the development of systematic approaches which are surveyed by [157] *etc.*

The main disadvantages of the weighted sum method are that despite many methods for determining weights, a satisfactory, a priori selection of weights does not necessarily guarantee that the final solution will be acceptable. The weights must be functions of the original objectives, not constants, in order for a weighted sum to mimic a preference function accurately [63]. Further, it is impossible to obtain points on non-convex portions of the Pareto optimal set in the criterion space [98]. In addition, the consistently and continuously varying weights may not necessarily result in an even distribution of Pareto optimal points and an accurate, complete representation of the Pareto optimal set [70].

- (c) No-preference methods: In no-preference methods, the solution process doesn't considers the opinions of the DM. Thus, the problem is solved using some relatively simple method and the idea is to find a compromise solution typically in the middle of the Pareto optimal set because there is no preference information available to direct the solution process otherwise. These methods are suitable for situations where there is no DM available or (s)he has no special expectations of the solution. An implicit function/ relation is responsible for selection of acceptable solution without the involvement of preferences. Therefore, this class is also known as a neutral preference approach.

2. Interactive approach: In the interactive method, the DM identifies a local preference function or trade-off among objectives and the solution process gradually proceeds towards the globally satisfactory solution. An iterative solution algorithm repeats several times to generate a solution pattern. After each iteration, the DM evaluates the available solution and specify preference information in an appropriate form that the iterative process can utilize. The analyst aims at determining the preference structure of the DM in an interactive way. The underlying solution method utilizes this preference information to generate new solution patterns at the next iteration. This procedure is continued till DM is satisfied with some of the selected solutions.

Algorithm 1.4 Interactive approach procedure

```
while DM not satisfied with solution do  
    Find a Pareto optimal solution  
    Interact with decision maker  
    Update preferences  
    Update priorities of DM  
end while
```

An important aspect of this approach is that the DM is also supported in the iterative process. The DM who is specifying preferences can learn about the interdependencies in the problem and adjust its priorities at each iteration. The information obtained during the current and the previous iterations can be used by DM to state preferences as per expected solution in the next iteration [284]. The DM works in the learning phase, where it is allowed to adapt his/ her mind to new requirements. In short, there are a large number of traditional approaches used to solve MOPs, yet these methods have limitations *viz.* requires a number of trials to obtain Pareto front points, domain knowledge requirement, sensitive to the shape of Pareto front. These limitations lead to the development of another class of MOP solution approaches called *Multi Objective Evolutionary Algorithm* (MOEA).

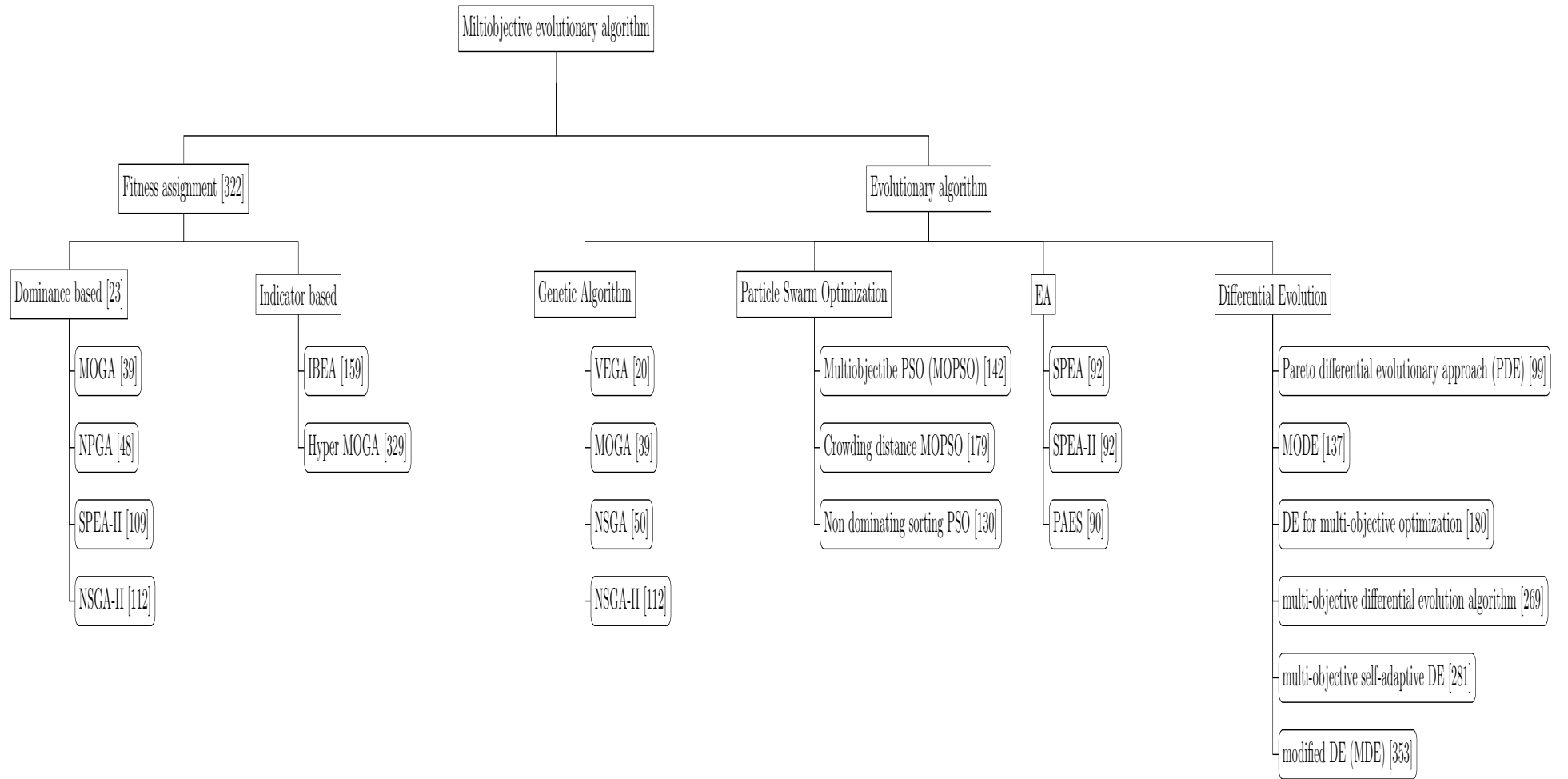


Figure 1.5: Multiobjective evolutionary algorithm classification

1.3.1 Multi objective Evolutionary Algorithm

The development of MOEA has originated from EA's working principle. The EAs are stochastic approaches that mimic some kind of a natural evolutionary process. David Schaffer, introduces the first MOEA in 1984 [20]. Since then, till date a wide variety of

Algorithm 1.5 Multiobjective evolutionary algorithm pseudo code

Initialize population $x(t)$, $i \leftarrow 0$, $t \leftarrow 0$

```
while stopping criteria unmet &  $++t \leq \text{max. number of iterations}$  do
  while  $++i \leq N_P$  do
    Evaluate objective vector  $F(x_i)$ 
    Assign a fitness to  $x_i$ 
  end while
  Select a group of parent  $x'$  based on fitness value
  Recombine  $x'$  to generate  $x''$  offsprings
  Perform mutation on  $x''$ 
  Combine  $x'$  and  $x''$ 
end while
```

MOEA based on famous stochastic approaches such as GA, PSO, ACO, SA *etc.* are proposed. In 2006, a GA based approach to solve MOP has been discussed [467]. A number of comprehensive review articles are published in literature to investigate the MOEAs such as [282, 322] *etc.* The MOEA offers important features like: ability to approximate true optimal set in reasonable computational time in a single run, less susceptibility to shape or continuity of Pareto front, requires less domain knowledge, easy implementation *etc.* The MOEA aims to achieve the objectives, namely i) minimization of distance of the generated vectors to true Pareto front. ii) maximization of diversity of Pareto front approximation. The basic structure of MOEA is explained by algorithm 1.5 [282].

1.3.2 Metrics for Multiobjective Evolutionary Algorithm

As seen in the above discussion that a wide classification of multiobjective solution procedures are proposed, and none of them is suitable for all kinds of application. In order to compare the strengths and weaknesses of various algorithms a metrics survey is presented by Carlos and Colelho [88]. An important literary study related to MOEA metrics is tabulated in Table 1.2.

Table 1.2: Metrics for multiobjective algorithm performance

Metric	Concept	Reference
Generational distance	Distance of solution from true Pareto-front	[86]
Pareto-front Spread	Statistical metric to measure Pareto front points distribution	[50]
Objective space size	Coverage of objective space by non dominated solutions	[92]
Metric fraction	Non dominated solution set comparison	[92]
Solutions number	Number of solutions obtained	[96]
Function evaluation	Time/algorithmic complexity of solution procedure	[82]
C-measure	Compare the relative quality of two sets of solutions	[92]
D-measure	Measures proximity to the optimal solution	[92]
U-measure	Uniformity of Pareto front	[129]

1.4 ECONOMIC LOAD DISPATCH

Economic load dispatch (ELD) is an optimization problem to schedule the electric power generators cost effectively to supply demand load power. The problem also involves transmission losses apart from non-linearities of power generators. Commonly considered the non-linearities are *Prohibited operating zones* (POZ) [126], security constraints [22], spinning reserve constraint [17], valve throttling losses [4], ramp rate limits [43] *etc.* Today's ELD problem exhibit complex and highly nonlinear characteristics, because of the large number of equality and inequality constraints.

In the 1930's the methods used to solve ELD are: the base point load method, best point loading, equal incremental approach [26]. The use of digital computers for load scheduling begins in 1954, thereby starting a new era of solution approaches to solve ELD problems. The efforts include mathematical programming methods and conventional optimization techniques such as lambda iteration method and gradient search techniques [41]. However, the applicability of gradient methods requires completely differentiable and continuously defined objective function [84]. The other applied methods are branch and bound method [35] and Newton method [33], but such methods require assumptions such as piecewise linear cost function. The conventional techniques may not converge to feasible solution with non-differentiable or discontinuous objective function(s) [364]. The optimization methods such as *linear programming* (LP) [53], *quadratic programming* (QP) [11], *Lagrangian relaxation* (LR) [29] and *dynamic programming* (DP) [32] are able to solve

non-smooth and non-convex characteristics of dispatch problem but these methods suffer from dimensionality curse. So, the classical methods are incapable of providing the global optimum in the aspect of rapidity and accuracy [340].

An alternative approach for economic dispatch problems is artificial intelligence techniques [104]. The contributed work includes neural network applications[87, 56], probabilistic neural network [28], two phase neural network [149], expert systems [25] *etc.* Further, the Hop-field network has been widely applied to solve the ELD problem [42]. Some of the examples are adaptive Hop-field neural network [83], improved Hop-field network [55], augmented Hop-field network [77], a fast computation Hop-field method [76] *etc.* Bansal [186] has provided an overview and literature survey of artificial neural networks applications to power systems. A view of applications of optimization methods for power system has been presented in [161].

Further, the population based random search techniques are widely used for solution of ELD problem. GA with multiplier updating has been used to solve the ELD with valve point effects and multi-fuel options [163]. A combination of GA with active power optimization based on the Newton's second order approach has been proposed. [316].The solution of security constrained ELD problem has been obtained through the application of EP [181]. The EP techniques has been used with three types of mutation: Gaussian, Cauchy and combined GaussianCauchy mutation [198]. An ES based approach has been proposed for the solution of the ELD problem with noncontinuous and non-smooth/non-convex cost functions considering generator constraints [177]. A survey of soft computing technique has been presented and the issues related to the implementation of the soft computing techniques are highlighted for a successful application to solve ELD problem. The discussion is focused on Hop-field neural network approach, multilayer perceptron neural networks, GA and tabu search application in the ELD problem and concludes that stochastic global optimization approaches exhibit very good performance [237].

The PSO technique has been applied to solve ELD [126, 176]. The basic PSO has certain deficiencies and to overcome these four modified versions of PSO have been applied to the ELD with valve-point effects. The proposed strategies, explore the vicinity of particle's best position found so far leads to a better result [190]. Moreover, combination of

chaotic sequence and PSO is presented to solve ELD problems with generator constraints. These methods are better alternative approaches than the traditional PSO algorithm [216, 246]. PSO performs well for less complicated problems but fails to locate global minima for complex multi-minima functions. Therefore, a Gaussian random variable is introduced to the velocity term which improves search efficiency and guarantees a high probability of obtaining the global optimum without considerably worsening the speed of convergence and the simplicity of the structure of PSO [411]. Further, an attractive and repulsive PSO [208], new particle swarm optimization [227] and anti-predatory activity based PSO [262] algorithm has been employed for ELD solution. Alternatively, a fast PSO approach [332], improved parallel PSO approach [315] for solving the ELD problem has been proposed. The experimentation on smooth and non-smooth load functions, indicate that the proposed approach provides good results in lower computational time. Another category of PSO variants used to solve ELD have entirely different operating mechanism, *e.g.* bare-bones PSO algorithm [371], θ -PSO [379] resulting in the control parameter tuning free particle updating strategy. A survey of PSO applications in electric power systems is presented and with the development of deregulated electricity market, PSO applications in bidding strategy, congestion management, market simulation and behavior analysis are needed to be studied further [234]. Alrashidi *et al.* have presented comprehensive coverage of PSO applications to solve electric power system optimization problems [185]. It highlights many applications in which PSO was successfully applied and concludes that PSO is still in its infancy and further development and research are needed to enhance its overall performance characteristics [185]. A summary of applications of PSO techniques to the ELD problems and their advantages and disadvantages in resolving the ELD problems have been discussed in [170]. Furthermore, the possible future applications of PSO in the area of electric power systems and its potential theoretical studies have been concluded [270]. The combination of PSO with a Quasi-Newton method has been proposed, where the former produces good potential solutions, and the latter fine-tunes the final solution [195]. An efficient and reliable GA based PSO technique to solve the ELD problem in power systems is presented. The comparison of results proved that the proposed algorithm can obtain higher quality solutions efficiently in ELD problems [230].

A self-organizing hierarchical PSO is used for the non-convex ELD. It is observed that the time-varying acceleration coefficients improve performance [243].

In another work, the DE algorithm has been studied for solving the ELD problem in power systems. The special measures were taken to satisfy equality and inequality constraints [254]. An improved DE algorithm for solving the ELD problem with nonlinear generator features is proposed, which outperforms other contenders [218]. A variable scaling hybrid DE is used to solve the large-scale power dispatch problems. The hybrid DE method based on parallel processors of the two-membered evolution strategy with variable scaling factor is used [274]. Further, three chaotic DE methods are proposed based on the Tent equation to solve problems valve-point effects. The chaotic sequences are applied for DE parameter tuning, a chaotic local search operation and to handle the complicated constraints efficiently [344]. A combination of chaotic DE and QP is used to solve ELD problem [194].

The application of other meta-heuristic to solve ELD includes the use of the taguchi method [172], SA [205], harmony search algorithm [280], self-organizing migrating algorithm [309], firefly algorithm [370], cuckoo search algorithm [387] *etc.* An efficient KHA with the crossover and mutation operation to solve both convex and non-convex ELD problems of thermal power units has been proposed [397]. A binary successive approximation based evolutionary search strategy has been applied to solve the economic emission load dispatch problem in which a slack generator unit is used to meet the load demand [277]. A random drift PSO algorithm, to solve the ELD problems from power systems area, employs a set of evolution equations that can enhance the global search ability of the algorithm [403]. A backtracking search algorithm for solving both convex and non-convex ELD problems has been successfully applied, which is another evolutionary algorithm for solving of numerical optimization problems as it uses a single control parameter and two crossover and mutation strategies for powerful exploration of the problem's search space [398]. A *modified flower pollination algorithm* (MFPA) is proposed in which the local pollination of FPA is controlled by a scaling factor and an intensive exploitation phase is added to tune the best solution. The effectiveness of MFPA is tested on some mathematical benchmarks and four large practical power system test cases [413]. An optimization

method called root tree optimization algorithm has been proposed as ELD solution procedure [445]. A modified artificial bee colony algorithm have been proposed that introduces a new relation to update the solutions within the search space, in order to increase the ability of algorithm to avoid premature convergence and to find stable and high quality solutions [424]. Further, the application of chaos theory has been widely explored in the solution procedures of ELD. The case studies include chaotic artificial immune network [278], chaotic teaching-learning-based optimization [442], chaotic bat algorithm [426] *etc.*

In addition, the integrated and hybrid techniques are widely accepted procedures for ELD problems. The integrated algorithm includes a combination of EP and SA [325], society-civilization algorithm with PSO [294], harmony search algorithm and PSO [348], ACO and a real coded GA [355], DE and harmony search [388], PSO and GSA based on fuzzy logic [414], DE algorithm based on PSO [386] *etc.*

Alternatively, the hybrid algorithms such as the BFA and simplex algorithm [255], multi-agent based hybrid PSO [343], chaotic PSO algorithm and SQP [354], bee colony optimization and SQP [374] *etc.* Further, opposition based hybrids such as harmony search with arithmetic crossover operation and opposition based learning [402], an invasive weed optimization merged with oppositional based learning has been used to solve the ELD problems [410].

A decentralized approach for solving the ELD has been presented. The proposed approach consists of two or three stages. In the first stage, a flooding-based consensus algorithm has been proposed in order to achieve consensus among the agents with respect to the units and system data. In the second stage, a suitable algorithm has been used for solving the economic dispatch problem in parallel. For cases in which a nondeterministic method is used in the second stage, a third stage is applied to achieve consensus about the final solution of the problem, with a flooding-based consensus algorithm for sharing the information required during this stage. The proposed approach is highly effective for solving the non-convex formulation of the economic dispatch problem and for incorporating transmission losses accurately in a fully decentralized manner [415].

To summarize, the ELD problem is widely modified since its introduction in early 20's. Today's ELD problem exhibit non-linear, non-smooth and non-convex characteristics,

which is difficult to solve by conventional techniques. The constraints consist of the generator ramp rate limits, non-convex and discontinuous nature of prohibited operating zones, non-smooth characteristic of valve-point effects, multiple fuel generation units, and transmission losses in realistic power systems. Therefore, the new solution procedures to solve ELD are invented, which evolves through phases of development, where each phase has its own significance. In the past decade, the population based techniques remain prominent and proved their proficiency to obtain the solution. A wide range of heuristic approaches with promising performance are introduced which are computationally efficient in terms of memory and speed. In the initial phase EP has dominated, whereas in the later phase the PSO and DE have emerged as better techniques, with the highest number of contributions. The integrated and hybrid techniques are also proved worth as an ELD solution procedure.

Emission dispatch problem plans the electric power generators with an aim to minimize the pollutants emission [46]. The earlier effort to solve the emission minimization problem is reported in 1971, where on-line steam unit dispatch program is developed, which results in the minimizing of Nitrogen Oxide emissions. A unique mathematical representation of the steam generating units coupled with a Newton-Raphson convergence technique for dispatch produces base points and participation factors for any load level and any unit configuration. It concludes that emission reduction results in a cost increase [8]. A study on scheduling of thermal generating units to meet the requirements of the 1990 amendments to the Clean air act with respect to SO_2 and NO_x emissions is presented and a set of dispatching algorithms is included along with a solution algorithm [57]. A dynamic dispatch procedure is proposed, which is capable of taking into account the integral nature of the emission constraints. The management of multi-fuel plants with different pollution rates, are taken into account with the purpose of obtaining a cost-effective operation for all thermal plants in compliance with emission limitations [31].

1.5 MULTIOBJECTIVE LOAD DISPATCH

A *multiobjective load dispatch* (MOLD) problem considers more than one non-commensurable objectives simultaneously and a single option is expected from the solution procedure.

The cost and gaseous pollutant emission are commonly considered objectives. As these objectives are of a conflicting nature, it is impossible to obtain such a solution. So, a compromising or satisficing solution is selected, which satisfies the various objectives at-least upto considerable extent [147]. Talaq and El-Hawary [51] have summarized the conventional procedures used to solve MOLD problem. A dynamic programming recursive approach is used for the emission constrained economic dispatch problem. The optimization algorithm has been developed with a sequential dynamic programming model [200]. A GA and fuzzy based approach has been applied to search for the active and reactive power scheduling for MOLD [188].

The MOLD requires a procedure to handle conflicting objectives of the problem. The most commonly employed techniques are price penalty factor, ϵ - constraint method and weighting sum method and its alternatives. Case studies that involve application of the price penalty factor method are optimized with shuffled frog leaping algorithm [273], PSO [298], GSA [358], spiral optimization algorithm [393] *etc.* The MOLD is handled using the price penalty factor method and flower pollination algorithm is used to solve dispatch problem [425]. The ϵ - constraint method involves minimizing of an objective, which represent the other objectives in the form of a constraint. Some of the exemplary cases to solve MOLD involve a GA approach with ϵ -dominance [288], bacterial foraging meta-heuristic algorithm [320]. Moreover, fuzzy decision making process is employed to extract one of the Pareto-optimal solutions as the best compromised non-dominated solution [327]. Fuzzy logic has been applied in combination with EP to solve the emission constrained economic dispatch problem [258].

The weighted sum strategy converts the multiobjective problem of minimizing the vector into a scalar problem by constructing a weighted sum of all the objectives. A MOLD problem relating to real and reactive power scheduling of thermal power generating units has been solved using a weighting method to generate non-inferior solutions and interaction with the decision maker is obtained via cardinal priority ranking of the objectives. The cardinal priority ranking is constructed in the functional space and then transformed into the decision space, so the cardinal priority ranking of objectives relates the decision makers preferences to non-inferior solutions through normalized weights [297]. A binary

successive approximation-based evolutionary search strategy has been proposed to solve the MOLD problem by searching the generation pattern of committed units, fuzzy set theory helps the decision maker to choose the optimal operating point over the trade-off curve and adjust the generation schedule in the most preferred manner. The weight pattern assigned to objectives is searched for more significant digits in a fixed number of iterations [277]. A multiobjective EP method to solve the MOLD has been proposed, where the single objective optimization problem is formulated using the weighted sum method [265]. An explicit interactive trade-off analysis based on the surrogate worth trade-off function to determine the most preferred solution has been undertaken [229]. A multiobjective backtracking search algorithm for MOLD problem, where the multiobjective problem is converted to single objective using a weighting sum approach [449]. A MOLD problem is converted into a scalar optimization problem which is solved for a different set of weight pattern to generate non-inferior solutions [207]. The PSO algorithm has been used to solve MOLD problem and considers deregulation of the power industry. It is formulated as a highly nonlinear constrained MOLD with conflicting objective functions and PSO algorithm is used to solve the formulated problem [184].

A fuzzified multiobjective PSO algorithm has been implemented for MOLD considering both economic and environmental issues [232]. An optimization technique based on progressive articulation of preference information is presented to solve MOLD. The proposed method handles the problem in an interactive way and does not need to know any global preference structure or some type of initial goals of the decision maker for the objectives [233]. An orthogonal least-squares method is combined with the enhanced PSO to construct the radial basis function network for real-time power dispatch. The goals considered are fuel cost, power wheeling cost, and NO_x/CO_2 emissions. The radial basis function network is composed of three-layer structures, which contain the input, hidden, and output layer [221]. A multiobjective PSO for MOLD is proposed by redefining global best and local best individuals in the multiobjective optimization domain. A clustering algorithm to manage the size of the Pareto-optimal set and fuzzy-based mechanism to extract the best compromise solution are imposed [211]. A PSO algorithm has been used to solve the MOLD problem formulated as a multiobjective one with two competing

functions [236]. A modified bacterial foraging algorithm applied for the solution of the MOLD problem. The approach utilizes the natural selection of global optimum bacterium having successful foraging strategies in the fitness function [312]. An application of the ABC algorithm has been used for MOLD problem. ABS strategy is based on mathematical modeling to solve economic, emission and combined economic and emissions dispatch problems by a single equivalent objective function [336].

A modified non-dominated sorting genetic algorithm for MOLD is proposed. The available NSGA-II has the drawbacks such as, lack of uniform diversity in obtained non-dominated solutions and absence of a lateral diversity-preserving operator among the currently-best non-dominated solutions. These two drawbacks have been overcome by introducing dynamic crowding distance and controlled elitism into the NSGA-II [335]. A non-dominated sorting-based hybrid cuckoo search algorithm using conventional cuckoo search algorithm and arithmetic crossover operations has been proposed. The effectiveness of the proposed algorithm is analyzed for various multiobjective cases to illustrate the effectiveness [408]. *Multiobjective Differential Evolution* (MODE) has been proposed to solve the MOLD problem [330]. A multiobjective harmony search algorithm has been proposed for MOLD problem. The algorithm uses a non dominated sorting and ranking procedure with dynamic crowding distance to develop and maintain a well distributed Pareto-optimal [351]. A multiobjective-teaching-learning-based optimization algorithm has been used to solve MOLD problem. In the suggested method, the optimization process is done based on the phase angles, instead of the design variables themselves whereby the nonlinear characteristics of the problem are considered. A fuzzy clustering approach is utilized to handle the size of the repository and obtain profitable solutions from the decision maker's point of view [362].

The other reported work in literature includes a discussion on the use of hierarchical decision structure for modeling and solving MOLD problems in a fuzzy decision environment has been presented. In the proposed approach, minimization of the functions of fuel-cost, environmental-emission and transmission-loss are considered at the three hierarchical levels to solve the problem within a power plant operational system. A priority based linear fuzzy goal programming method is employed to achieve the highest member-

ship value of the defined fuzzy goals [421]. The decision maker preferences are used to guide the search and to select the population for the next generation. An improved DE method is proposed where the selection operation is modified to reduce the complexity of multi-attribute decision making with the help of a fuzzy framework. Solutions are assigned a fuzzy rank on the basis of their level of satisfaction for different objectives before the population selection and then the fuzzy rank is used to select and pass on better solutions to the next generation [422]. A nonlinear fractional programming approach is provided for addressing the environmental economic power dispatch problem. The objective to be minimized are realized by two simultaneous models. The first model is to minimize the quotient of two competing and conflicting functions (i.e. [total emission function]/[total fuel cost function]), and the second one is to minimize the total fuel cost expressed by a quadratic objective function. In particular, the process of solving the first model uses Dinkelbachs algorithm to convert the minimization problem with the nonlinear fractional objective function into a sequence of non-fractional minimization problems [435]. A stochastic MOLD problem has been formulated with consideration of the uncertainties in the system production cost and nature of the load demand along with risk as another conflicting objective to be minimized. The weighted minimax technique is used to simulate the trade-off relation between the conflicting objectives in the non-inferior domain [37].

A review presents major studies of power system ELD techniques, such as neural network, fuzzy logic, genetic algorithm and other optimization techniques available in the literature. From the literature, it is clearly observed that PSO provides better performance than other conventional techniques. Other modern optimization techniques like ACO can be used for the significant performance of the power system [342].

To summarize the MOLD survey, due to clean fuel act act 1980, the pollutant emission is an essential consideration of present day ELD problem. Since, the two objectives are of a conflicting nature, it is difficult to achieve consensus of suitable solution. In order to accomplish an acceptable solution of the problem that satisfies both the objectives, special multiobjective handling techniques are adopted. The most commonly used methods are weighting method, price penalty method and ϵ -method. Moreover, the MOEA such as

NSGA, Strength Pareto Evolutionary algorithm (SPEA), Pareto Differential Evolution (PDE), Multiobjective Particle Swarm Optimization (MPSO) *etc.* are developed. These are based on population dependent optimization techniques. In both the categories the most commonly used algorithms are GA, PSO, DE, BFA, GSA *etc.* A discussion on optimization techniques is presented in the ensuing sections.

1.6 SCOPE OF WORK

After careful review of the recent reported work by various authors, the intent of the proposed research is outlined here. No single optimization technique suits best for various kinds of real life problems. Direct search methods provide local optimal solution, but it is sensitive to initial guess and step size. Random search methods such as GA, EP, DE, PSO *etc.*, which mimic the behavior of natural species have proved their effectiveness, but still they have problems like sensitive to tuning parameters, encoding schemes, operators *etc.* Improper selection of these parameters results in high computational costs or sometimes pre-mature convergence. The solution procedure of the multiobjective optimization problem is non-inferior due to contradictions among objectives taken and has been obtained either through weighting method or ϵ -constraint method. The major weakness of the ϵ -constraint method is computational inefficiency where there are several objective functions. The number of solutions of the scalar problem required to identify the non-inferior solution set, increases exponentially with the number of objectives. In existing weight simulation technique, weights are simulated in a systematic manner to generate non-inferior solutions so that the trade-off relationship among participating objectives can be investigated. But the number of non-inferior solutions of the problem increases exponentially with the number of objectives, which leads to increase in the burden on decision maker. Another limitation with the simulated weight problem is that the decision maker may not be provided with the weight set that corresponds to the global solution during simulation by virtue of step size considered. Thus decision maker selects the best solution only among generated non-inferior solutions and provided to him/her. Therefore, there is great scope to consider ramp rate limits, POZs, system spinning reserve, valve loading effects and multiple fuel source options to solve a realistic economic dispatch problem,

which makes the finding of the optimum solution difficult. Further, it is required to search for better and realistic strategies to achieve various objectives along with desired quality of power supply and satisfy simultaneously various system constraints.

1.7 OBJECTIVES

The research work will investigate the solution methodology of multiobjective optimization problems by exploring interactive and non-interactive solution procedures utilizing fuzzy set theory. The work will be extended to explore the global optimization techniques to solve optimization problems as well as multiobjective optimization problems. It is also the plan of the proposed research to develop refined methodologies which can be applied for optimal power dispatch. To be more specific the proposed objectives of research is outlined below:

1. To develop hybrid optimization techniques using Evolutionary Programming and pattern search, Predator Prey Optimization and pattern search, Differential Evolution and Powells method to solve constrained optimization problems.
2. To validate the applicability of hybrid optimization techniques on the standard test problems.
3. To apply proposed hybrid optimization techniques on multiobjective electric power dispatch problem.

1.8 OUTLINE OF THESIS

This thesis considers the *Evolutionary Programming* (EP), *Predator Prey Optimization* (PPO) and *Differential Evolution* (DE) algorithms to propose their hybridized versions with local search techniques as self sufficient and robust technique to find the solution of single objective and multiobjective optimization problems. Single objective optimization problems undertaken are standard benchmark functions, and real life economic load dispatch of power systems. The multiobjective optimization problems includes power generation cost and emission of gaseous pollutant's objectives, which are conflicting in

nature. Apart from energy balance and decision boundary constraint, the complexities of multi-fuel, valve point loading effect, ramp rate limits, prohibited operating zone have been considered for the power system. A constraint handling strategy has been devised to tackle the equality and inequality constraints violated in the solution search process. To solve multiple objective optimization problem, the interactive weighting method and surrogate worth trade-off approach have been implemented. The work undertaken in the present study has been arranged in the thesis in the following manner:

The significant contributions of various authors related to advances of optimization techniques and its relevant topics have been briefly reviewed in the present chapter. The chapter also reviews the existing practices and options to solve scalar and multiobjective optimization problems. A review of significant contributions to solve ELD and MOLD have been presented.

Chapter 2, presents the mathematical formulation of the economic load dispatch problem along with equality and inequality constraints. In order to obtain the solution of ELD problem, an algorithm *Synergic Predator Prey Optimization* (SPPO) is introduced. In this algorithm the opposition based population initialization process has been incorporated. The behavior of prey has been modified with the concept of cognitive behavior split-up. The cognitive behavior of prey is quantized to local best and local not best parts. Similarly, social behavior depends upon global best and global not best constituents. In order to, ensure exploitation throughout the search process collaborative predator effect has been introduced. The predator's effect is treated as a function of cognitive and social behavior of prey. The concept of variable elimination approach has implemented to satisfy the energy balance constraint. Results are tabulated and compared with the results obtained by other algorithms. Statistical analysis of the results obtained from the proposed algorithm is done with the help of student's T-test and non-parametric Wilcoxon signed rank test.

In chapter 3, the mathematical foundation of multiobjective load dispatch problem has been introduced. A hybrid of chaotic evolutionary programming and Powell's pattern search called *Chaotic Evolutionary Programming Pattern Search* (CEPPS) has been proposed and applied to obtain the solution of MOLD problem. In this algorithm, the con-

ventional evolutionary programming algorithm has been modified with chaotic sequence based population initialization, mutation and selection operators. Two types of chaotic sequence *viz.* Tent map and Gauss map have been implemented. The Powell's pattern search has been hybridized with proposed chaotic EP as an interleaved process. The interleaved improvement is activated under stochastic control. The two objectives of MOLD problem, power generation cost and gaseous pollutant emission have been handled using surrogate worth trade-off approach. The comparison of the obtained results has been performed with other results available in the literature.

In chapter 4, *Adaptive Predator Prey Optimization* (APPO) is established as a solution procedure to solve multiobjective load dispatch problem. The established approach is a refinement of predator prey optimization algorithm. This approach treats the fear factor of prey from predator in relation to the indecisive state of mind. The fear of prey is bifurcated into cognitive and social behavior. An adaptive mechanism is incorporated to decide the velocity limits of prey. In case of the velocity constraint violation by prey, a discrete velocity variation concept has been implemented. An interactive weighting method has been applied to solve MOLD, whereby the weight vector is a part of search variables explored by APPO. It resulted in advantage that the weight vector is updated automatically, to generate complete Pareto front. Fuzzy decision making is adopted to select the best compromised solution of the problem. The results obtained by APPO are tabulated and compared with the results available in the literature.

Chapter 5, presents the mathematical ground work of a natural foraging behavior based algorithm *Fly and Walk Predator Prey Optimization* (FWPO) as a solution procedure. The foraging behavior decisions have been explored to introduce the concept of fly and walk. The fly operation ensures adaptive exploration, whereas walk is a Hooke Jeeves motivated exploitation process. A heuristic approach called 'direction of turn' has been incorporated to satisfy the system decision boundary and power balance constraints. A fuzzy theory oriented surrogate worth trade-off concept has been implemented as an interactive process to decide the compromised solution of multiobjective problem. A Pareto uniformity approach is exploited to indicate the distribution of compromise solutions. The impact of velocity limit variation on convergence behaviour of FWPO has

been verified using standard test functions. The results obtained by FWPO for ELD and MOLD problem are tabulated and are compared with the available literature.

In chapter 6, an algorithm *Chaotic Differential Evolution and Powell's method* (CDEPS) is proposed. This algorithm considers chaotic sequence to establish stochastic behavior as well as maintain control the parameters of differential evolution algorithm. Two chaotic sequence maps, namely Guass map and Tent map have been incorporated in the population initialization, evolution and selection operation. The Powell's pattern search has been implemented as an interleaved process to improve the quality of best solution searched during the iterative process. The resultant algorithm is implemented to obtain the solution of MOLD. The surrogate worth trade-off approach has been applied as a non-interactive approach to handle the conflicting objective functions. This modification helps to obtain the solution of the multiobjective problem without the need of Pareto front. The result obtained by chaotic differential evolution Powell's pattern search method are compared with the available literature.

Finally, Chapter 7 presents the observations and conclusion of the research work carried over and documented in this thesis. Further, new search directions identified while carrying over this research work, which can be explored in the future are highlighted.

CHAPTER 2

SYNERGIC PREDATOR PREY OPTIMIZATION FOR ECONOMIC LOAD DISPATCH

2.1 INTRODUCTION

Economic load dispatch (ELD) is a constrained optimization task in power system to schedule committed generating units for an economical power generation. Practical ELD problem exhibits nonlinear characteristics, because of the multi-fuel options [397], valve-point loading effect, ramp-rate limits, *prohibited operating zone* (POZ) constraints [270] and transmission losses [277]. Further, [183] has considered the effect of spinning reserve requirement in ELD problem.

The conventional optimization techniques [41] requires completely differentiable and continuous objective function and constraints [84] to solve optimization problems. In the last decade, various researchers have solved ELD problem by conventional techniques after simplifying the ELD problem with a number of assumptions. Because of this, conventional techniques are unable to search the global optimal solution. The dynamic programming can handle the nonlinear and discontinuous optimization problems, but requires large computational time and suffers from the curse of dimensionality [170]. To overcome disadvantages of conventional techniques, various global optimization techniques such as *Genetic Algorithm* (GA) [66], *Particle Swarm Optimization* (PSO) [126], *Simulated Annealing* (SA) [94], *Evolutionary Programming* (EP) [135], *Biogeography Based Optimization* (BBO) [263], *Differential Evolution* (DE) [334] etc. and their variants are applied to solve ELD problem. Recently, the ELD problem is solved using teaching learning al-

gorithm [409], particle diffusion algorithm [416], social spider algorithm [443], modified artificial bee colony algorithm [424], chaotic teacher learning algorithm [442] *etc.*

Among population based techniques, the PSO introduced by Kennedy and Eberhart [54], is found to be robust in solving nonlinear optimization problems. PSO caught attention of the power system community and succeeded to solve many complex power system optimization problems. Oca et al. presents the basic concepts and variants of PSO [276]. In brief the advantages of PSO are concluded as: simple concept, easy implementation, computational efficiency in comparison to other heuristic techniques [218]. At the same time the unresolved issues of PSO are: lack of mathematical background, large computational time compared to mathematical approaches, initial point and parameter dependency, optimal design parameters setup, stochastic nature of the final solution, inability to handle scattering and optimization problem, premature convergence/ partial optimism [218] due to homogeneity of particles.

One of the major concerns for PSO performance is a diversity of particles. The attempts to tackle this crucial issue include neighborhood topologies [224], adjustment of parameters in standard PSO [85], parameter adaptation mechanism [219], modified models of PSO [148]. In another effort to ameliorate the performance of PSO, it is blended with local/global search techniques. The examples are a combination of genetic algorithm and PSO [145], hybrid PSO [314]. These approaches improved the solution quality and convergence behavior, but leads to the drawback of algorithmic complexity, computationally onerous [375], problem specific approach *etc.*

In the hunt of betterment usually the existing solution procedures are complexed, although simple natural biological approaches exists for such problems, which are often ignored. These approaches usually lacks theoretical background, but are effective in obtaining solution. Among these the introduction of multi-swarm systems [155], bio-inspired PSO [206] and predator effects [262] based mechanism are also proposed to improve performance of PSO. The *predator prey optimization* (PPO) introduces a second population called predator, to ensure population diversity. The predator and prey are responsible for convergence and diversification respectively [197] The predator always targets the best position of the swarm and the preys avoid it based on its individual decision. The PPO

has better convergence and solution accuracy than PSO [118].

The PPO has the capability to solve large scale nonlinear optimization problems and the numerical result comparison shows the supremacy of PPO over PSO and its variants [400, 401]. In another effort Silva *et al.* [118] have proposed biogeography based optimization method combined with the PPO technique for designing a brush-less DC wheel motor. The success of search algorithm requires balanced exploration and exploitation. The PPOs exploitation capability is uncertain as the predator's effect is a random process. In spite of various successful applications of PPO to solve complex optimization problems, the psychology aspect of prey particles lacks exploration. The prey being a part of the swarm, its individual decision-making is partial. The decision of prey for new place must be a group activity as the swarm together influences decision-making. A groups tend to make decisions that are more extreme than those of its individual members, in the direction of the individual inclinations [155].

In optimization algorithms, the convergence rate of algorithm is highly correlated to quality of the initial population. The proximity of starting point to the optimal solution, results a faster convergence and contrary [405]. In the worst-case initialization, the convergence will take much more time or the solution can be intractable. Looking simultaneously for a better candidate solution in both current and opposite directions may help to solve this problem quickly and efficiently. In opposition based learning research field a variety of new models have been studied for dealing with significant problems. Xu *et al.* [405] presented a survey of opposition-based meta heuristics recently applied in engineering applications.

In the light of various observations of the literature survey, the contributions related to ELD solution methodology are: A new version of the PPO model called *Synergic PPO* (SPPO). The SPPO blends the psychology of preys along with a mathematical model of the PPO. This outcomes in collaborative intelligence that drives the search variables. The proposed SPPO divides the self-intelligence of the prey into two evolutionary elements, i.e., best and non-best experiences. These elements help in information exchange at a steady rate, which ensures uniform diversity. The prey movement will be guided by best and non best experience along with the predator's effect. The predator always chases the

global best prey and the predator's effect depend on the swarms realization of predator. In this research work the collaborative predator's effect is introduced as a gizmo to counteract search in initial and final phase. In the initial phase the predator's effect being small, the search will be more explorative and in the final juncture as the predator's effect will dominate and the search will shift to exploitation. An opposition based initialization of prey particle positions is utilized to start with good candidates of prey particles. As the priori information about the solution is ambiguous, it is better to check the quality of their opposite solutions instead of using random guess. The ELD problem, having multi-fuel options considering nonlinearities of generators, is solved. A variant of the multi-fuel system with transmission losses is proposed and analyzed. The transmission losses are obtained using Kron's relations. In this chapter, generalized benchmark problems and three test power system with different cases are considered to prove the supremacy of proposed SPPO based ELD solution approach.

The chapter is organized into six sections. Section 2.2 presents the ELD problem. Section 2.3 presents the constraint handling procedures applied in the search process, Section 2.4 explains the SPPO based solution procedure to solve the ELD test problems. The comparison of results is presented in section 2.5. Finally, section 2.6 concludes this chapter.

2.2 ECONOMIC LOAD DISPATCH PROBLEM

The ELD problem is defined as to minimize the entire operating cost of a power system and simultaneously meet the system operational constraints. Mathematically, the multi-fuel ELD is stated as below [463]:

Minimize operating cost

$$f(P_j) = \sum_{j=1}^{N_g} (a_{1jk}P_j^2 + b_{1jk}P_j + c_{1jk} + |d_{1jk} \sin\{e_{1jk}(P_j^{min} - P_j)\}|) \quad (P_{jk}^{min} \leq P_j \leq P_{jk}^{max}) \quad (2.1)$$

where P_j is the generated real power and $P = [P_1, P_2, \dots, P_{N_g}]^T$; N_g is the number of generators; a_{1jk} , b_{1jk} , c_{1jk} , d_{1jk} and e_{1jk} are thermal generators' cost coefficients of

j^{th} generator's k^{th} fuel option; P_{jk}^{min} and P_{jk}^{max} are the generator's lower and upper limits for k^{th} fuel option.

subjected to:

(i) Power balance equality constraint [463]

$$\sum_{j=1}^{N_g} P_j - (P_D + P_L) = 0 \quad (2.2)$$

(ii) Generator operating limits [463]

$$P_j^{min} \leq P_j \leq P_j^{max} \quad (j = 1, 2, \dots, N_g) \quad (2.3)$$

(iii) Ramp rate limit [126]

- As generation increases

$$P_j - P_j^0 \leq UR_j \quad (j = 1, 2, \dots, N_g) \quad (2.4)$$

- As generation decreases

$$P_j^0 - P_j \leq DR_j \quad (j = 1, 2, \dots, N_g) \quad (2.5)$$

(iv) Prohibited operating zone constraint [126]

$$\begin{aligned} P_j^{min} &\leq P_j \leq P_{j,1}^L && (j = 1, 2, \dots, N_g) \\ P_{j,i-1}^U &\leq P_j \leq P_{j,i}^L && (i = 1, 2, \dots, N_{zj}; j = 1, 2, \dots, N_g) \\ P_{j,Nzj}^U &\leq P_j \leq P_j^{max} && (j = 1, 2, \dots, N_g) \end{aligned} \quad (2.6)$$

where P_D and P_L are the forecast demand and transmission loss of the network respectively; N_{zj} is number of prohibited zones of the j^{th} generator; P_j^0 is the previous generated power; UR_j and DR_j are the up-ramp limit and down-ramp limit of the j^{th} generator; $P_{j,i}^L$ and $P_{j,i}^U$ are lower and upper range of i^{th} prohibited zone respectively of j^{th} generator.

The transmission loss P_L is represented by Kron's loss formula [277]:

$$P_L = B_{00} + \sum_{j=1}^{N_g} B_{j0} P_j + \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} P_i B_{ij} P_j \quad (2.7)$$

and B_{00}, B_{j0} and B_{ij} respectively are transmission loss coefficients.

2.3 CONSTRAINT HANDLING

The constrained nonlinear optimization problem can be solved by direct and indirect methods. In the direct methods, the constraints are handled in an explicit manner, whereas indirect methods convert constrained problems into unconstrained minimization problem. The equality constraints are handled by variable elimination method and penalty method as discussed in ensuing subsection:

2.3.1 Equality Constraints Handling

In the equality constrained optimization problem, one variable is theoretically eliminated by using one equality constraint. The variable to be eliminated is called slack variable. Let N_g committed generating units are responsible for delivering the power output subject to their respective energy balance constraint and the capacity constraints. N_g-1 variables are searched using SPPO algorithm. Let P_s is the power of a randomly selected slack generator among committed units to meet the energy balance constraint. The power output of the slack unit is computed by rewriting the energy balance Eq.(2.2). Considering Eq.(2.2) and (2.7), a simplified quadratic equation is written as follows:

$$X P_s + Y P_s + Z = 0 \quad (2.8)$$

where

$$X = B_{ss}$$

$$Y = \sum_{j=1}^{N_g} (B_{js} + B_{sj}) P_j + B_{s0} - 1$$

$$Z = P_D + B_{00} + \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} P_i B_{j0} P_j + \sum_{j=1}^{N_g} B_{j0} P_j - \sum_{j=1}^{N_g} P_j$$

The positive roots of the equation are obtained as

$$P_s = (-Y \pm \sqrt{Y^2 - 4XZ})/2X \quad \text{and} \quad Y^2 - 4XZ \geq 0 \quad (2.9)$$

In the process of equality constraint handling, the power of slack thermal unit can violate its boundary. The exterior penalty method is applied to handle the violation. So, the augmented objective function $A(P)$ becomes:

$$A(P) = f(P) + r_k \langle T_s \rangle \quad (2.10)$$

where

$$\langle T_s \rangle = \begin{cases} (P_s - p_s^{max})^2 & ; P_s \geq P_s^{max} \\ (P_s^{min} - P_s) & ; P_s < P_s^{min} \\ 0 & ; otherwise \end{cases} \quad (2.11)$$

and r_k is the penalty parameter with high value.

2.3.2 Ramp Rate Limit Constraint Handling

The generator ramp rate being an inequality constraint restricts the operating scope of on-line units between two adjacent points. Mathematically, it is specified by Eq. (2.4) and (2.5). This constraint is undertaken by adjusting the minimum and maximum limit of generator's power as follows:

$$max(P_j^{min}, P_j^0 - DR_j) \leq P_j \leq min(P_j^{max}, P_j^0 - UR_j) \quad (j = 1, 2, \dots, N_g) \quad (2.12)$$

2.3.3 Prohibited Operating Zone Constraint Handling

The POZ constraints on the system are specified by the Eq. 2.6. The operation in these areas is to be avoided, for better performance of the system. A heuristic approach is

applied to adjust the generator power within a permitted range, in case the POZ limits are breached. The prohibited zone constraint violation is avoided by updating the generation as described under:

$$P_j = \begin{cases} P_{j,NZ-1}^L - rand() \frac{P_{j,NZ-1}^U - P_{j,NZ-1}^L}{P_{j,NZ-1}^U} & ; P_j \leq P_{j,NZ}^L \\ P_{j,NZ+1}^L - rand() \frac{P_{j,NZ+1}^U - P_{j,NZ+1}^L}{P_{j,NZ+1}^U} & ; P_j \geq P_{j,NZ}^U \quad (j = 1, 2, \dots, N_g) \\ P_j & ; otherwise \end{cases} \quad (2.13)$$

The power of the generator is updated repeatedly until there is no violation of prohibited zones.

2.4 SYNERGIC PREDATOR PREY OPTIMIZATION

Kennedy and Eberhart, (1995) [54] introduced PSO, which considers that solution of the problem is population of particles moving through the problem hyperspace with distinct velocities. At each iteration, the velocities of the individual particles are stochastically adjusted according to self and social best experiences decided according to objective function [206]. The movement of each particle naturally evolves to an optimal or near-optimal solution. The mathematical foundations of PSO are elaborated in section A.1 of appendix A. Higashitani *et al.* [197] proposed a modification called *predator prey algorithm* (PPO), to shun the problem of stagnation at local minima of the basic PSO algorithm. PPO offsets exploration and exploitation by using an element called predator. To enhance the computational time to achieve optimal result and secure diversity of prey population in the evolutionary process, this chapter suggests improvements in prey behavior as: (i) opposition based initiation, to ensure good quality solutions at initial state and (ii) collective intelligence based dynamics, to ensure the prey dynamism in exploration and exploitation phase. The proposed algorithm named SPPO is shown as a flowchart in Fig. 2.1 and is discussed in the following sections:

2.4.1 Representation of Prey Particles

In N_g dimensional search space, P and v denote prey particles position and velocity. Therefore, the position and velocity of i^{th} prey at t^{th} iteration is represented as $P_i^t = [P_{i1}^t, P_{i2}^t, \dots, P_{iN_g}^t]$ and $v_i^t = [v_{i1}^t, v_{i2}^t, \dots, v_{iN_g}^t]$ respectively. The previous best position of the i^{th} prey particle is represented as $P_i^B = [P_{i1}^B, P_{i2}^B, \dots, P_{iN_g}^B]$. g^B represents the best prey particle position among all the particles. The preys position and velocity are initialized as follows:

2.4.2 Initialization of Prey Swarm

In SPPO algorithm, the prey particle and predator are two principle components. The prey particle positions denote generator's power within the operational constraints of generators, Eq.(2.3). In the absence of any prior knowledge about the optimal resolution, the power generations, which are treated as prey particle positions P_i^t , are generated randomly using the mathematical relation given below.

$$P_{ij}^0 = P_j^{min} + rand()(P_j^{max} - P_j^{min}) \quad (j = 1, 2, \dots, N_g; i = 1, 2, \dots, N_P) \quad (2.14)$$

where P_{ij}^0 is the initial position of i^{th} prey particle at $t = 0$ iteration representing generation of j^{th} generator, N_P is number of prey particles.

2.4.3 Opposition Based Prey Swarm Initialization

The computation time to arrive at the optimal solution depends on the quality of the initial population of prey particle. If the starting point is close to the optimal solution, it can result in faster convergence. On the other hand, if it is very far from the optimal location (opposite location in worst case), the convergence will be slower or even the optimal solution can be untraceable. Therefore, to improve the computational time, it is better to start with better population of prey particles giving quality solutions. An opposition based scheme is applied to the SPPO during the population initialization that selects good candidatures of the current population of prey particles. One of the ways to achieve this is to simultaneously generate the initial solution randomly and check its

opposite solutions in the specified search range and finally keep best half solutions as initial guess out of the total $2N_P$ (random and its opposite) solutions. The main idea is to find better candidate solutions out of an estimate and its corresponding opposite estimate within specified search range. The opposite solutions representing prey position are generated as follows:

$$P_{i+N_P,j}^0 = P_j^{max} - P_j^{min} - P_{ij}^0 \quad (j = 1, 2, \dots, N_g; i = 1, 2, \dots, N_P) \quad (2.15)$$

2.4.4 Prey Swarm Velocity Initiation

The prey behavior is followed by position and velocity. The limit of prey velocity is given by:

$$v_j^{min} \leq v_{ij} \leq v_j^{max} \quad (j = 1, 2, \dots, N_g; i = 1, 2, \dots, N_P) \quad (2.16)$$

where $[v_j^{min}v_j^{max}] = k[p_j^{min}p_j^{max}]$; $k < 1.0$.

The initial velocities are generated randomly as follows:

$$v_{ij}^0 = v_j^{min} + rand()(v_j^{max} - v_j^{min}) \quad (j = 1, 2, \dots, N_g; i = 1, 2, \dots, N_P) \quad (2.17)$$

2.4.5 Predator Velocity and Position Initialization

The predator position, PP_j and velocity, Pv_j has dimensions N_g and are initialized randomly within the desired operational limits as follows:

$$PP_j^0 = PP_j^{min} + rand()(PP_j^{max} - PP_j^{min}) \quad (j = 1, 2, \dots, N_g; i = 1, 2, \dots, N_P) \quad (2.18)$$

$$Pv_j^0 = Pv_j^{min} + rand()(Pv_j^{max} - Pv_j^{min}) \quad (j = 1, 2, \dots, N_g; i = 1, 2, \dots, N_P) \quad (2.19)$$

where PP_j^{min} , PP_j^{max} are the lower and upper limits on predator position; Pv_j^{min} , Pv_j^{max} are the lower and upper limits on predator velocity.

2.4.6 Generation of Feasible Solution

The heuristic random search strategy is applied to generate feasible solution by varying the thermal generation represented by prey position to satisfy the power balance constraint. The unmatched power is distributed randomly among the committed generators, while satisfying the power generation limits. Mathematically, the operation is represented below. The generators are updated as:

$$P_{ij}^0 = P_j^{min} + rand()(P_j^{max} - P_j^{min})\alpha_i^t \quad (j = 1, 2, \dots, N_g; i = 1, 2, \dots, N_P) \quad (2.20)$$

where

$$\alpha_i^t = \frac{1}{P_D} \sum_{j=1}^{N_g} P_{ij}^t - (P_D + P_{Li}^t)$$

The procedure is iterated until the equality constraint given by Eq. (2.2) is satisfied.

2.4.7 Updating Prey Positions

The modified velocity and position of each prey is calculated using the current position *w.r.t.* best individual and global experience. One of the usual problems faced by the majority of PSO variants is problem of premature clustering in the early phases of the search operation. This affects in limited positional information exchange in the swarm and hence further improvement of prey positions gets restricted. This phenomenon reduces the capability of preys to explore the search area effectively. This trouble is more prevalent in multi-modal problems, and absence of any mechanism to pass through the vale of multi-modal search domain, results in poor convergence behavior of the algorithm. Hence, the algorithm is unable to realize truly global solution. Even the introduction of local search techniques or other adaptations to counter this problem merely increases algorithmic complexity. Hence, it becomes necessary to introduce a simple bio-inspired mechanism that can elevate above stated problem. It is widely accepted fact that bio-inspired techniques have simple mechanism and are able to solve complex problems effectively. This chapter presents, a modification called collaborative intelligence of prey swarm to balance ex-

ploration and exploitation, which ensure diversity of the swarm. In SPPO, the swarm consists of prey particles moving within search space, whose conduct is regulated by different aspects *viz.* attraction for the best position, repulsion for the not-best position and fear of predator. Each individual will contribute to movement decisions in proportion of these effects. These aspects are detailed as follows:

1. Cognitive behavior split up: Silvakumar et al. [118] proposed a split-up of cognitive behavior of PSO to personal best P_i^B and personal worst position/ experience P_i^B . In such, the particle has to recall both the best and worst experiences (refer Section A.3 Appendix A). The adjustment helps to explore the solutions effectively as it aids in preserving diversity. As the iterative process advances, the best situation is improved, but the worst position remains approximately same. Thus, the effect of P_{ij}^W to update the velocity remains almost negligible and the expected diversity level remains ambiguous. In order to ensure the exploration capability of particles, it is better if the cognitive behavior depends on local best P_i^B and local not-best P_i^{NB} positions, instead of best and worst positions. Similarly, social behavior depends on global best g^B and global not-best g^{NB} positions. Mathematically, velocity is updated by the expression given below:

$$v_{ij}^{t+1} = wv_{ij}^t + c_{11}rand()(P_{ij}^B - P_{ij}^t) + c_{12}rand()(g_j^B - P_{ij}^t) + c_{21}rand()(P_{ij}^t - P_{ij}^{NB}) + c_{22}rand()(P_{ij}^t - g_j^{NB}) \quad (j = 1, 2, \dots, N_g; i = 1, 2, \dots, N_P) \quad (2.21)$$

and

$$P_{ij}^{B,t} = \begin{cases} P_{ij}^t & ; A(P_{ij}^t) \leq A(P_{ij}^{B,t-1}) \\ P_{ij}^{t-1} & ; A(P_{ij}^t) \geq A(P_{ij}^{B,t-1}) \end{cases}$$

$$P_{ij}^{NB,t} = \begin{cases} P_{ij}^t & ; A(P_{ij}^t) \leq A(P_{ij}^{NB,t-1}) \\ P_{ij}^{t-1} & ; A(P_{ij}^t) \geq A(P_{ij}^{NB,t-1}) \end{cases}$$

$$g_j^{B,t} = \begin{cases} g_j^t & ; A(g_j^t) \leq A(g_j^{B,t-1}) \\ g_j^{t-1} & ; A(g_j^t) \geq A(g_j^{B,t-1}) \end{cases}$$

$$g_j^{NB,t} = \begin{cases} g_j^t & ; A(g_j^t) \leq A(g_j^{NB,t-1}) \\ g_j^{t-1} & ; A(g_j^t) \geq A(g_j^{NB,t-1}) \end{cases}$$

where $A(\cdot)$ is the augmented objective function (Eq. 2.10) of the optimization problem to be minimized. C_{11}, C_{12}, C_{21} and C_{22} are the acceleration coefficients. The inertia weight, w is calculated by adopting following relation and it shows the decreasing trend *w.r.t.* iterations

$$w = f_w[(w^{max} - w^{min})/(t/IT_{max})] \quad (2.22)$$

where IT^{max} is the maximum number of iterations. Randomness is maintained by applying a chaotic sequence, which is given below:

$$f_w = 4f_w(1 - f_w) \quad ; f_w \notin [0, 0.25, 0.5, 0.75, 1] \quad (2.23)$$

Exploration is process of visiting an entirely new region of the search space, whilst exploitation is the process of visiting those regions of a search space within the neighborhood of previously visited points. So, after a few initial iterations, the swarm particles improved and P_{ij}^{NB} and g_j^{NB} of i^{th} particle loses its dynamic behavior and becomes a static limit and after that as the iterations progress towards best solution, exploration capability of PSO does not remain affective and the exploitation becomes effective.

2. Collaborative predator effect: In order to exploit the region of interest, particularly when the particles cluster at a better solution, the purpose of fear factor due to predator becomes significant. The promising candidate region is to be intensely searched to locate optimum precisely. The predator 's effect has the ability to

perturb the particles. The predator targets the global best prey particle and the prey particle tries to get rid of predator (refer Section A.2 Appendix A). The predator 's position PP_j^t and velocity Pv_j^t update equations are [197]:

$$Pv_j^{t+1} = c_p rand()(g_j^B - PP_j^t) \quad (j = 1, 2, \dots, N_g) \quad (2.24)$$

$$PP_j^{t+1} = PP_j^t + Pv_j^{t+1} \quad (j = 1, 2, \dots, N_g) \quad (2.25)$$

where c_p is acceleration constant for predator. Pv_j^t and PP_j^t is predator 's velocity and position at t^{th} iteration, respectively. In the turbulent environment because of predator, the prey particles are forced to attain new positions, which help in exploitation. This chapter treats the predator effect on prey as a function of cognitive and social behavior of prey. This further is likely to be influenced by two factors: the individual's realization and the swarms realization of the predator 's position. The favorable experiences reinforce the decision, whereas unfavorable aspects have an inhibitory impact on the decision-making. Therefore, the cognitive and social fear of prey depends on *best* and *not-best* position of prey, resulting in continuous behavior guided perturbation of dynamic magnitude. Mathematically, velocity of prey particles is expressed by rewriting Eq. (2.21) as:

$$\begin{aligned} v_{ij}^{t+1} = & wv_{ij}^t + c_{11}rand()(P_{ij}^B - P_{ij}^t) + c_{12}rand()(g_j^B - P_{ij}^t) \\ & + c_{21}rand()(P_{ij}^t - P_{ij}^{NB}) + c_{22}rand()(P_{ij}^t - g_j^{NB}) \\ & + c_{31}rand()(ae^{-b\beta}) \quad (j = 1, 2, \dots, N_g; i = 1, 2, \dots, N_P) \end{aligned} \quad (2.26)$$

where $\beta = [w_1|P_{ij}^B - PP_j| + w_2|PP_j - P_{ij}^{NB}| + w_3|g_j^B - PP_j| + w_4|PP_j - g_j^{NB}| + w_5|P_{ij}^t - PP_j|]$ and $\sum_{k=1}^5 w_k = 1$. The weights are set giving equal weightage as $w_1 = w_2 = w_3 = w_4 = 0.125$ and $w_5 = 0.5$; Nevertheless, random variation may also be applied.

As and when convergence happens, the prey particles are nearer to predator and a behavior guided perturbation in the swarm is introduced. Further P_{ij}^{NB} and g_j^{NB} of i^{th} particle loses its dynamic behavior and becomes a static limit and controls the

perturbation resulting in better exploitation. On the other hand, when the prey particles are away from predator during the exploration process in initial iterations and the predator has a small influence on prey particles. Prey position of the swarm is updated as follows:

$$P_{ij}^{t+1} = P_{ij}^t + v_{ij}^{t+1} \quad (j = 1, 2, \dots, N_g; i = 1, 2, \dots, N_P) \quad (2.27)$$

In this work, maximum number of iterations is taken as the stopping criterion. If the stopping criterion is unmet, the procedure stated above is repeated. The Fig. 2.1 presents the flowchart representation of SPPO based solution approach.

2.5 EXPERIMENTAL RESULTS AND ANALYSIS

To prove the capability of the SPPO solution approach, the generalized benchmark test functions as well as standard real world problems related to power system operation have been undertaken. The SPPO algorithm has been implemented on Pentium-4, 80GB and 2 GHz system using Fortran Power Station 4.0.

2.5.1 Test Systems

In order to illustrate the effectiveness of the proposed methodology, the considered generalized benchmark problems that are uni-modal, multi-modal, separable, non-separable and discontinuous in nature are tabulated in Table 2.1. The Table 2.1 presents the search space of and optimal value of respective generalized functions which have been simulated to check the convergence speed and robustness. The details of undertaken functions are presented in Appendix C.

Further, the algorithm has been applied to solve the ELD problem of electric power system. The small generation capacity, electric power system consists of ten multi-fuel operated generators along with the additional complexities of power generators and transmission losses. Six different cases of this system are undertaken, which are designated as Case 1 to 6. The Case 1 has a quadratic cost function, the Case 2 has non-smooth cost function, whereas Case 3 has POZ and ramp rate constraints imposed on power gen-

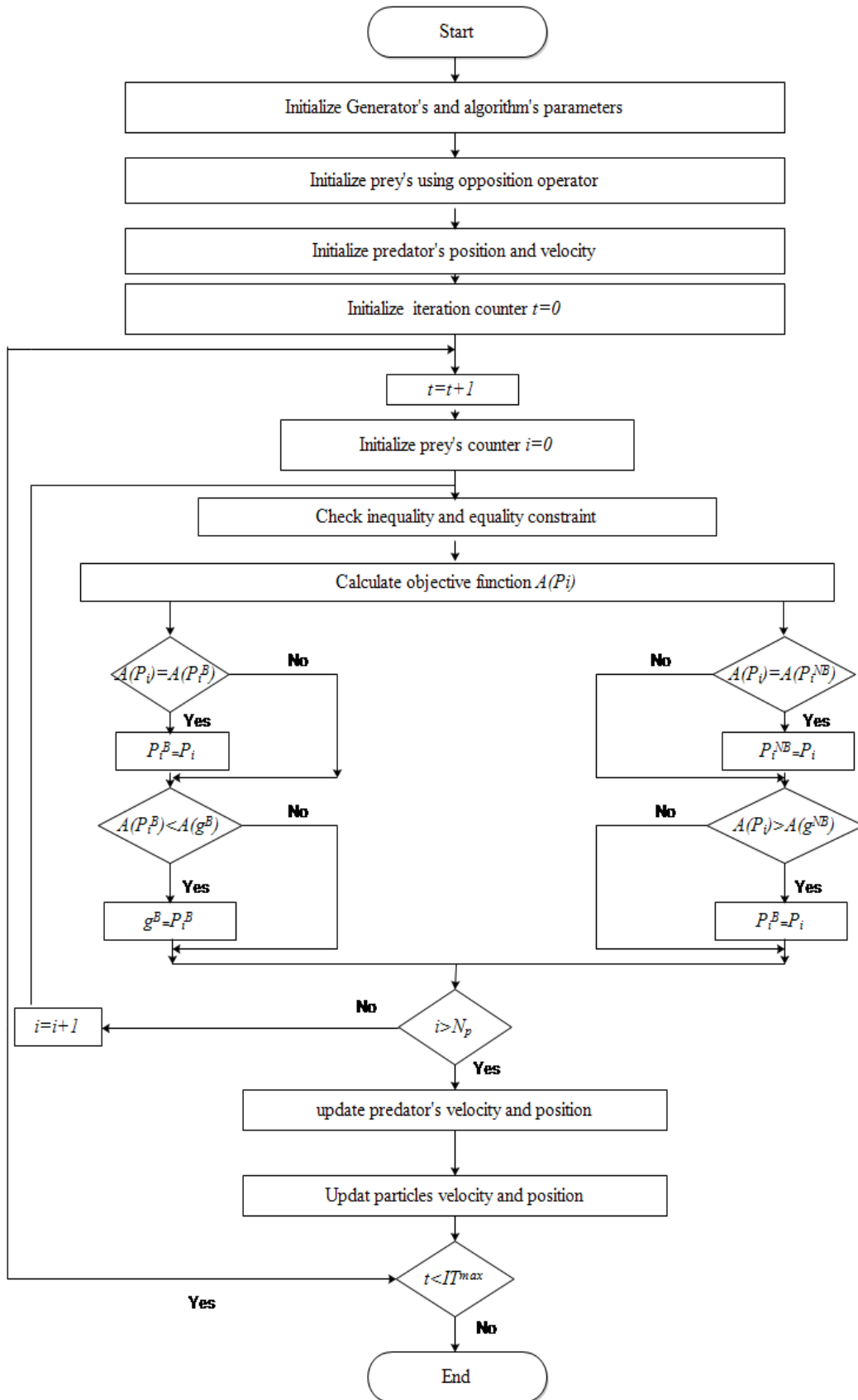


Figure 2.1: Synergic predator prey optimization

erators. The Case 4 to 6 includes transmission losses in addition to the complexities of Case 1 to 3 respectively. In order to introduce transmission losses to Test system-1, the *B-coefficients* are obtained from [330] and are tabulated in Appendix D.1. The medium capacity system has forty power generators with non-smooth behavior and its generator characteristics are shown in Appendix D.2. The large capacity test power system has 140 power generators and its power generator characteristics are presented in Appendix D.3. The undertaken electric power systems have different cases, which are classified according to additional complexity as shown in Table 2.2. The additional complexity is due to valve point loading effect, ramp rate limit constraint, POZ avoidance and transmission losses. The literature referred for system characteristics is also depicted in Table 2.2. The parameters of any global search technique affect the quality of solution, so the parameters of proposed technique are set after number of trails. The best parameters of the algorithm selected for various problems are tabulated in Table 2.3.

Table 2.1: Generalized bench mark functions considered for analysis of SPPO

	Test function	Search space	Global minima
F_1	Griewank function	[-600,600]	0.0
F_2	Rastrigin function	[-10, 10]	0.0
F_3	Rosenbrock function	[-30, 30]	0.0
F_4	Schwefel 2.22 function	[-10,10]	0.0
F_5	Sphere function	[-100, 100]	0.0
F_6	Step function	[-10, 10]	0.0
F_7	Step 2 function	[-10, 10]	0.0

2.5.2 Generalized Test Functions

In order to, test the efficiency of SPPO to obtain solution, the experiments are performed for 50 trials on every test problem. The experimental results for each algorithm on each test function are listed in Table 2.4. The graph, in Fig. 2.2 illustrates the evolution of the best fitness curve for different test functions, out of 50 trials. The curves present the convergence in initial 200 iterations, on \log_{10} scale. The Table 2.4 presents the best, worst and mean fitness value after 1000 iterations. It is observed that the SPPO is able to converge to the expected value in almost all the functions. In the worst performance case also the final fitness value is near to the expected optimal solution. For convenience

Table 2.2: Electrical power test systems undertaken for performance analysis

Test power system	Case	Power generators	Demand power (MW)	Valve point loading	Ramp rate	Prohibited operating zone	Transmission loss
1	1	10 [163]	2700	×	×	×	×
	2	Multi-fuel system		✓	×	×	×
	3			×	×	✓	×
	4			×	×	×	✓
	5			✓	×	✓	✓
	6			×	×	✓	✓
2	1	40 [330] Taiwan power system	10500	✓	×	×	✓
3	1	140 [321]	49342	×	✓	✓	×
	2	Korean power system		✓	✓	✓	×

Table 2.3: SPPO parameter setting

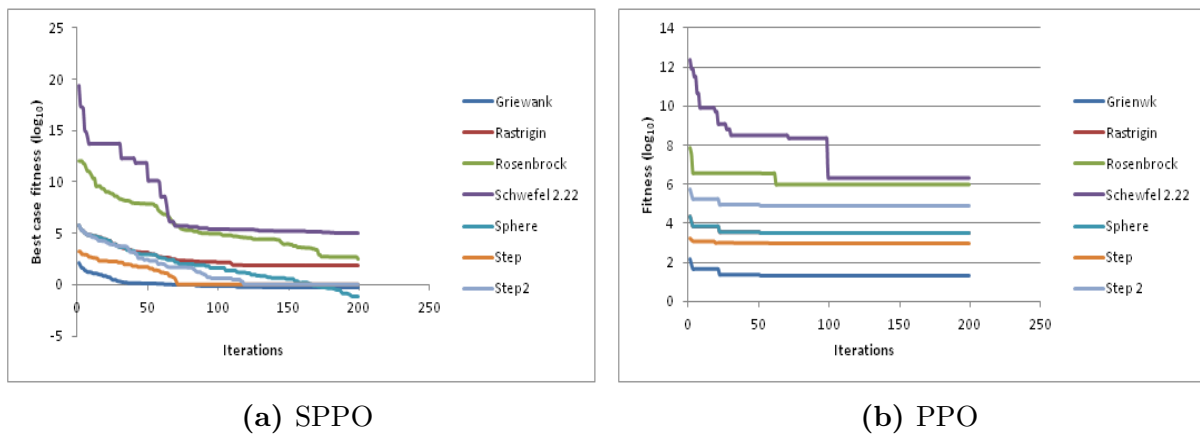
Parameter	Generalized test function	Test power system		
		1	2	3
a	0.001	0.001	0.01	0.01
b	1.0	1.0	1.0	1.0
P_f	0.5	0.5	0.001	0.5
N_p	10	10	50	50
IT^{max}	1000	50	100	1000

Note: $c_{11} = c_{12} = c_{22} = c_{31} = c_p = 2.0$

to show search ability, the Fig. 2.2a presents the convergence behavior of SPPO for 200 iterations, as the major improvement in solution in case of each test function is observed in this region. The convergence behavior shows that in case of Griewank (F_1), Rastrigin (F_2) and Sphere (F_5) functions the algorithm progresses swiftly, and achieves an optimal solution, whereas for Rosenbrock (F_3) and Schwefel (F_4) functions the SPPO is slow to converge, but achieves an optimal solution (Table 2.4) in 1000 iterations. The Fig. 2.2b illustrates the convergence behavior of basic PPO of undertaken benchmark problems. It can be clearly observed by comparing the Fig. 2.2a and 2.2b that for each function SPPO converges gradually, to find the optimum solution.

Table 2.4: Analysis of generalized test functions under investigation (10-D) [380]

Test function	Search space	Best fitness	Worst fitness	Mean fitness
Griewank function, F_1	[-600,600]	6.7×10^{-3}	1.0	2.3×10^{-2}
Rastrigin function, F_2	[-10, 10]	0.0	2.3×10^{-1}	5.6×10^{-2}
Rosenbrock function, F_3	[-30, 30]	4.5×10^{-1}	6.3	1.6
Schwefel 2.22 function, F_4	[-10,10]	0.0	0.0	0.0
Sphere function, F_5	[-100, 100]	0.0	0.0	0.0
Step function, F_6	[-10, 10]	0.0	0.0	0.0
Step 2 function, F_7	[-10, 10]	0.0	0.0	0.0

**Figure 2.2:** Generalized benchmark problems convergence analysis

2.5.3 Test Power System-1

The test power system-1 is a multi-fuel system with 10 generators. The comparison of power generation cost with other algorithms available in literature, for Case 1 to 3 has been illustrated in Table 2.5. The cost has been compared with *Ant Colony Optimization* (ACO), *Biogeography based optimization* (BBO) [306], *Composite Particle Swarm Optimization* (CCPSO) [386], *Conventional Genetic Algorithm with Mutation Update* (CGA-MU) [163], *Differential Evolution* (DE) [373], *Enhanced Augmented Hopfield Neural Network* (ELHN) [369], *Improved Genetic Algorithm* (IGA) [163], *Improved Genetic Algorithm with multiplier update* (IGA-MU) [163], *Improved Particle Swarm Optimization* (IPSO) [373], *Real Coded Genetic Algorithm* (RCGA) [386], *Seeker Based Optimization Algorithm* (SBOA) [386], *Quadratic Programming Augmented Hopfield Neural Network* (QP-ALHN) [376], *Particle Swarm Optimization* (PSO) [373], *Hybrid of DE and PSO* (DEPSO) [386], *Hybrid of DE and BBO* (DEBBO) [386], *Krill herd optimization* (KHA) [397]. The comparison shows that the cost obtained by SPPO for both the cases-1 and

Table 2.5: Test power system-1, ELD cost comparison ($P_D=2700\text{MW}$)

Algorithm	Cost(\$/h)		
	Case-1	Case-2	Case-3
ACO[373]	–	623.7, (624.193)a	–
BBO [306]	624.5178	–	–
CCPSO[386]	–	623.82	–
CGA-MU [373]	623.8095	624.7193	–
DE[373]	623.8098	624.4606	–
DEBBO [305]	624.5178	–	–
DEPSO[386]	–	623.83, (624.5283)a	–
ELHN [369]	624.5178	–	–
IGA [163]	624.5178	–	–
IGA-MU [163]	623.8093	624.5178	–
IPSO[373]	623.8089, (623.80910)a	623.8730	–
KHA [397]	624.5178	–	–
PSO[373]	623.8094	624.2449	–
QP-ALHN [376]	623.809	–	624.3212
RCGA [386]	–	623.83, (624.52)a	–
SBOA [386]	–	598.16, (686.273)a	–
SPPO	623.809	623.8279	624.3210

Note: a Actual cost calculated from given power distribution

Table 2.6: Test power system-1, ELD Power distribution plan by SPPO ($P_D=2700\text{ MW}$)

Generator	Case-1		Case-2		Case-3	
	Fuel type	Power (MW)	Fuel type	Power (MW)	Fuel type	Power (MW)
P_1	2	218.3507	2	218.1014	2	221.999
P_2	1	211.5895	1	209.9070	1	212.828
P_3	1	280.7704	1	280.6587	1	284.275
P_4	3	239.679	3	239.8207	3	240.462
P_5	1	278.4467	1	279.9406	1	259.999
P_6	3	239.5304	3	239.5267	3	240.503
P_7	1	288.5161	1	287.6992	1	293.308
P_8	3	239.6159	3	240.0893	3	240.515
P_9	3	428.6269	3	426.3904	3	437.115
P_{10}	1	274.8742	1	275.8660	1	269.998

Table 2.7: Test power system-1, ELD Power distribution plan by SPPO ($P_D = 2700MW$)

Generator	Case-4		Case-5		Case-6	
	Fuel type	Power (MW)	Fuel type	Power (MW)	Fuel type	Power (MW)
P_1	2	229.708	2	223.204	2	226.340
P_2	1	222.829	1	220.878	1	216.586
P_3	1	304.310	1	311.486	1	293.766
P_4	3	240.348	3	242.933	3	243.282
P_5	1	316.169	1	311.179	1	335.00
P_6	3	246.015	3	240.267	3	243.474
P_7	1	317.186	1	314.904	1	305.201
P_8	3	236.998	3	240.222	3	242.982
P_9	3	432.065	3	434.135	3	439.999
P_{10}	1	295.962	1	302.488	1	295.005
Cost(\$/h)	700.296		700.776		700.482	
P_L (MW)	141.593		141.642		141.635	

3, is less than the cost obtained by other algorithms. However, the cost obtained for case-2 by SPPO is better than others, while the results are competitive with CCPSO, as proposed method gives 9.4×10^{-3} \$/h higher operating cost, which is very small difference. The obtained optimal generation schedule is tabulated in Table 2.6. Table 2.7 presents optimal generation schedule for cases 4-6. The convergence characteristic comparison with other principally similar techniques such as PPO and APSO has been performed using test power system-1, case-6 as shown in Fig. 2.3. The convergence plot shows that in the initial phase, the SPPO performance is near to other algorithms, but after a few iterations, SPPO swiftly moves towards the optimal solutions at a uniform pace under the influence of collective intelligence based mechanism. The results obtained by SPPO are better with lower operating cost than the other applied techniques.

2.5.4 Test Power System-2

The test power system 2 shown in Table 2.2 is a medium size power system problem having 40 generators. The comparison of results obtained by SPPO has been performed with *Classical Evolutionary Programming* (CEP) [135], *Differential Evolution* (DE) [193], *Differential Evolution Sequential Quadratic Programming* (DE-SQP) [193], *Fast Evolutionary Programming* (FEP) [135], *hybrid of PSO with local random search* (PSO-LRS) [227], *hybrid CEP-PSO* (CEP-PSO) [135], *Improved Fast Evolutionary Programming* (IFEP) [135], *Multiobjective Differential Evolution* (MODE) [330], *Modified Particle Swarm Op-*

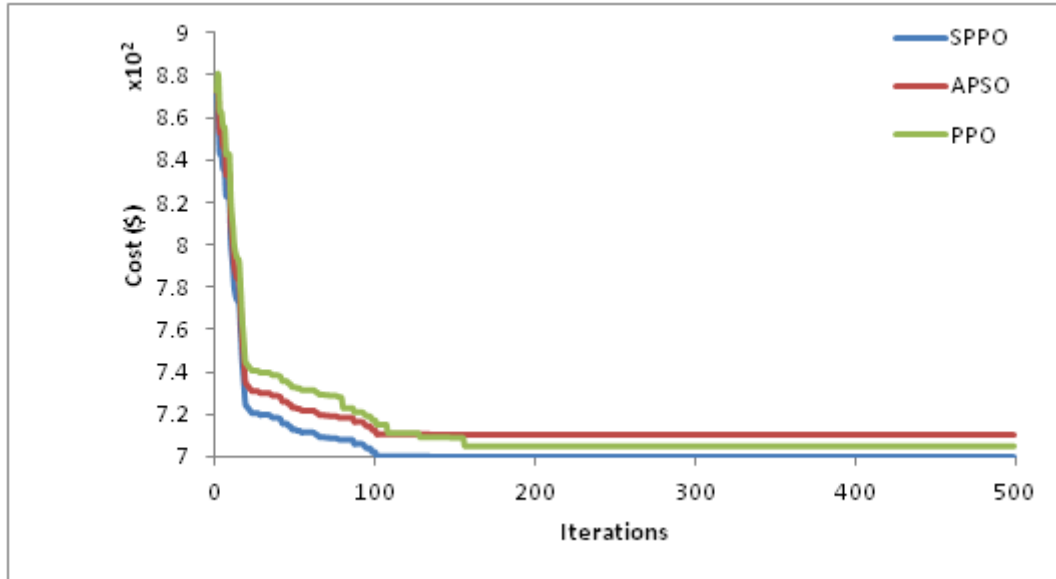


Figure 2.3: Convergence characteristics of Test power system-1, case-2

timization (MPSO) [176], *New Particle Swarm Optimization* (NPSO) [227], *New Particle Swarm Optimization with Local Random Search* (NPSO-LRS) [227], *Particle Swarm Optimization* (PSO) [135], *Quasi Teacher Learner Based Optimization* (QTLBO) [384], *Real Coded Chemical Reaction Optimization* (RCCRO) [394], *Teacher Learner Based Optimization* (TLBO) [384], and *Predator Prey Optimization* (PPO). The comparison of cost of power generation has been presented in Table 2.8. The cost comparison shows that the SPPO has obtained a solution with lesser power generation cost. The corresponding power generation schedule is presented in Table 2.9.

2.5.5 Test Power System-3

The test power system-3 mentioned in Table 2.1 has 140 generators with ramp rate limits and POZ constraints. The Case-1 of this system has been undertaken for study and corresponding power generator characteristics are referred from Appendix D.3. The comparison of the cost of power schedule obtained by SPPO with other algorithms available in literature is shown in Table 2.10. The results are compared with *Anti Predator Swarm Optimization* (APSO), *Composite Particle Swarm Optimization* (CCPSO) [321], *Continuous Quick Group Search Optimizer* (CQGSO) [360], *Differential Harmony Search* (DHS) [388], *Predator Prey Optimization* (PPO). The comparison with aforementioned techniques shows the better quality of obtained solution than CCPSO, DHE, PPO, APSO.

Table 2.8: Test power system-2, ELD cost comparison ($P_D = 10500MW$)

Algorithm	Cost (x10 ⁵ \$/h)
CEP [135]	1.23500
CEP-PSO [152]	1.23670
DE [193]	1.21900
DE-SQP [193]	1.21741
FEP [135]	1.24500
IFEP [135]	1.23000
MODE [330]	1.21840
MPSO [176]	1.22252
NPSO [227]	1.21704
NPSO-LRS [227]	1.21664
PSO [152]	1.26848
PSO-LRS [227]	1.21704
PPO	1.21413
QTLBO [384]	1.21428
RCCRO [394]	1.21412
TLBO [384]	1.21685
SPPO	1.21402

Table 2.9: Test power system 2, ELD power distribution plan by SPPO ($P_D = 10500MW$)

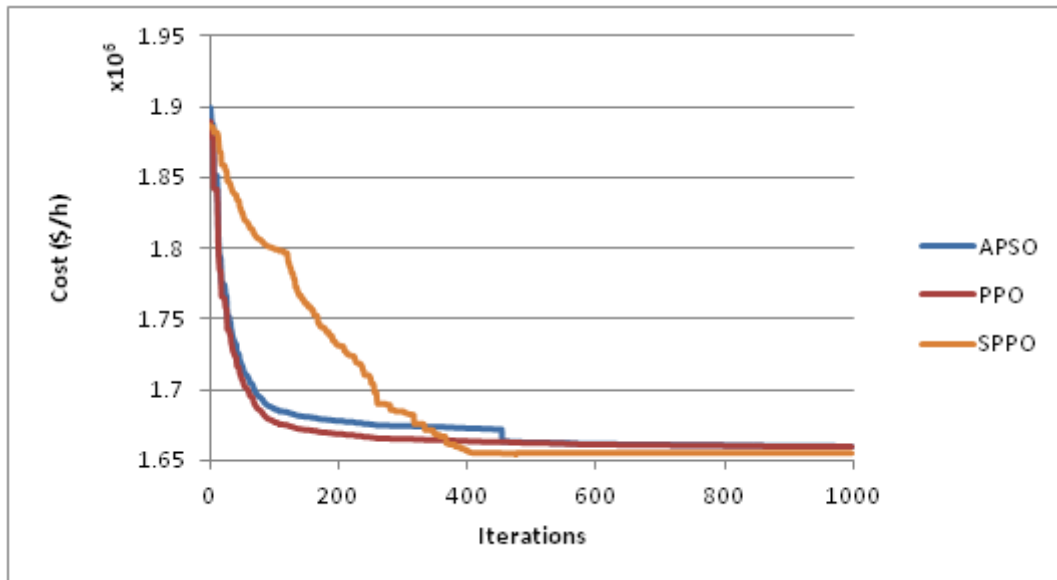
Unit	Power	Unit	Power	Unit	Power	Unit	Power
P_1	111.87	P_{11}	168.71	P_{21}	523.28	P_{31}	190.0
P_2	110.86	P_{12}	94	P_{22}	523.54	P_{32}	190.0
P_3	97.59	P_{13}	215.09	P_{23}	523.31	P_{33}	190.0
P_4	179.52	P_{14}	394.40	P_{24}	523.21	P_{34}	165.34
P_5	88.91	P_{15}	394.27	P_{25}	523.42	P_{35}	200.0
P_6	140.0	P_{16}	304.63	P_{26}	523.31	P_{36}	200.0
P_7	260.80	P_{17}	489.37	P_{27}	10.0	P_{37}	110.0
P_8	284.61	P_{18}	489.60	P_{28}	10.0	P_{38}	110.0
P_9	284.92	P_{19}	511.29	P_{29}	10.0	P_{39}	110.0
P_{10}	130.0	P_{20}	511.64	P_{30}	90.96	P_{40}	511.36

Table 2.10: Test system-3, ELD cost comparison

Algorithm	Cost (\$/h)	
	Case-1	Case-2
APSO	1655684.5	1657963.0
CCPSO [321]	1655685.0	1657962.75
CQGSO [360]	1655679.42	1657962.74
DHS [388]	–	1657944.86
PPO	1655684.0	1657963.5
SPPO	1655680.0	1657962.0

The solution obtained by SPPO is competitive with solution of CQGSO as there is difference in obtained operating cost is 0.58 \$/h, which is insignificant. The Table 2.11 depicts corresponding generation schedule. The cost versus iteration plot of SPPO shown in Fig. 2.4, exhibits steady convergence and shows supremacy to obtain better optimal solution in less number of iterations. Further, from the frequency distribution comparison in Fig. 2.5 by performing 30 independent trial runs, it is evident that for most of the random independent trials, the SPPO algorithm was able to obtain the solution in the range near optimal solution as compared to others.

2.5.6 Statistical Performance Analysis of SPPO Algorithm

**Figure 2.4:** Convergence characteristics comparison test power system-3

1. Parameter sensitivity: To investigate the effect of parameter variation and dependency on initial generated solution, parameter sensitivity and robustness of SPPO

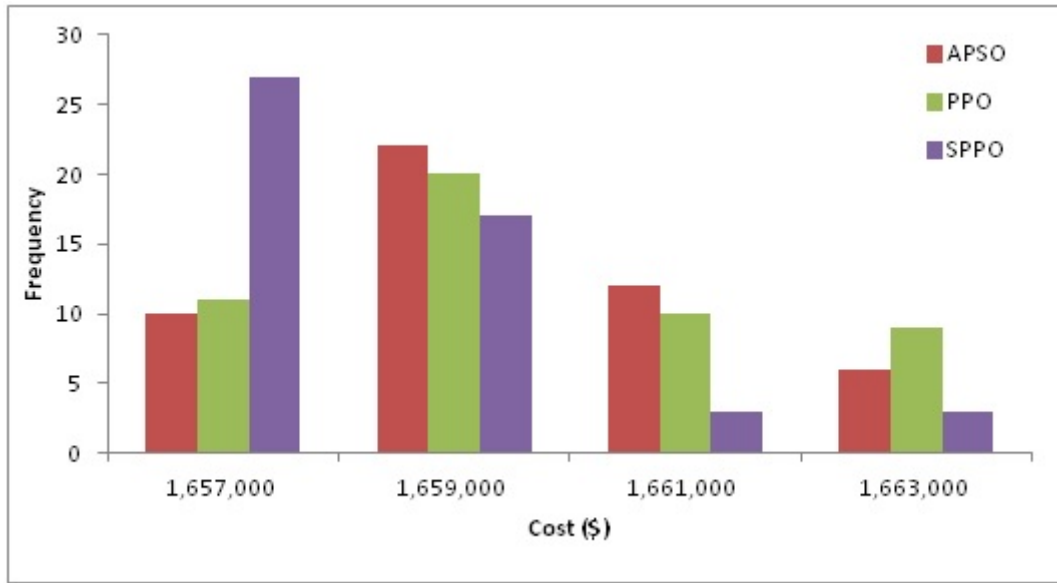


Figure 2.5: Frequency distribution of test power system-3

Table 2.11: Test system-3, Optimal power generation schedule (MW) using SPPO

Unit	Case-1	Case-2	Unit	Case-1	Case-2	Unit	Case-1	Case-2	Unit	Case-1	Case-2
P_1	118.988	119.0	P_{36}	500.0	500.0	P_{71}	142.640	138.7773	P_{106}	880.9	880.9
P_2	164.0	164.0	P_{37}	241.0	241.0	P_{72}	368.612	388.3275	P_{107}	873.7	873.7
P_3	190.0	190.0	P_{38}	241.0	241.0	P_{73}	195.0	195.0	P_{108}	877.4	877.4
P_4	190.0	190.0	P_{39}	774.0	774.0	P_{74}	219.608	187.5903	P_{109}	871.7	871.7
P_5	190.0	168.45	P_{40}	769.0	769.0	P_{75}	217.556	203.9351	P_{110}	864.8	864.8
P_6	190.0	190.0	P_{41}	3.0	3.0	P_{76}	267.225	265.8373	P_{111}	882.0	882.0
P_7	490.0	490.0	P_{42}	3.0	3.0	P_{77}	389.802	399.3057	P_{112}	94.0	94.0
P_8	490.0	490.0	P_{43}	250.0	250.0	P_{78}	330.0	330.0	P_{113}	94.0	94.0
P_9	496.0	496.0	P_{44}	250.0	250.0	P_{79}	531.0	531.0	P_{114}	94.0	94.0
P_{10}	496.0	496.0	P_{45}	250.0	250.0	P_{80}	531.0	531.0	P_{115}	244.0	244.0
P_{11}	496.0	496.0	P_{46}	250.0	250.0	P_{81}	542.0	542.0	P_{116}	244.0	244.0
P_{12}	496.0	496.0	P_{47}	250.0	250.0	P_{82}	56.0	56.0	P_{117}	244.0	244.0
P_{13}	506.0	506.0	P_{48}	250.0	250.0	P_{83}	115.0	115.0	P_{118}	95.0	95.0
P_{14}	509.0	509.0	P_{49}	249.99	250.0	P_{84}	115.0	115.0	P_{119}	95.0	95.0
P_{15}	506.0	506.0	P_{50}	250.0	250.0	P_{85}	115.0	115.0	P_{120}	116.0	116.0
P_{16}	505.0	505.0	P_{51}	165.08	165.0	P_{86}	207.0	207.0	P_{121}	175.0	175.0
P_{17}	506.0	506.0	P_{52}	165.0	165.0	P_{87}	207.0	207.0	P_{122}	2.0	2.0
P_{18}	506.0	506.0	P_{53}	165.0	165.0	P_{88}	175.0	175.0	P_{123}	4.0	4.0
P_{19}	505.0	505.0	P_{54}	165.0	165.0	P_{89}	175.0	175.0	P_{124}	15.0	15.0
P_{20}	505.0	505.0	P_{55}	180.0	180.0	P_{90}	178.63	176.93	P_{125}	9.0	9.0
P_{21}	505.0	505.0	P_{56}	180.0	180.0	P_{91}	175.0	175.0	P_{126}	12.0	12.0
P_{22}	505.0	505.0	P_{57}	103.0	103.0	P_{92}	575.4	575.4	P_{127}	10.0	10.0
P_{23}	505.0	505.0	P_{58}	198.0	198.0	P_{93}	547.5	547.5	P_{128}	112.0	112.0
P_{24}	505.0	505.0	P_{59}	312.0	312.0	P_{94}	836.8	836.8	P_{129}	4.0	4.0
P_{25}	537.0	537.0	P_{60}	311.25	307.68	P_{95}	837.5	837.5	P_{130}	5.0	5.0
P_{26}	537.0	537.0	P_{61}	163.0	163.0	P_{96}	682.0	682.0	P_{131}	5.0	5.0
P_{27}	549.0	549.0	P_{62}	95.0	95.0	P_{97}	720.0	720.0	P_{132}	50.0	50.0
P_{28}	549.0	549.0	P_{63}	511.0	511.0	P_{98}	718.0	718.0	P_{133}	5.0	5.0
P_{29}	501.0	501.0	P_{64}	511.0	511.0	P_{99}	720.0	720.0	P_{134}	42.0	42.0
P_{30}	499.0	499.0	P_{65}	490.0	490.0	P_{100}	964.0	964.0	P_{135}	42.0	42.0
P_{31}	506.0	506.0	P_{66}	252.99	258.51	P_{101}	958.0	958.0	P_{136}	41.0	41.0
P_{32}	506.0	506.0	P_{67}	490.0	490.0	P_{102}	947.9	947.9	P_{137}	17.0	17.0
P_{33}	506.0	506.0	P_{68}	490.0	490.0	P_{103}	934.0	934.0	P_{138}	7.0	7.0
P_{34}	506.0	506.0	P_{69}	130.0	130.0	P_{104}	935.0	935.0	P_{139}	7.0	7.0
P_{35}	500.0	500.0	P_{70}	296.46	339.44	P_{105}	876.5	876.5	P_{140}	26.01	26.0

Table 2.12: Analysis of the parameter sensitivity (Test power System-3, Case-1)

Parameter	Operating Cost (\$/h)						
	Scope	Step	Mean	Minimum	Maximum	S.D.	variation (%)
c_{11}	0-4.0	0.2	1662744.0	1655680.0	1667609.2	1032.7	0.72%
c_{12}	0-4.0	0.2	1662695.5	1655680.0	1667710.0	1532.1	0.72%
c_{21}	0-4.0	0.2	1658835.3	1655680.0	1659990.5	2032.8	0.26%
c_{22}	0-4.0	0.2	1662286.35	1655680.0	1666892.7	2055.1	0.67%
c_{31}	0-4.0	0.2	1662792.9	1655680.0	1667905.8	2322.2	0.73%
c_P	0-4.0	0.2	1663985.45	1655680.0	1668290.9	1372.3	0.76%
a	0-0.2	0.002	1658450.0	1655680.0	1659220.0	1566.6	0.21%
b	0-2.0	0.01	1658445.0	1655680.0	1659210.0	3632.5	0.21%
P_f	0-0.1	0.01	1668756.0	1655680.0	1679832.0	4057.0	1.45%

Table 2.13: Test power system-3, case-1, Descriptive statistics

Algorithm	Operating Cost (\$/h)			
	Mean	Minimum	Maximum	S.D.
PPO	1658200.9	1655684.0	1667800.0	2067.12
APSO	1657684.5	1655684.5	1667800.0	3343.51
SPPO	1657265.04	1655680.0	1667800.0	2872.80

Table 2.14: Student's T-test for sample mean ($\alpha = 0.05$)

Statistics	Test power system-1	Test power system-3
P-value	0.282924	0.974071
t-calculated	1.09482	0.032784
t-critical	2.048407	1.699127

Table 2.15: One-sample Wilcoxon signed rank test ($\alpha = 0.05$)

Statistics	Test power system-1	Test power system-3
P-value	0.069967	0.061143
z-score	1.812121	1.872462
T-critical	23.8986	136.7475

algorithm has been tested on test power system-3. Table 2.12 shows the experimental study undertaken for parameter sensitivity analysis. The various parameters of the SPPO are varied by 2% of the values chosen for investigation. From the observation table, it is observed that the obtained solution varies in the small range around the mean value and relative deviation from minimum operating cost is less than 1%. An almost similar trend has been observed in statistical analysis of other investigated systems.

2. Robustness test: In order to verify the capability of the proposed SPPO approach to obtain solution thirty independent trial runs are performed on Test power systems 3, case-1. The standard deviation of the obtained solution is computed for independent trial runs. The descriptive statistics are tabulated in Table 2.13. It is observed from the statistics that the operating cost obtained by proposed technique shows minimum standard deviation, so the results obtained by proposed technique have high precision as compared to others. Although the standard deviation for the proposed approach is least in comparison to others. As, its magnitude is significant. Therefore, student's T-test (one sample) two tailed is applied to test the equality of sample mean and population mean at 5% confidence level. The test statistics are tabulated in Table 2.14. As the p-value is greater than 0.05, there is very little evidence to reject the null hypothesis, hence the population mean and sample mean are nearly equal. Similarly, the t-test also validates the solutions of test system-3, as shown in Table 2.14, result statistics.

3. Wilcoxon signed rank test: In order to study the performance of the proposed algorithm SPPO statistically, the simulation experiments are performed for 50 observations, with different initialization. The one-sample Wilcoxon signed rank test has been applied with significance level of 5% to test power system-1, case-1. The results of the test presented in Table 2.14 proves that there is no evidence to reject null hypothesis. Thus, the obtained observations are distributed symmetrically about the median.

2.6 CONCLUSIONS

In this chapter, an optimization algorithm synergic predator prey optimization that is a modification of the basic behavior of predator prey optimization has been presented. The proposed algorithm splits intelligence of prey to best and not-best components and fear factor of prey has been treated as a collaborative function of cognitive and social behavior of prey particle. The proposed algorithm has been presented as an efficient search technique for solving ELD. Moreover, proposed method ensures diversity of solutions during the search and explores the search space effectively. The operating cost functions of generating units are non-smooth and involve complexity because of additional constraints like multi-fuel options, ramp rate limits, and prohibited zones. The applied approach ensures that prey guiding components do not stay at a particular position for a large number of iterations. Computer simulations are performed to verify the effectiveness of SPPO. The numerical results reveal the better practicability, solution accuracy and mature convergence capability of SPPO algorithm. The results of synergic predator prey optimization are competitive with composite particle swarm optimization and continuous quick group search algorithm. Overall, the SPPO emerges out as an alternative solution procedure for economic load dispatch problem's solution.

CHAPTER 3

MULTIOBJECTIVE LOAD DISPATCH USING CHAOTIC EVOLUTIONARY PROGRAMMING AND POWELL'S METHOD

3.1 INTRODUCTION

The practical problems related to electrical power generation scheduling and planning are highly constrained and multiobjective in nature. In order to obtain the solution of these problems population based algorithms are widely accepted. These techniques are widely accepted, due to their derivative free mechanism, parallel processing nature, fast convergence rate, performance independent of hyperspace *etc.* In the recent years, *Genetic Algorithm* (GA), *Evolutionary Programming* (EP), *Simulated Annealing* (SA), *Particle Swarm Optimization* (PSO) are the most preferred solution approaches in diverse fields. Among the various application fields, power system is a prominent field. *Economic Load Dispatch* (ELD), *Minimum emission dispatch* (MED) and *Multiobjective load dispatch* (MOLD) is an important power system optimization task with an objective to allocate generation among committed units in such a way that the system constraints are satisfied and power is available at the minimum generation cost as well as minimum pollutant emission. The input-output characteristics of modern units are highly non-linear due to ramp rate limits, valve point loading effects *etc.* These problems are multi-modal in nature and the classical approaches such as gradient approaches are unable to address the problems satisfactorily.

In this respect, stochastic search algorithms such as GA [52], ES [177], EP [67], SA

[44], PSO [126] *etc.* have been proved effective in solving nonlinear ELD problems without any restrictions on the shape of the cost curves. Although these heuristic methods do not always guarantee discovering the globally optimal solution in finite time, however provide a fast and reasonable solution, which may be a suboptimal or near-globally optimal solution.

EAs have more flexibility and robustness as compared to conventional calculus-based methods. Among various EAs, EP has dominated the field of optimization in the decade of the nineties. The EP is a little bit different from other EA's as: firstly, it requires the real valued control parameters, secondly, EP's behavior is guided by mutation and selection, *i.e.* EP uses single evolution operator. Therefore, EP uses lesser computational resources and may result in smaller computational time. The field of EP has matured through development of classical EP [69], self adaptive EP [47], fast EP [68], scaled Gaussian mutation EP [67], EP with the mean of Gaussian and Cauchy mutations with empirical learning rate [69], EP with Gaussian mutation with empirical learning [34], EP with the better of Gaussian and Cauchy mutations with empirical learning rate [91]. All these modifications have been proposed to improve the limitation of EP, that it is slow to converge for multi-modal functions [91]. Further, a fast EP algorithm based on Levy probability distribution based mutation operator has been proposed, thereby resulting in better performance than CEP [146]. Liu *et al.* have shown empirically that EP with cooperative convolution can speed up convergence rates on the large-scale problems [105]. To summarize the literature survey, it has been observed that the modifications in the EP are primarily focused on mutation operator. In particular, these alterations are random number oriented.

Recently, the application of non-linear dynamics has drawn attention in the field of optimization. One of the application is the use of chaos map to select algorithm control parameters [368], tune the parameters of search algorithms [194, 340]. The generation of random sequences with a long period and good uniformity is very important for a stochastic algorithm. Chaotic map is a random number generation function. It is ergodic, non-repetitive, non-linear, and dynamic in nature [434]. Although, the nature of chaos is apparently random, it also possesses an element of regularity, exhibits sensitive dependence on initial conditions and includes infinite unstable periodic motions in nonlin-

ear systems [406]. Various chaotic maps such as tent map, logistic map, Chebyshev map, and iterative chaotic map with infinite collapses map have been reported in the literature. Since different chaotic maps lead to diverse algorithm behavior, therefore chaos based methods are potential alternatives to pseudo random sequences. In almost all stochastic meta-heuristic algorithms randomness is achieved by using probability distributions, often uniform or Gaussian. In principle, it can be advantageous to replace such randomness by chaotic maps because chaos has very similar properties of randomness with better statistical and dynamical properties. Such dynamical mixing is important to ensure that algorithm generates diverse solutions and potentially explores the multi-modal objective landscape. Due to its properties, the algorithms can potentially carry out iterative search steps at higher speeds than standard stochastic search with standard probability distributions [435].

On the basis of the above facts and arguments, it is required to empirically investigate the performance of the EP by incorporating chaotic sequence and local search technique. Therefore, the objective of this chapter is to introduce and study chaotic sequence based EP (CEP). In order to, accomplish the above mentioned tasks, firstly, a chaotic random number based mutation operator is proposed. Secondly, a chaotic sequence guided tournament selection operator is implemented. Further, the Powells pattern search is applied to enhance the exploitation of proposed solution procedure. The resultant algorithm is called *Chaotic Evolutionary Programming Pattern Search* (CEPPS). The performance of CEPPS algorithm is analyzed for solution optimality using standard benchmark test function and hence implemented to solve MOLD problem. The conflicting objectives are handled by the surrogate worth trade off method. The case studies of electrical power system networks having small, medium and high power generation capacity are simulated. At the same time, to give a naturalistic aspect of load dispatch problem, multi-objective case of multi-fuel system and 140-units Korean system are simulated. Multi-fuel system considers transmission losses along with non-linearity due to valve point loading, ramp rate limits and *Prohibited Operating Zone* (POZ). The results obtained by proposed solution approach are compared with the results of available state-of-the-art algorithms.

The chapter is organized into five parts. Section 3.2, presents the mathematical foun-

dation of multi-objective load dispatch problem. Section 3.3 discuss conceptual details of CEPPS algorithm. Section 3.4 presents the CEPPS based solution approach to solve *Multiobjective load dispatch* (MOLD) problem is conversed. Section 3.5, presents numerical results by the proposed method and compare it with some of the most recently published work. Finally, Section 3.6 concludes the chapter.

3.2 MULTI-OBJECTIVE LOAD DISPATCH

The MOLD problem aims to allocate committed thermal generators total load demand in such a way that the two conflicting objectives *viz.* operating cost and pollutants' emission are minimized simultaneously, in the feasible space. In case of multi-fuel option, the power generators have alternative fuels and each unit is represented by several piecewise, quadratic functions of fuel. Mathematically, MOLD problem is expressed as:

Minimize operating cost [121]

$$f_1(P_j) = \sum_{j=1}^{N_g} (a1_{jk}P_j^2 + b1_{jk}P_j + c1_{jk} + |d1_{jk} \sin\{e1_{jk}(P_{jk}^{min} - P_j)\}|) \quad (P_{jk}^{min} \leq P_j \leq P_{jk}^{max}) \quad (3.1)$$

Minimize pollutant emission [121]

$$f_2(P_j) = \sum_{j=1}^{N_g} (a2_{jk}P_j^2 + b2_{jk}P_j + c2_{jk} + d2_{jk} \exp(e2_{jk}P_j)) \quad (P_{jk}^{min} \leq P_j \leq P_{jk}^{max}) \quad (3.2)$$

where P_j is the generated real power and $P = [P_1, P_2, \dots, P_{N_g}]^T$; N_g is the number of generators; $a1_{jk}$, $b1_{jk}$, $c1_{jk}$, $d1_{jk}$ and $e1_{jk}$ are thermal generators' cost coefficients of j^{th} generator's k^{th} fuel option; P_{jk}^{min} and P_{jk}^{max} are the generator's lower and upper limits for k^{th} fuel option; $a2_{jk}$, $b2_{jk}$, $c2_{jk}$, $d2_{jk}$ and $e2_{jk}$ are pollutant emission coefficient of j^{th} thermal generators for k^{th} fuel option.

subjected to:

(i) Power balance equality constraint [463]

$$\sum_{j=1}^{N_g} P_j - (P_D + P_L) = 0 \quad (3.3)$$

(ii) Generator operating limits [463]

$$P_j^{\min} \leq P_j \leq P_j^{\max} \quad (j = 1, 2, \dots, N_g) \quad (3.4)$$

(iii) Ramp rate limit [126]

- As generation increases

$$P_j - P_j^0 \leq UR_j \quad (j = 1, 2, \dots, N_g) \quad (3.5)$$

- As generation decreases

$$P_j^0 - P_j \leq DR_j \quad (j = 1, 2, \dots, N_g) \quad (3.6)$$

(iv) Prohibited operating zone constraint [126]

$$P_j^{\min} \leq P_j \leq P_{j,1}^L \quad (j = 1, 2, \dots, N_g) \quad (3.7)$$

$$P_{j,i-1}^U \leq P_j \leq P_{j,i}^L \quad (i = 1, 2, \dots, N_{zj}; j = 1, 2, \dots, N_g) \quad (3.8)$$

$$P_{j,N_{zj}}^U \leq P_j \leq P_j^{\max} \quad (j = 1, 2, \dots, N_g) \quad (3.9)$$

where P_D and P_L are the fore-casted demand and transmission loss of the network respectively; N_{zj} is number of prohibited zones of the j^{th} generator; P_j^0 is the previous generated power; UR_j and DR_j are the up-ramp limit and down-ramp limit of the j^{th} generator; $P_{j,i}^L$ and $P_{j,i}^U$ are lower and upper range of i^{th} prohibited zone respectively of j^{th} generator.

3.3 PROPOSED HYBRID SEARCH ALGORITHM

The proposed solution approach clubs *Chaotic Evolutionary Programming* (CEP) approach with *Powell's pattern search* (PS), to optimize the generation schedule of MOLD problem. The CEP aims to explore the global solution, whereas PS focuses on exploitation of search area. The PS phase remains normally in hibernation and is activated after a prefixed number of iterations. These aspects are explained in the ensuing subsections:

3.3.1 Chaotic Evolutionary Programming

The principle of EP is to generate offspring and select better quality solution among offspring and existing parents. The mutation is the only evolutionary operator operating on parent population and the selection operation selects parents for the next generation from the existing pool of parents and offspring. In CEP the random numbers are generated by randomly initialized preselected chaotic sequence. The well accepted chaotic map, namely Gauss map and Tent map are used for randomization in the search process. So, in a N_g dimensional search space, an i^{th} individual vector (p_i, η_i) is a possible solution of the problem, where P_i is the objective variable and η_i is the associated strategy parameter. Mathematically, the evolution concept of chaotic EP to generate offspring is illustrated as follows:

$$P_{ij}^{t+1} = P_{ij}^t + \eta_{ij} \chi_j(0, 1) \quad (j = 1, 2, \dots, N_g, i = 1, 2, \dots, N_P) \quad (3.10)$$

$$\eta_{ij}^{t+1} = \eta_{ij}^t \exp(\tau^{t+1} \chi(0, 1) + \tau \chi_j(0, 1)) \quad (j = 1, 2, \dots, N_g, i = 1, 2, \dots, N_P) \quad (3.11)$$

where $\chi_j(0, 1) \in [0, 1]$ is a chaotic process generated random number, generated for j^{th} individual. The parameters τ and τ^{t+1} are given by $\left(\sqrt{2\sqrt{N_g}}\right)^{-1}$ and $(\sqrt{2N_g})^{-1}$ respectively. A Gauss map based chaotic sequence is represented as:

$$\chi_{n+1} = \begin{cases} 0 & ; \chi_n < 0 \\ 1/\chi_n & ; otherwise \end{cases} \quad (3.12)$$

and Tent map chaotic sequence follows:

$$\chi_{n+1} = \begin{cases} \chi_n/0.7 & ; \chi_n < 0.7 \\ 10/3\chi_n(\chi_n - 1) & ; otherwise \end{cases} \quad (3.13)$$

Once, the offspring are generated at t^{th} generation, each individual of combined population (parent and offspring) compete with others for survival in $t + 1^{th}$ generation. For selection an individual's score w_i in stochastic competition is given by

$$w_i = \sum_{n=1}^{N_P} w_n \quad (i = 1, 2, \dots, N_P) \quad (3.14)$$

with

$$w_n = \begin{cases} 1 & ; \chi_1 < \frac{f_m}{f_i + f_t} \\ 0 & ; otherwise \end{cases} \quad (3.15)$$

where f_m is the fitness of m^{th} randomly selected competitor in combined population; $m = int(2L\chi_2 + 1)$, f_i is the fitness value of i^{th} individual, $\chi_1, \chi_2 \in [0, 1]$ are random numbers generated by chaotic sequence.

3.3.2 Powell's Pattern Search

Powell's method is a direct numerical search technique to find the solution of optimization problem. The move from one point to another will be successful if the solution quality improves. The pattern move is to locate a solution of problem in already ascertained direction. The pattern move accelerates the search space exploitation to enhance convergence speed. These steps are elaborated as follows: Initially linearly independent search directions S , of order N_g are created as

$$S_{ij} = \begin{cases} 1 & ; i = j \\ 0 & ; otherwise \end{cases} \quad (i = 1, 2, \dots, N_g, j = 1, 2, \dots, N_g) \quad (3.16)$$

and a unidirectional search is performed in each direction considering P_i as a best point as follows:

$$P_j = \begin{cases} P_j + \lambda_i^* S_{ij} & ; f(P_j + \lambda_i^* S_{ij}) < f(P_j) \\ P_j & ; otherwise \end{cases} \quad (i = 1, 2, \dots, N_g) \quad (3.17)$$

where $\lambda_i^* \in [\lambda_i^{min}, \lambda_i^{max}]$ is a randomly generated step size and $\lambda_i^{min}, \lambda_i^{max}$ are two extremes of step size.

The following equation gives pattern search direction:

$$S_{ij} = P_j - P_{j-N} \quad (j = 1, 2, \dots, N_g) \quad (3.18)$$

and unidirectional search is performed.

3.4 SOLUTION METHODOLOGY

The solution approach to solve the multiobjective load dispatch problem, involves three main parts (i) The chaotic evolutionary programming pattern search procedure to evolve potential solutions (ii) Generation of Pareto-front (iii) Interaction with the surrogate worth trade-off decision maker to decide compromise solution. The stepwise flowchart of solution approach is shown in Figure 3.3 and is explained in the following sections:

3.4.1 Solution Representation and Initialization

The population, individuals represent the power of a generator and has dimensions as that of number of generators N_g . The initial population is randomly generated within respective limits of power generators. Therefore, each individual position in the population presents a solution in the feasible space of MOLD problem. An i^{th} solution at t^{th} iteration is denoted by $[P_{i1}^t, P_{i2}^t, \dots, P_{iN_g}^t]$. The associated strategy parameter is represented by $[\eta_{i1}^t, \eta_{i2}^t, \dots, \eta_{iN_g}^t]$. Mathematically, the initialization process for individual's is illustrated

as follows:

$$P_{ij} = P_j^{min} + \chi(P_j^{max} - P_j^{min}) \quad (j = 1, 2, \dots, N_g, i = 1, 2, \dots, N_P) \quad (3.19)$$

where $\chi \in [0, 1]$ is a chaotic sequence generated number and the j^{th} generator's operational limits is represented as $[P_j^{min}, P_j^{max}]$.

3.4.2 Non-inferior Solution Set Reformation

The Pareto front obtained from feasible solutions is modified by incorporating non-dominated solutions generated at each iteration. Considering the imprecise nature of DM's judgment, the objective functions are transformed to fuzzy representation by considering extremum of objective function as follows:

$$\mu(f_k(P)) = \begin{cases} 1 & ; f_k(P) \leq f_k^{min} \\ \frac{f_k^{max} - f_k(P)}{f_k^{max} - f_k^{min}} & ; f_k^{min} < f_k(P) < f_k^{max} \\ 0 & ; f_k(P) \geq f_k^{max} \end{cases} \quad (k = 1, 2, \dots, M) \quad (3.20)$$

where $\mu(f_k(P))$ is the membership function of k^{th} objective function. The objective function is accepted if its membership is 1 and rejected if it is 0. f_k^{min} and f_k^{max} are the extremum of k^{th} objective function, M is number of objective functions. The behaviour of membership function $\mu(f_k(P))$ with respect to the objective function $f_k(P)$ is presented in Fig. 3.1.

A feasible solution P_1 will replace a point P_2 in the Pareto-front, iff

$$\mu(f_k(P_1)) > \mu(f_k(P_2)) \quad (k = 1, 2, \dots, M) \quad (3.21)$$

3.4.3 Surrogate Worth Trade-off Decision Maker

The *Decision maker* (DM) makes trade-off within the frontier, instead of considering a complete range of available solutions. The SWT function is a quality inference tool that acts as an interface between the Pareto-front and the decision maker. The *Surrogate worth*

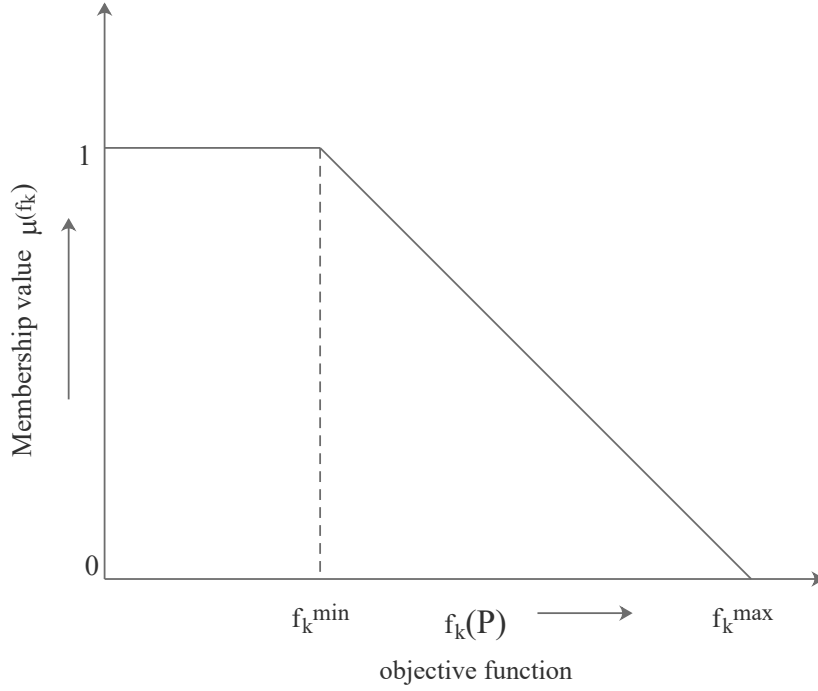


Figure 3.1: Membership function behaviour

trade-off (SWT) function takes into account the membership representation of objective functions, as an alternative to objective function value. It assigns trade-off values to the available non-inferior solutions at each iteration. Let μ_A be the aggregating function defined as follows:

$$\mu_A = \frac{\sum_{x=1}^M \sum_{\substack{y=1 \\ y \neq x}}^M \frac{\mu(f_x(P))}{\mu(f_y(P))}}{M(M-1)} \quad (3.22)$$

Using, the aggregated information available from Eq. (3.22) the SWT function is mathematically defined as follows:

$$\mu_D = \begin{cases} \mu_A & ; \mu_A \leq 1 \\ \mu_A^{-1} & ; \mu_A > 1 \end{cases} \quad (3.23)$$

Here μ_D indicates the degree of satisfaction of objectives and as it approaches unity, the problem's objectives are better satisfied. Figure 3.2 graphically illustrates the relation of μ_D and $\mu(f_k)$.

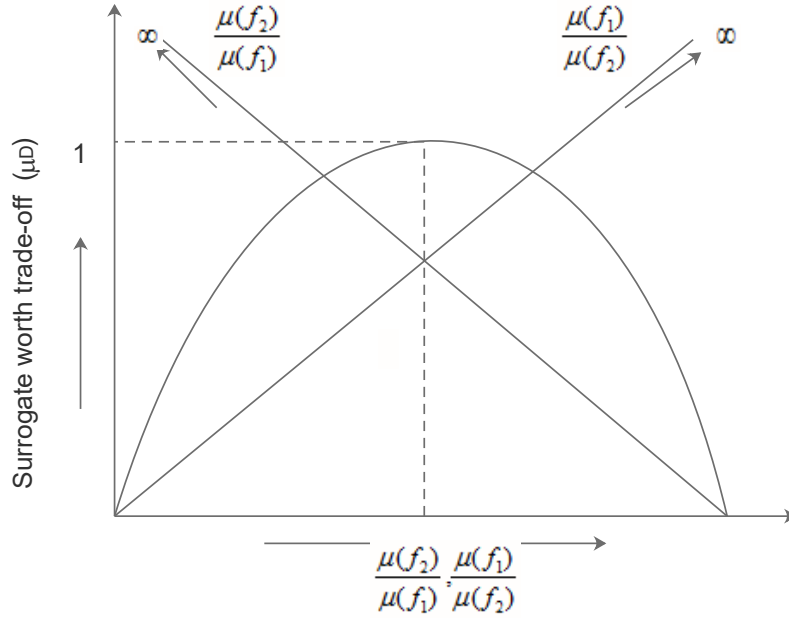


Figure 3.2: Graphical representation of surrogate worth trade-off (μ_D) behaviour

3.4.4 Population Individuals Update

The iterative process updates individual's population that represents the power generation of thermal generators. The chaotic mutation operator Eq. (3.10) and (3.11) is responsible for generation of offspring. The tournament selection operator (Section 3.3.1) selects better performing individuals for $k + 1^{th}$ generation out of existing parents and their offspring at k^{th} generation, on the basis of the number of assigned wins. The Eq. (3.15) used to assign wins is rewritten considering Eq. (3.23) as follows:

$$w_n = \begin{cases} 1 & ; \chi_1 < \frac{\mu_{Dm}}{\mu_{D_i} + \mu_{D_t}} \\ 0 & ; otherwise \end{cases} \quad (3.24)$$

The quality of population individual's is improved by Powell's pattern search as mentioned in section 3.3.2 as an interleaved process. The iterative process continues until a termination criterion is satisfied. The termination criteria used in this paper is the number of swarm moves (IT^{max}). Figure 3.3 depicts a stepwise solution procedure.

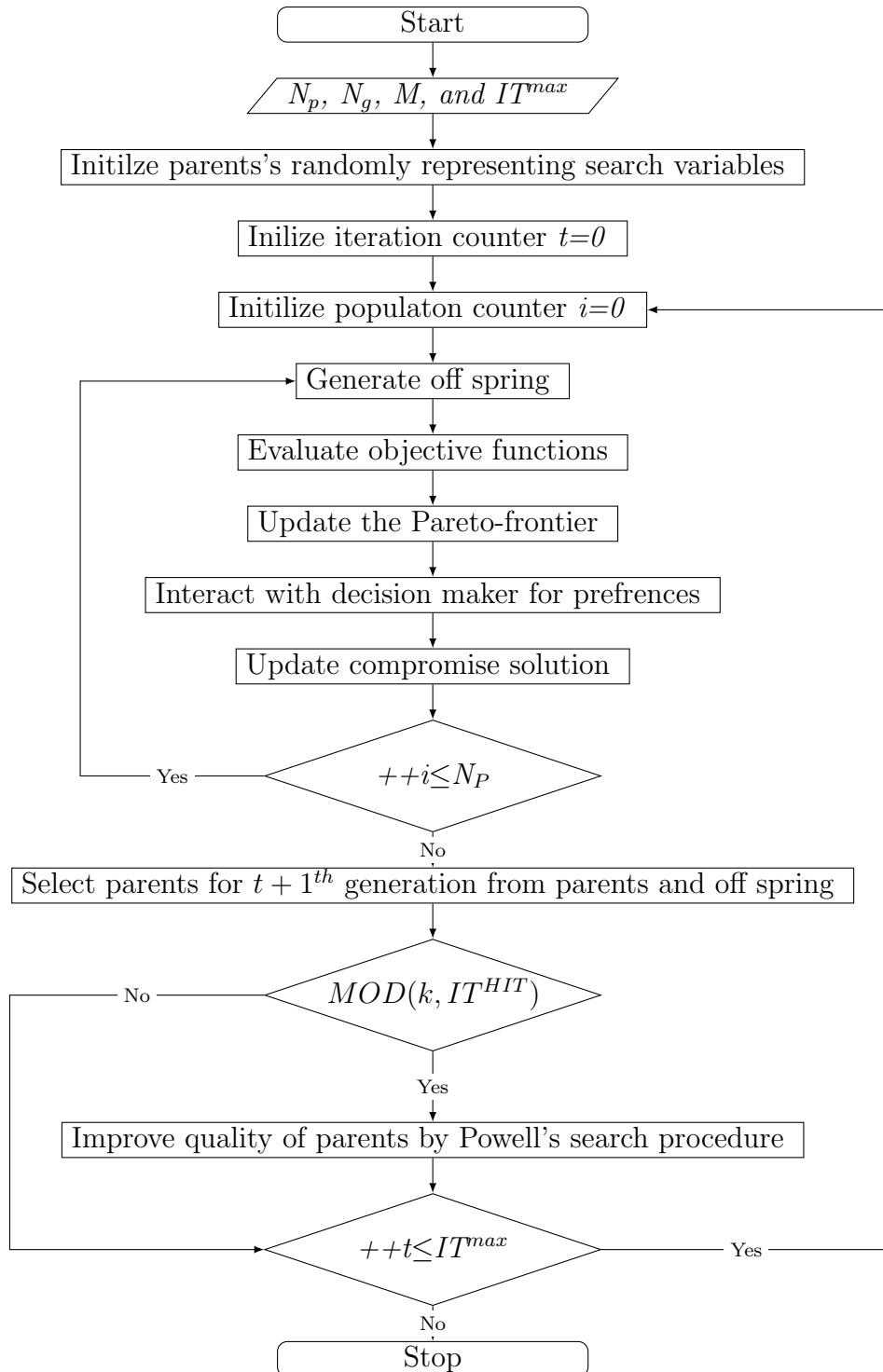


Figure 3.3: Flow chart of CEPPS based solution approach

3.5 SIMULATION CASES AND RESULTS

To prove the capability of the CEPPS solution approach, the generalized benchmark test functions as well as standard real world problems related to power system operation are undertaken. The Gauss and Tent map sequences are applied to investigate their behavior while implementing CEP and CEPPS. On using Gauss map and Tent sequence the CEP method is termed as CEP-1 and CEP-2. A similar notation is used for CEPPS procedures.

3.5.1 Test Systems

In order to illustrate the effectiveness of the proposed methodology, the generalized benchmark problems specified in Table 2.1 are simulated. The considered generalized benchmark problems that are uni-modal, multi-modal, separable, non-separable and discontinuous in nature. The details of these functions has been presented in Appendix C. The performance investigation, aims to compare the convergence behavior of CEP and CEPPS. The small, medium and large capacity power dispatch problems shown in Table 2.2 are undertaken to verify the practicability of SWT method whereby search is performed by CEPPS algorithm to generate the non-dominated solutions. The power generator characteristics are referred from Appendix D. The achieved results from proposed CEPPS variants are compared with the results obtained from CEP while implementing on generalized benchmark problem. The better one is further implemented to solve electric power system under different situation. The important observations of the simulation results are discussed in the following sections.

3.5.2 Generalized Test Functions

In order to compare the performance of CEPPS variants with CEP, seven benchmark test functions listed in Table 2.1 are solved using CEPPS and CEP variants. The convergence behavior of considered algorithms for test functions is plotted in Figure 3.4 for comparison. The best, worst and mean experimental results after 1000 iterations for each test function performing 30 trial runs are also listed in Table 3.1.

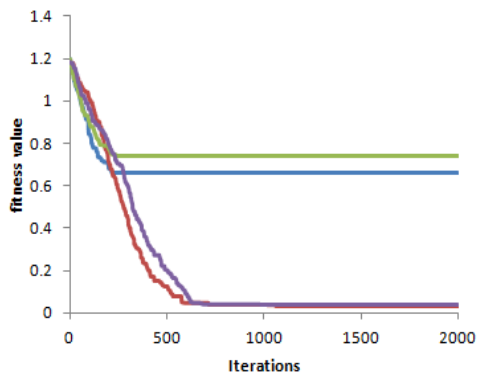
The convergence comparison plots for considered benchmark test functions are shown

Table 3.1: Performance analysis of generalized benchmark test functions

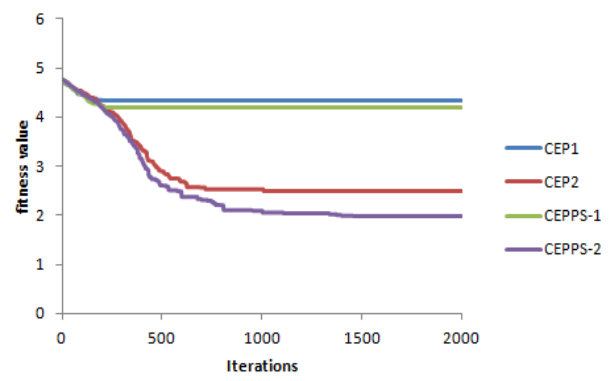
Test function	Fitness	CEP-1	CEP-2	CEPPS-1	CEPPS-2
Griewank function [380]	Worst	4.600014	1.083315	11.05058	9.29×10^{-01}
	Average	4.600014	1.083315	11.05058	9.29×10^{-01}
	Best	4.600014	1.083315	5.48433	0.01×10^{-01}
Rastrigin function [380]	Worst	21893.57	305.4496	40041.02	97.11104
	Average	21893.57	305.4496	40041.02	61.70118
	Best	21893.57	305.4496	15691.13	61.70180
Rosenbrock function [380]	Worst	$7.54 \times 10^{+08}$	222.4695	$1.00 \times 10^{+10}$	43304.03
	Average	$7.54 \times 10^{+08}$	22.36219	$1.00 \times 10^{+10}$	7300.95
	Best	$7.54 \times 10^{+08}$	22.36219	$2.91 \times 10^{+08}$	7300.95
Schwefel's 2.22 function[380]	Worst	48.16846	68.45393	7.574684	27.71013
	Average	48.16846	68.45393	7.574684	27.71013
	Best	48.16846	68.45393	7.574684	27.71013
Sphere function [380]	Worst	13094.38	8.236108	50471.00	4.79×10^{-19}
	Average	13094.38	8.236108	50471.00	9.40×10^{-20}
	Best	13094.38	8.236108	18642.14	9.40×10^{-20}
Step function [380]	Worst	670	37	969	28
	Average	670	37	969	28
	Best	670	15	532	15
Step-2 function [380]	Worst	15349	6	39277	8
	Average	15349	6	39277	5
	Best	15349	6	17381	5

in Figure 3.4 on a logarithmic scale. Figure 3.4a presents the convergence behavior of the Griewank's function which shows that the CEP-1 and CEPPS-1 are fast to converge in initial phase but have resulted in premature convergence. However, the CEP-2 and CEPPS-2 which were slow to converge initially, have resulted in much better solution. Finally, the best solution is obtained by CEPPS-2, as compared to others. The Figure 3.4b, depicts the behavior of Rastrigin's function, where same trend as in the case of the Griewank's function shown by various contenders is observed. However, for this function CEPPS-2 has performed significantly better than CEP-2 and achieved better solution.

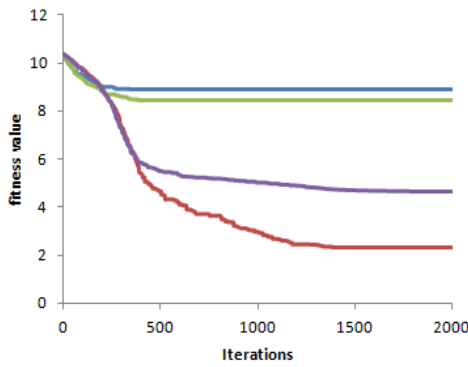
The appraisal of Figure 3.4c shows that in case of Rosenbrock's function the performance of CEP-2 is better than CEPPS-2, where as both CEP-1 and CEPPS-1 resulted in premature convergence. Further, the experimental result of Sphere function in Figure 3.4e, shows that the CEPPS-2 algorithm has performed best in terms of convergence rate. In case of Schwefel's function, both CEP-1 and CEP-2 are worst performers, and CEPPS-1 resulted in best solution of the problem. The convergence plot in Figure 3.4f and Figure 3.4g for step and step-2 function, presents CEP-2 and CEPPS-2 as a better performing



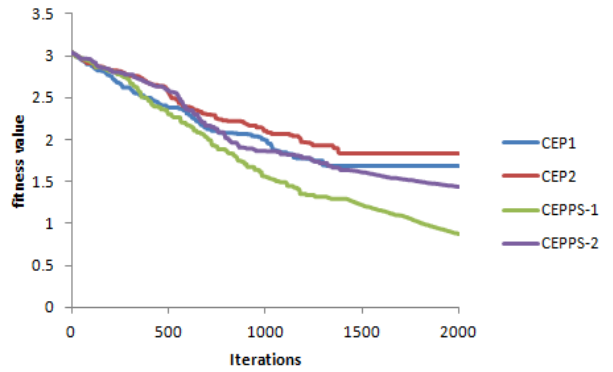
(a) Griewank function



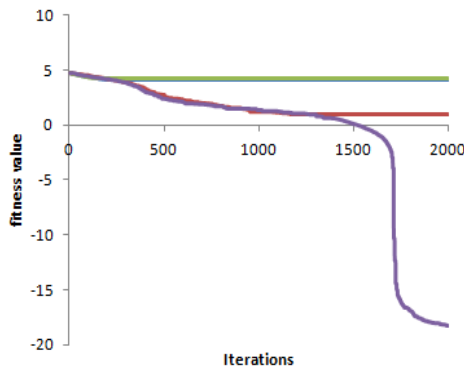
(b) Rastrigin function



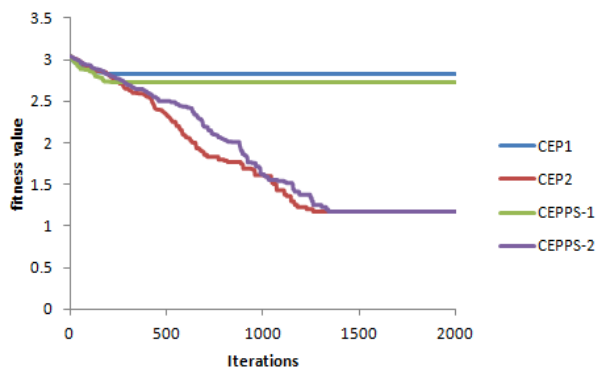
(c) Rosenbrock function



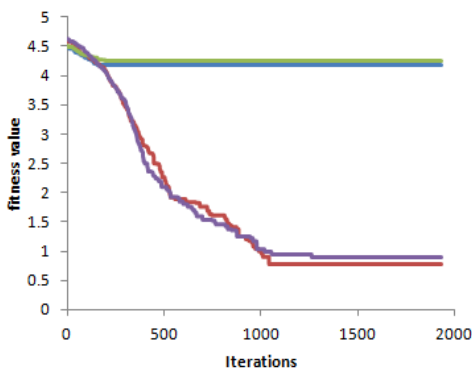
(d) Schwefel's 2.22 function



(e) Sphere function



(f) Step function



(g) Step-2 function

Figure 3.4: Generalized benchmark test functions convergence comparison

variant than others. Further, CEP-2 has performed even better than CEPPS-2 in case of Step-2 function.

Hence, the convergence plots clearly show that in comparison to CEP-1 and CEPPS-1, the CEP-2 and CEPPS-2 variants perform better. The better performance is either in terms of solution quality or/ convergence rate. Moreover, it is observed that CEPPS-1 performs better on some of the benchmark, whereas CEP-2 and CEPPS-2 shows superiority for other considered benchmark functions.

Further, in comparison to the synergic predator prey optimization algorithm (SPPO) results given in Table 2.4, the CEPPS has performed poorly for all the test functions.

Table 3.2: Test power system-1, Comparison of ELD ($P_D = 2700MW$)

Algorithm	cost(\$/h)					
	Case-1	Case-2	Case-3	Case-4	Case-5	Case-6
BBO [306]	624.51	–	–			
CCPSO [321]	–	623.82	–	–	–	–
CGA-MU [373]	623.80	624.71	–	–	–	–
DE [373]	623.80	624.46	–	–	–	–
DEBBO [305]	624.51	–	–			
ELHN [369]	624.51	–	–			
IGA [163]	624.51	–	–			
IGA-MU [163]	623.80	624.51	–			
KHA [397]	624.51	–	–			
PSO[373]	623.80	624.24	–			
QP-ALHN [376]	623.80	–	624.32			
SPPO	623.80	623.82	624.32	700.29	700.77	700.48
CEPPS-1	623.75	623.87	623.76	699.70	699.54	704.94
CEPPS-2	623.75	623.88	623.77	699.77	699.73	700.60

3.5.3 Test Power System-1

Firstly, this system depicted in Table 2.2, has been solved for ELD by applying CEPPS-2. In all the case population size N_P is fixed at 50 and the number of iterations IT^{max} is fixed at 2000. The cost comparison is performed with other published work and is presented in Table 3.2. The results of *Biogeography based optimization* (BBO) [306], *Composite Particle Swarm Optimization* (CCPSO) [386], *Conventional Genetic Algorithm with Mutation Update* (CGA-MU) [163], *Differential Evolution* (DE) [373], *Improved Genetic Algorithm* (IGA) [163], *Improved Genetic Algorithm with multiplier update* (IGA-MU)

[163], *Enhanced Augmented Hopfield Neural Network* (ELHN) [373], *Hybrid of DE and BBO* (DEBBO) [386], *Krill herd optimization* (KHA) [397], *Quadratic Programming Augmented Hopfield Neural Network* (QP-ALHN) [376], *Particle Swarm Optimization* (PSO) [373] and *Synergic Predator Prey Optimization* (SPPO). The result comparison shows

Table 3.3: Test power system-1, Cost and emission for ELD and MED respectively

Case	ELD		MED	
	Min. Cost(\$/h)	Max. Emission(lb/h)	Min. Emission(lb/h)	Max. Cost (\$/h)
1	623.75	11929.08	103.84	697.84
2	623.87	10798.01	103.84	703.26
3	623.76	12173.60	100.64	699.47
4	699.70	77292.20	126.34	791.57
5	699.54	39404.71	149.31	796.45
6	700.60	21074.90	163.47	776.40

that CEPPS-2 has resulted in a solution with lower power generation cost in comparison to other results available in literature in all the cases except for Case-2 and Case-6. In Case-2, the results are comparable with CCPSO and SPPO algorithm. The proposed method gives 0.05 and 0.06 \$/h, higher cost than SPPO and CCPSO respectively that is very small difference. In Case-6, also the results of CEPPS-2 are competitive with SPPO, having only a difference of 0.12\$/h. However, among CEPPS-1 and CEPPS-2, the CEPPS-1 has obtained better solution in the majority of cases. The minimum value of pollutant emission while performing MED is shown in Table 3.2 along with generation cost which is contrary in nature. Further, the Test system-1 is formulated in the multi-objective framework. The quality of compromise solution is indicated by μ_D (Eq. 3.23). The comparison of MOLD solution obtained by CEPPS-1 and CEPPS-2 is presented in Table 3.4 and 3.5. The comparison of MOLD results on basis of μ_D shows that CEPPS-1 has obtained a better quality compromised solution in the majority of the cases in comparison to CEPSS-2. The corresponding MTPLD power distribution plan is tabulated in Table 3.6 and 3.7 respectively.

3.5.4 Test Power System-2

This test system has 40 generators and to obtain the solution N_P and IT^{max} are considered as 100 and 3000 respectively. The optimal solution comparison for ELD, MED and

Table 3.4: Test power system-1, MOLD solution comparison ($P_D = 2700MW$)

Algorithm	Case-1			Case-2			Case-3		
	Cost(\$/h)	Emission(lb/h)	μ_D	Cost(\$/h)	Emission(lb/h)	μ_D	Cost(\$/h)	Emission(lb/h)	μ_D
CEPPS-1	629.49	1185.10	0.922	629.97	1263.65	0.922	629.62	1208.05	0.922
CEPPS-2	629.52	1192.03	0.922	630.15	1277.18	0.920	628.07	663.66	0.947

Table 3.5: Test power system-1, MOLD solution comparison ($P_D = 2700MW$)

Algorithm	Case-4			Case-5			Case-6		
	Cost(\$/h)	Emission(lb/h)	μ_D	Cost(\$/h)	Emission(lb/h)	μ_D	Cost(\$/h)	Emission(lb/h)	μ_D
CEPPS-1	701.31	1049.95	0.999	701.33	981.0	0.999	705.11	879.83	0.999
CEPPS-2	705.10	1129.55	0.998	704.97	972.46	0.999	704.97	3086.99	0.995

Table 3.6: Test power system 1, MOLD Power distribution ($P_D = 2700MW$)

Generator	Case-1		Case-2		Case-3	
	Fuel type	Power(MW)	Fuel type	Power(MW)	Fuel type	Power(MW)
P ₁	2	209.20	2	210.93	2	208.19
P ₂	1	209.30	1	207.41	1	207.26
P ₃	1	228.65	1	231.42	1	230.15
P ₄	3	247.41	3	241.68	3	241.41
P ₅	1	294.35	1	299.23	1	291.67
P ₆	3	245.72	3	242.48	3	245.47
P ₇	1	311.21	1	319.04	1	321.27
P ₈	3	244.56	3	247.48	3	245.11
P ₉	3	438.64	3	439.15	3	439.66
P ₁₀	1	270.80	1	261.03	1	269.66

Table 3.7: Test power system-1, MOLD Power distribution ($P_D=2700$)

Generator	Case-4		Case-5		Case-6	
	Fuel type	Power(MW)	Fuel type	Power (MW)	Fuel type	Power(MW)
P ₁	2	223.88	2	219.30	2	220.17
P ₂	1	213.60	1	219.33	1	216.78
P ₃	3	334.84	3	334.75	3	250.99
P ₄	3	240.45	3	241.04	3	245.43
P ₅	1	305.03	1	302.75	1	345.44
P ₆	3	243.10	3	244.49	3	240.19
P ₇	1	306.92	1	312.07	1	347.66
P ₈	3	242.22	3	242.50	3	240.33
P ₉	3	435.86	3	439.12	3	439.89
P ₁₀	1	295.56	1	285.92	1	294.78
P _L (MW)		141.62		141.41		141.80

Table 3.8: Test power system-2, Optimal solution (cost, emission) comparison ($P_D = 10500MW$)

Algorithm	ELD		MED		MOLD		μ_D
	Cost $\times 10^5 \$/h$	Emission $\times 10^5 lb/h$	Cost $\times 10^5 \$/h$	Emission $\times 10^5 lb/h$	Cost $\times 10^5 \$/h$	Emission $\times 10^5 lb/h$	
GSA [358]	–	–	–	–	1.25782	2.10932	0.988
LMPSO [417]	–	–	–	–	1.24668	2.19048	0.947
MABS I [424]	1.21412	3.59901	1.29995	1.766820	1.24490	2.5656	0.872
MABS II [424]	1.21412	3.59901	1.29996	1.766820	1.24491	2.56551	0.872
MODE [330]	1.2184	3.7479	1.2996	1.7668	1.25792	2.1119	0.989
NSGA II [330]	–	–	–	–	1.25825	2.10949	0.989
PDE [330]	–	–	–	–	1.25730	2.1177	0.987
SEPA II [330]	–	–	–	–	1.25807	2.1109	0.989
SMPSO [417]	–	–	–	–	1.24660	2.17256	0.949
TLBO [384]	1.21685	3.64593	1.29953	1.766835	1.26020	2.06648	0.995
PPO	1.21413	3.60420	1.29979	1.766843	1.25750	1.99125	0.994
SPPO	1.21402	3.56724	–	–	–	–	–
CEPPS-1	1.27180	3.74117	1.34516	2.46508	1.28124	3.14471	0.944
CEPPS-2	1.21487	3.63871	1.32889	2.25013	1.28515	3.27227	0.938

MOLD are presented in Table 3.8. The results are compared with the results of *Gravitational Search Algorithm* (GSA) [358], *Non-dominated sorting Genetic Algorithm* (NSGA) [330], *Modulated Artificial Bee Colony* (MABS) [424], *Multiobjective Differential Evolution* (MODE) [330], *Modulated Particle Swarm Optimization* (LMPSO) [417], *Pareto Differential Evolution* (PDE) [330], *Quasi Teacher Learner Based Optimization* (QTLBO) [384], *Real Coded Chemical Reaction Optimization* (RCCRO), *Sinusoidal Modulated Particle Swarm Optimization* (SMPSO) [417], *Strength Pareto Evolutionary algorithm* (SPEA) [330], *Teacher Learner Based Optimization* (TLBO) [384], *Predator Prey Optimization* (PPO) and *Synergic Predator Prey Optimization* (SPPO). The ELD solution comparison with other algorithm's results from the literature shows that CEPPS-2 has resulted in lesser cost as compared to MODE and TLBO. However, it has resulted in slightly higher cost in comparison to others. While considering the MOLD, the comparison of compromise solution with results available in literature is performed using index μ_D . The comparison of the results shows that the CEPPS results are better than MABS-1 and MABS-2 and RCCRO, whereas the quality of compromise solution is poorer than other algorithms considered for comparison.

Table 3.9: Test power system-2, Power generation (in MW) schedule by CEPPS
($P_D = 10500MW$)

Unit	ELD	MED	MOLD	Unit	ELD	MED	MOLD
P_1	76.56	41.68	114.0	P_{21}	537.22	483.03	488.34
P_2	41.11	41.56	40.90	P_{22}	522.15	490.19	519.37
P_3	69.06	94.99	93.53	P_{23}	521.68	458.42	512.01
P_4	128.93	92.90	169.44	P_{24}	530.18	446.54	525.58
P_5	54.69	55.74	97.0	P_{25}	522.90	434.08	520.71
P_6	107.07	77.87	104.19	P_{26}	526.14	483.41	519.46
P_7	192.53	294.42	243.85	P_{27}	22.46	32.23	16.86
P_8	300.0	296.31	300	P_{28}	17.59	59.97	17.38
P_9	300.0	300.0	240.17	P_{29}	16.32	53.38	27.79
P_{10}	137.40	167.17	200.48	P_{30}	62.67	56.73	88.08
P_{11}	306.15	306.83	337.45	P_{31}	166.27	189.89	190.0
P_{12}	237.89	320.66	248.28	P_{32}	160.88	164.31	190.0
P_{13}	446.40	442.46	477.94	P_{33}	142.32	185.87	124.33
P_{14}	301.61	474.39	336.43	P_{34}	163.10	200.0	164.43
P_{15}	384.94	458.03	442.79	P_{35}	176.74	191.14	177.55
P_{16}	493.50	457.63	298.09	P_{36}	158.08	174.85	167.33
P_{17}	490.38	441.23	477.61	P_{37}	72.10	78.70	46.29
P_{18}	500.0	462.37	413.35	P_{38}	56.46	33.78	88.70
P_{19}	512.42	452.31	432.98	P_{39}	27.13	46.39	43.10
P_{20}	511.44	472.01	491.75	P_{40}	505.33	486.33	512.30

Table 3.10: Test power system 3, ELD solution comparison ($P_D = 49342MW$)

Algorithm	cost(\$/h)	
	Case-1	Case-2
CCPSO [321]	1655685.0	1657962.75
CQGSO [360]	1655679.42	1657962.74
DHS [388]	–	1657944.86
GSO	1734405.7	–
PPO	1655684.0	1657963.5
APSO	1655684.5	1657963.0
SPPO	1655680.0	1657962.0
CEPPS-1	1778880.0	1795760.0
CEPPS-2	1778471.0	1784659.0

Table 3.11: Test power system 3, Solution comparison ($P_D = 49342MW$)

Algorithm	Case-1				Case-2			
	ELD		MED		ELD		MED	
	Cost \$/h	Emission lb/h	Cost \$/h	Emission lb/h	Cost \$/h	Emission lb/h	Cost \$/h	Emission lb/h
CEPPS-1	1778880	6613403	1947279	5112745	1795760	6562387	1953030	5112745
CEPPS-2	1778471	6538250	1970072	5051650	1784659	6550347	1975453	5051650

Table 3.12: Test power system 3, MOLD solution comparison ($P_D = 49342MW$)

Algorithm	Case-1			Case-2		
	Cost(\$/h)	Emission(lb/h)	μ_D	Cost(\$/h)	Emission(lb/h)	μ_D
CEPPS-1	1831802	5902779	0.876	1832679	5932484	0.883
CEPPS-2	1815388	5695509	0.940	1821931	5790238	0.929

3.5.5 Test Power System-3

This is a Korean power system having 140 power generators. In order to, obtain the solution for all cases the N_P and IT^{max} is fixed at 200 and 5000 respectively. Firstly, the optimal ELD solution of this problem has been obtained using CEPPS and is compared with other available solutions from the literature, in Table 3.10. The results of CEPPS-1 and CEPPS-2 are compared with *Composite Particle Swarm Optimization* (CCPSO) [321], *Continuous Quick Group Search Optimizer* (CQGSO) [360], *Differential Harmony Search* (DHS) [388], *Anti Predator Swarm Optimization* (APSO), *Predator Prey Optimization* (PPO). The cost comparison, shows that CEPPS variants have resulted in premature convergence for both cases-1 and 2. However, among CEPPS-1 and CEPPS-2, the latter one has obtained a solution with lesser power generation cost. Similarly, while performing the minimum emission dispatch, the outcomes of two CEPPS variants are presented in Table 3.11, which shows that CEPPS-2 has ability to obtain the better solution. Further, while performing multi-objective load dispatch the compromise solution obtained by CEPPS-1 and CEPPS-2 has been compared in Table 3.12. From the comparison on basis of μ_D , it is concluded that CEPPS-2 has obtained a better compromise solution. The optimal power distribution plan for MOLD by CEPPS-2 has been tabulated in Table 3.13.

Table 3.13: Test power system-3, Optimal power allocation using CEPPS-2 (in MW)

Unit	ELD	MED	MOLD	Unit	ELD	MED	MOLD	Unit	ELD	MED	MOLD	Unit	ELD	MED	MOLD
1	81.79	80.42	91.50	36	500.00	408.81	407.12	71	311.11	236.70	312.52	106	838.85	836.09	845.68
2	124.54	142.58	150.77	37	210.89	180.35	187.33	72	396.29	443.19	455.00	107	859.75	820.68	827.12
3	153.86	145.06	168.19	38	210.14	206.70	225.69	73	541.00	463.65	527.01	108	821.38	806.10	832.95
4	190.00	128.20	183.70	39	774.00	511.13	725.92	74	351.42	393.93	477.72	109	848.20	806.66	810.12
5	183.84	105.86	113.68	40	769.00	501.04	769.00	75	449.12	333.77	458.42	110	864.80	833.14	845.70
6	149.29	133.52	95.49	41	6.26	5.83	11.06	76	366.21	449.18	394.31	111	852.80	842.12	827.11
7	490.00	405.15	490.00	42	10.31	28.00	5.96	77	239.63	447.10	289.23	112	120.65	149.12	126.00
8	470.58	417.01	405.72	43	216.80	164.13	192.26	78	378.87	488.69	441.42	113	100.75	182.52	147.75
9	494.53	449.81	496.00	44	173.13	177.09	184.36	79	531.00	309.15	438.33	114	125.30	149.51	122.34
10	496.00	377.50	496.00	45	185.53	206.03	199.08	80	467.42	455.69	512.98	115	298.37	340.08	313.36
11	494.38	314.20	481.99	46	174.30	176.23	176.94	81	442.30	441.76	516.68	116	303.56	372.85	305.40
12	496.00	298.93	496.00	47	224.58	178.91	175.61	82	79.54	60.81	72.35	117	275.89	288.03	282.76
13	505.46	386.35	481.50	48	201.86	234.61	180.79	83	177.53	245.00	147.30	118	110.04	124.31	98.43
14	509.00	367.02	506.71	49	198.88	168.88	197.76	84	148.74	191.48	148.68	119	110.64	151.62	116.88
15	506.00	506.00	430.76	50	218.93	244.34	216.07	85	134.06	141.94	150.97	120	170.47	142.34	144.39
16	460.39	364.46	488.25	51	199.63	500.83	216.87	86	209.39	258.02	213.90	121	194.51	246.68	215.21
17	506.00	503.07	422.48	52	192.28	504.00	264.85	87	258.43	214.07	219.49	122	6.26	6.30	3.60
18	506.00	393.56	506.00	53	189.75	495.30	344.10	88	199.48	301.37	281.46	123	15.91	26.67	23.54
19	505.00	315.00	498.29	54	207.98	504.00	320.35	89	230.98	335.60	280.06	124	66.74	32.51	37.06
20	505.00	393.80	505.00	55	253.72	471.00	237.43	90	219.16	231.48	201.30	125	22.18	25.28	14.55
21	407.28	505.00	365.67	56	529.01	514.79	500.48	91	188.72	336.21	185.39	126	17.07	28.23	14.56
22	421.09	475.24	438.14	57	248.47	278.25	341.00	92	549.15	552.50	575.40	127	14.60	22.55	13.95
23	466.82	505.00	495.76	58	379.10	566.60	379.24	93	547.50	519.27	547.50	128	268.17	334.00	205.69
24	486.78	354.65	505.00	59	153.05	190.76	194.27	94	803.24	820.40	820.58	129	7.35	8.03	4.55
25	537.00	515.00	537.00	60	216.15	263.31	226.14	95	831.14	822.52	813.95	130	19.10	14.35	17.13
26	432.83	449.13	428.29	61	320.09	464.24	398.08	96	638.49	680.79	651.09	131	5.53	16.18	11.57
27	445.39	288.66	382.87	62	168.23	302.00	131.30	97	663.23	711.92	656.49	132	83.33	55.57	63.42
28	452.40	346.54	444.97	63	329.93	511.00	472.44	98	718.00	661.85	718.00	133	10.00	6.19	5.91
29	501.00	322.41	351.97	64	249.00	380.40	397.41	99	720.00	682.96	720.00	134	54.21	56.51	50.99
30	417.09	445.15	458.29	65	327.49	389.66	269.54	100	948.43	917.37	956.53	135	53.94	48.10	55.88
31	506.00	506.00	488.05	66	289.10	452.53	261.46	101	928.88	945.12	953.14	136	53.56	94.81	49.73
32	492.20	431.42	451.56	67	435.30	213.40	271.36	102	912.91	895.75	939.74	137	20.85	21.65	26.04
33	506.00	411.95	496.48	68	408.58	200.53	307.17	103	934.00	864.53	871.95	138	18.51	8.22	11.75
34	342.66	454.56	414.50	69	147.85	140.04	207.53	104	935.00	888.83	892.83	139	10.13	7.98	17.92
35	484.32	500.00	476.74	70	344.12	356.00	285.94	105	829.84	837.44	854.30	140	28.47	28.11	31.79

3.5.6 Robustness Analysis

A statistical analysis is performed with 30 independent trials to compare the performances of CEPPS-1 and CEPPS-2. The Test power system-1 has been considered for the robustness analysis. Thirty independent trials are performed in both cases 1-3, for ELD of the system using CEPPS-1 and CEPPS-2. The corresponding data statistics are tabulated in Table 3.14, from where it is evident that CEPPS-1 has provided better solution quality, whereas CEPPS-2 has provided problem's solution with lesser standard deviation.

Table 3.14: Test power system-1, Descriptive statistics

Attributes	CEPPS-1			CEPPS-2		
	Case-1	Case-2	Case-3	Case-1	Case-2	Case-3
Maximum	623.78	624.23	623.8	623.76	626.45	623.76
Minimum	623.75	623.87	623.76	623.75	623.88	623.77
Mean	623.76	623.96	623.76	623.76	624.11	623.76
S.D.	0.0047	0.061	0.009	0.0031	0.051	0.002

3.6 CONCLUSION

In this chapter, a hybrid called chaotic evolutionary programming and pattern search (CEPPS) has been presented. The CEPPS follows a chaotic map sequence of random numbers. In order to enhance the exploration capability Powell's pattern has been introduced in the search process under stochastic control. The two chaos maps, namely Gauss map and Tent map are considered for performance analysis of CEPPS. The proposed algorithm is investigated as a solution procedure to solve MOLD. The conflicting objectives of MOLD have been taken care by surrogate worth trade-off approach. Computer simulations are performed on generalized test functions and MOLD problems to verify the effectiveness of CEPPS variants. The numerical results of generalized benchmark problems reveal that tent map based CEPPS has better search capability to find the optimal solution in the majority of test problems. The experimentation on Test power systems have shown that for a smaller power generation system, the CEPPS variants have obtained a better solution than the results available in the literature. In specific cases, the results are comparable with the synergic predator prey optimization and composite particle swarm optimization algorithm. However, while handling the ELD of higher power generation systems, the CEPPS variants have resulted in premature convergence. While, dealing with MOLD of smaller system, it is observed that the CEPPS has obtained the better solution than a few algorithms considered for comparison, whereas for other systems it has resulted in premature convergence. Further, it is observed in comparison to the Gauss map based CEPPS, tent map based CEPPS has better ability to search solution, while dealing with test power systems.

CHAPTER 4

MULTIOBJECTIVE LOAD DISPATCH USING ADAPTIVE PREDATOR PREY OPTIMIZATION

4.1 INTRODUCTION

Real-world optimization problems, related to engineering field, are mostly multiobjective optimization problems with conflicting goals, which have a cumbersome solution procedure. *Multiobjective load dispatch* (MOLD) is unitary of the main problems of power system operation. The objective of MOLD problem is to operate the power generators in such a way that it generates required load demand and minimizes the operating costs and pollutants emission objectives simultaneously while satisfying physical and operational constraints. In later years, the MOLD has attained considerable attention, as it results in beneficial economics and pollutant emission.

On the basis of the process applied to tackle multiple objectives of a problem, the MOLD solution approaches are classified in three categories [268]. The first approach treats pollutant emission as a constraint with permissible limit, but it is silent regarding trade-off front information [31], time consuming, results in weak non-dominated solutions [122] and requires a set of parameters to provide feasible solutions [147]. The price penalty method is an alternative approach, but it is unable to come up to the conflict of objectives. The second approach treats gaseous pollutant emission as an objective along with operating fuel cost function. The problem is undertaken as single objective, using the weighting-sum method [93, 277]. The weighting method belongs to priori articulation of preference category [147] and allows the user to specify preferences, which may be articulated in

terms of relative importance of different objectives. These parameters are continuously varied in an attempt to present the complete Pareto front. This method has advantages of least programming and computation complexity. The major drawbacks of weighting method are that the priori selection of weights does not assure acceptability of the final solution, inability to obtain points on non-convex portions of Pareto front, lesser preference articulation potential, require multiple trails of algorithm with user specified preferences and uneven spread of points on Pareto-curve [70]. The third approach considers both operating fuel cost and gaseous pollutant emission simultaneously as competing objectives [268]. Due to the development of *multiobjective evolutionary algorithm* (MOEA) this approach has received wide interests as a Pareto-optimal solution set can be obtained in a single run. Some of the MOEA techniques are summarized as a *Nitced Pareto Genetic Algorithm* (NPGA) [359], *Non-dominated sorting Genetic Algorithm* (NSGA) [286], MOEA [121], *Strength Pareto Evolutionary algorithm* (SPEA) [121], SPEA-II [330] and adaptive group search algorithm [438]. Still, these algorithms have some drawbacks such as NSGA is unable to maintain lateral diversity and uniformity of Pareto-front [341], *Multiobjective Differential Evolution* (MODE) ends up with premature convergence while handling multi-modal problems [345]. The MOEA approach, when used with a problem having large number of objectives, Pareto dominance sorting based approach becomes ineffective in finding out the quality solution [341]. The MOEA results in Pareto solution set and if these solutions are more uniformly distributed in Pareto front, the quality of the solution becomes better and one such quality measure is discussed in [129].

To get the quality global solution of the MOLD problem with better computational efficiency is still a potential research field. A view of conventional techniques applied to solve MOLD problem is summarized by Talaq and El-Hawary [51]. The traditional gradient based optimization techniques may not converge to feasible solution with non-differentiable and/or has a discontinuous objective function [364]. Abido [268] has applied Pareto based multiobjective procedures to handle conflicting targets of MOLD problem. The alternative techniques to solve MOLD are multiobjective chaotic ant swarm optimization algorithm [307], multiobjective harmony search algorithm [351], combination of differential evolution and biogeography based algorithm [331], chaotic artificial bee colony

[423]. The fuzzy theory based approaches, such as fuzzy logic controlled genetic algorithm [74], fuzzy based modified bacterial foraging activity [312] are also used for solving MOLD.

Evolutionary algorithms (EA) are stochastic optimization techniques that mimic the simplified model of the natural development process. The success of EA is due to their parallel processing mechanism, which results in an extended exploration of the search space. In the previous decade population based optimization techniques *Genetic Algorithm* (GA) [126], *Simulated Annealing* (SA) [133], *Evolutionary Programming* (EP) [91] and *Particle Swarm Optimization* (PSO) [54] are applied on a variety of applications. In comparison to other population based techniques, PSO has emerged as computationally better technique in terms of memory and speed [317]. Despite having the benefits of computational efficiency, the basic PSO suffers from problem of parameters tuning and uniqueness of solution [197]. To improve the global search capability many revised versions of PSO are available in the literature. These mainly focus on improvement in the evolution process, exploring best parameter combinations, reducing inertia weight, constriction factor [317], quantum behavior [317], chaotic concepts [245], neighborhood topologies [114] etc.

The introduction of predator-prey behavior was another attempt to ameliorate the performance of PSO resulting in *Predator Prey Optimization* (PPO) [197, 118]. In the PPO, a second population of particles, called predators balance exploration and exploitation. The predator has a different dynamic behavior of the prey particles. A best prey individual in the swarm enchain predator, while the prey particles are annoyed by the predator's presence (refer section A.2 appendix A). The intensity and frequency of the interactions between predators and prey sustain diversity in the swarm, especially when it is approaching convergence, thus cutting the risk of convergence to local sub-optima [197]. PPO is successfully applied to power system optimization problem [361]. At the same time, it is observed that the PPO's performance is also parameter dependent and the decision of prey to retreat from predator does not resemble a natural decision making procedure. So, there is still a scope to modify the model of PPO and recently, PPO hybrids [400] are suggested to solve multi-modal problems. Further, a PPO's redefined model based on prey's decision making is introduced, but this model too uses partial

aspect of the decision making.

The natural decision-making process is influenced by recognition of circumstances and one such paradigm is *Recognition Primed Model* (RPD) [396]. Reinforcement and inhabitation are two important aspects of decision-making. Thus, *Adaptive Predator Prey Optimization* (APPO) algorithm is suggested, where the fear factor depends on the cognitive and social behavior of preys. It ensures continuous mobility (magnitude and direction) of prey and hence resulting better diversification of solutions in the domain throughout the search process. In addition, in case of velocity limit violations, the velocity changes in discrete steps in view of predators position. It results in steady movement of particles and hence better exploitation of the search area.

The intent of the chapter is to solve MOLD problem using the modified weighting method. The modified weighting method proceeds as an interactive process instead of a priori approach of conventional weighting method, as it overcomes inability to generate an acceptable solution in a single trial of conventional weighting method. The proposed APPO approach is used to obtain the optimal solution of MOLD. In APPO, the state of mental ambiguity of prey is exploited to determine the effective direction of prey's movement. Preys follow kinematic rules to handle velocity limit violation and have a feature of adaptive velocity limits for preys. A fuzzy based decision-making approach updates the best-compromised solution. To corroborate the effectiveness of proposed solution approach standard electrical power system networks having small, medium and big power generation capacity is employed. At the same time, to give a naturalistic aspect of load dispatch problem, multiobjective case of multi-fuel system and 140 units Korean system are simulated. Multi-fuel system considers transmission losses along with nonlinearities of valve point loading, ramp rate limits and *Prohibited Operating Zone* (POZ). The results obtained by the proposed solution approach are compared with the results of available state-of-the-art algorithms. The proposed solution procedure has a lesser computation requirement and ensures acceptable solution in a single run.

This chapter is divided into seven sections. Section 4.2, presents conceptual details of APPO algorithm. Section 4.3 proposes weighting approach as an interactive approach. Section 4.4 presents decision making to find the best solution. The section 4.5 presents

the proposed solution approach to solve MOLD problem. Section 4.6, presents numerical results by the proposed method and compares with some of the most recently published work. Finally, Section 4.7 concludes the chapter.

4.2 ADAPTIVE PREDATOR PREY OPTIMIZATION

APPO is a refined swarm based search procedure, sharing mathematical foundations of the PPO concept (section A.2, Appendix A). The possible solutions of the problem are prey. Another principle component of APPO is a predator, whose intention is to chase the best position of prey particle. During this tracking, the prey particles near predator try to move away. In PPO, the preys decision for a safe position is based on fear, which is related to specific behavior of escape and avoidance (section A.2, Appendix A). Whereas in APPO, the fear factor is proposed in relation to the indecisive state of mind having previous best experiences (reinforcement) as well as revulsion for a predator (inhibition). Thus, to exemplify this behavior of prey, it becomes necessary to split fear factor into two roles *i.e.* cognitive fear and social fear. It is considered that fear of predator is remembered by the prey, considering its previous experience cognitively as well as socially. Therefore, in APPO previous individual and swarm experiences decide the diversity of the prey particles. At the same time, prey particles restrict their movement by adjusting the minimum and maximum velocities adaptively. Moreover, prey particle adjusts its velocity on the violation of velocity limits considering the predator's position. The ensuing sub-sections outline the detail of APPO. The fundamental of APPO approach is to roughly mimic the RPD model of naturalistic decision making [40]. To implement this model prey's fear is treated as *social fear* and *cognitive fear*. The new direction of prey's movement will depend upon fear factor, which is the ability of prey to recognize danger leading to an impulse to flee/ confront it. The relative proximity of individual's best position and global best position from the predator jointly influence fear factor. The ultimate purpose of the alteration is to maintain diversification among swarm positions, by applying varying impact of individual and social behavior according to fear component.

Thus, the velocity update rule becomes:

$$v_{ij}^{t+1} = wv_{ij}^t + c_1 \text{rand}() (x_{ij}^B - x_{ij}^t) + \text{rand}() a_1 \gamma e^{-b\beta_{i1}} + c_2 \text{rand}() (g_j^B - x_{ij}^t) + \text{rand}() a_2 \gamma e^{-b\beta_2} \quad (j = 1, 2, \dots, N_g; i = 1, 2, \dots, N_P) \quad (4.1)$$

and

$$\gamma = \begin{cases} 0 & ; P_f < F_f \\ 1 & ; P_f \geq F_f \end{cases}$$

where a_1 and a_2 represents the maximum amplitude of the predator's effect over prey due to cognitive and social behaviors, respectively. β_{i1} and β_2 are factors that depend on prey's experience and predator position given by expression:

$$\beta_{i1} = \sqrt{\sum_{j=1}^{N_g} (x_{ij}^B - x_{ij})^2} \text{ and } \beta_2 = \sqrt{\sum_{j=1}^{N_g} (g_j^B - x_{pj})^2}$$

4.2.1 Discrete Velocity Variation

In addition, the prey's velocity should not violate the extreme limits, so that, the movement of a particle remains within the search domain. In case of velocity limit violation, the prey cannot change velocity abruptly due to the presence of a predator. On the other hand, the prey may violate limits to have a narrow escape from a predator. So, the change in velocity change is based on predator's position resulting in fear to prey. Thus, in this paper an iterative procedure is taken over to adjust the velocities, until the velocity limits are met. Mathematically, it is presented as:

$$v_{ij}^t = v_{ij}^t + \text{rand}() (v_{ij}^{\max} - v_{ij}^{\min}) + c_4 \text{rand}() \gamma (x_{pj}^t - x_{ij}^t) \quad (4.2)$$

$$(j = 1, 2, \dots, N_g; i = 1, 2, \dots, N_P)$$

The positions of prey particles are updated using following relation:

$$x_{ij}^{t+1} = x_{ij}^t + v_{ij}^{t+1} \quad (j = 1, 2, \dots, N_g; i = 1, 2, \dots, N_P) \quad (4.3)$$

4.2.2 Adaptive Velocity Limit

Further to explore the search area effectively, it becomes necessary that as the swarm moves ahead towards a solution, the velocity limits are adjusted in a certain fashion, so that the swarm particles changes their position in variable step size within velocity restrictions. Therefore, velocity range reduces depending on the moves of the swarm as described below:

$$v_j^{\min} = v_j^{\min} + \delta^t \quad (j = 1, 2, \dots, N_g) \quad (4.4)$$

$$v_j^{\max} = v_j^{\max} - \delta^t \quad (j = 1, 2, \dots, N_g) \quad (4.5)$$

where $\delta^t = \frac{\delta^t}{IT^{\max}} (m - MOD(t^{\max}, m))$ and IT^{\max} is the maximum number of swarm moves. m is the value to be fixed such that $m < IT^{\max}$.

4.3 WEIGHTING SUM METHOD

The conventional weighting sum method converts MOLD constrained optimization problem into a single objective constrained optimization problem. Weighting method gives non-inferior solution set when searched by varying assigned weights, w_k to the objectives and it requires enormous computation time to determine the complete Pareto front by varying the weights. To surmount these problems, the present work treats the preference weight vector as search variables and is updated as iterative process continues. Therefore, the assigned weights, w_k to objectives are searched along with decision vector P_j using APPO. Each movement of swarm provides a set of non-inferior solutions. The thermal power dispatch problem (Eq. (3.1)) is rewritten by assigning preference weights as below:

Minimize

$$F(x_j) = w_k f_k(x_j) \quad (k = 1, 2, \dots, M) \quad (4.6)$$

subject to

$$\sum_{k=1}^M w_k = 1, \quad w_k \geq 0$$

and constraints represented by Eqs. (3.3) to (3.7)

where M is number of objectives, $w_k \in [0, 1]$ is k^{th} normalized weight, f_k is the k^{th} objective function.

Pareto front/efficient frontier is a set of Pareto efficient choices from the available options. multiobjective optimization problems with conflicting goals result in a set of optimal solutions, instead of single optimal solution, because no solution can be called better than other solutions with respect to all the objectives [268]. Let P is a non-inferior solution vector of multiobjective problem, when w_1 and w_2 weights are assigned to $f_1(P)$ and $f_2(P)$ objectives, respectively. Similarly P^* is non-inferior solution of multiobjective optimization problem when w_1^* and w_2^* weights are assigned to $f_1(P)$ and $f_2(P)$ objectives respectively. A point P^* is non-inferior if there does not exist another point $P^*\chi$, such that $F(P) = F(P^*)$ and $f_1(P) < f_2(P^*)$ for at-least one objective function. As weightage w_1 decreases, the weightage, w_2 to the objective function, f_2 will increase because the weights are normalized. Since the objectives are of a conflicting nature, so the solution generated by a given weightage identifies the non-inferior solution that may provide compromised solution. A decision maker has to make trade-off within the Pareto-frontier, rather than considering the full range of every parameter.

4.4 FUZZY DECISION MAKING

The main purpose of decision maker is to select a solution, which satisfies all conflicting goals, while living up to the constraints of the system. Considering the imprecise nature of decision maker's judgment, the decision maker may have imprecise goals for each objective

function. The decision maker finds membership function, $\mu(f_k)$ in a subjective manner by taking account of minimum and maximum value of each objective function together with the rate of increase of membership functions. The membership value of objective function in a fuzzy set, indicates the satisfaction of a solution to the $k^{th} \in [0, 1]$ objective function $\mu_k(f_k)$ assumes strictly monotonically decreasing and continuous function, defined as:

$$\mu_k(f_k) = \begin{cases} 1 & ; f_k \leq f_k^{min} \\ \frac{f_k^{max} - f_k}{f_k^{max} - f_k^{min}} & ; f_k^{min} < f_k < f_k^{max} \\ 0 & ; f_k \geq f_k^{max} \end{cases} \quad (k = 1, 2, \dots, M) \quad (4.7)$$

where f_k^{min} and f_k^{max} are the extrema of k^{th} objective function, M is number of objective functions.

Fuzzy operator *max – min* evaluates the dominance level by using satisficing function mentioned below, of the generated solution by considering all the objectives simultaneously. Maximum value of satisficing function value decides the best solution.

$$\mu_C = \text{Max} \{ \text{Min} \{ \mu_k(f_k) \quad (k = 1, 2, \dots, M) \} \} \quad (4.8)$$

4.5 SOLUTION METHODOLOGY

The solution approach used in this paper to solve MOLD problem involves an interactive weighting sum method that is a modification of the conventional weighting sum method. In this method, the preference weight is integral part of search variables. The quality of preference weights is evaluated as a function of the objective function and weights are updated by APPO along with control variables of the MOLD problem. This change results in generation and modification of Pareto optimal set as the APPO search procedure proceeds. Fuzzy decision-making decides the best solution. Variable elimination method handles equality constraints and penalty method takes care of inequality constraints. The following section discusses these steps in detail.

4.5.1 Representation of Swarm

The positions of the swarm are representing the output power of generators and preference weights of objectives. Each individual position in the swarm presents the solution of the MOLD problem. The number of generating units and number of objective functions together gives the dimension of prey particles in swarm. X_i^t is the position of i^{th} individual in swarm at t^{th} move and vector X_i^t is represented as:

$$X_i^t = [P_{i1}^t, P_{i2}^t, \dots, P_{iN_g-1}^t, w_{i2}^t, w_{i3}^t, \dots, w_{iM}^t]^T \quad (i = 1, 2, \dots, N_P) \quad (4.9)$$

where $w_{i1} = 1.0 - \sum_{j=2}^M w_{ij}$ and N_P is the swarm size. N_g is the number of generators and one generator is considered slack generator so $N_g - 1$ generator and $(M-1)$ weights are considered as a swarm. The best previous position of the i^{th} particle is recorded and represented as $X_i^B = [x_{i1}^B, x_{i2}^B, \dots, x_{in}^B]^T$ and $(n = N_g + M - 2)$. The best particle position among all the particles in the group is represented by g_i^B . The velocity of particle is represented as $V_i^t = [v_{i1}^t, v_{i2}^t, \dots, v_{in}^t]^T$.

4.5.2 Initialization of Swarm and Predator Position

In APPO, the initial swarm and predator positions are randomly generated within the operational constraints of generators. In the absence of any prior knowledge about the optimal solution, the positions of prey particles and their corresponding velocities are generated randomly within their specified limits $[x_j^{min}, x_j^{max}]$ and $[v_j^{min}, v_j^{max}]$ respectively as:

$$x_{ij} = x_j^{min} + rand()(x_j^{max} - x_j^{min}) \quad (j = 1, 2, \dots, n; i = 1, 2, \dots, N_P) \quad (4.10)$$

$$v_{ij} = v_j^{min} + rand()(v_j^{max} - v_j^{min}) \quad (j = 1, 2, \dots, n; i = 1, 2, \dots, N_P) \quad (4.11)$$

where v_j^{min} and v_j^{max} values are considered as suggested by [118] and $n = N_g + M - 2$.

A second particle called predator balances exploration and exploitation. The predator po-

sition, x_{pj} and velocity v_{pj} having dimensions ($n=N_g+M-2$) are initialized randomly within the maximum and minimum limits as follows:

$$x_{pj} = x_{pj}^{\min} + rand() (x_{pj}^{\max} - x_{pj}^{\min}) \quad (j = 1, 2, \dots, N_g) \quad (4.12)$$

$$v_{pj} = v_{pj}^{\min} + rand() (v_{pj}^{\max} - v_{pj}^{\min}) \quad (j = 1, 2, \dots, N_g) \quad (4.13)$$

where x_{pj}^{\min} , x_{pj}^{\max} , v_{pj}^{\min} and v_{pj}^{\max} are the lower and upper limits on predator's position and velocity respectively.

4.5.3 Heuristic Random Search to Generate Feasible Solution

The heuristic random search strategy is applied to generate feasible solution by varying the thermal generation to satisfy the power balance constraint. The unmatched power is distributed randomly among the committed generators, while satisfying the power generation limits. The generator's power is updated as:

$$x_{ij} = x_{ij} + \frac{R_5 \alpha_i}{P_D} (x_j^{\max} - x_j^{\min}) \quad (j = 1, 2, \dots, N_g; i = 1, 2, \dots, N_P) \quad (4.14)$$

where $\alpha_i = \left(\sum_{j=1}^{N_g} x_{ij} - P_D - P_{Li} \right)$ is scaled random number.

The process is repeated until the equality constraint given by Eq. (3.3) is satisfied or $|\alpha_i| = \varepsilon$ (tolerance criteria).

4.5.4 Constraint Handling

The non-linear, scalar objective constrained optimization problem can be solved by direct and indirect methods. In direct methods, the constraints are incorporated in an explicit manner, whereas in indirect methods, the constrained optimization problem is converted into unconstrained optimization problems and is solved as an unconstrained minimization problem [472]. In case dependent generator violates an inequality constraint, it is secured to its respective minimum/maximum limit. Variable elimination method handles equality constraints and penalty method takes care of inequality constraints of slack unit.

1. **Equality constraint:** Theoretically, variable elimination method eliminates one

variable in an optimization problem with one equality constraint. The eliminated variable is called slack variable. The slack variable (generator unit) acts as a dependent variable, which is liable for maintaining power balance between generators and load demand. Let P_s be the randomly selected slack generator among committed N_g units to satisfy the energy balance constraint. The power output of the slack unit is computed from the energy balance Eq. (3.3). On simplifying Eq. (3.3) and (3.4), a quadratic equation is obtained and is written as given below:

$$B_{ss}P_s^2 + \left(\sum_{\substack{j=1 \\ j \neq s}}^{N_g} (B_{js} + B_{sj})P_j + B_{so} - 1 \right) P_s \quad (4.15)$$

$$+ \left(P_D + B_{oo} + \sum_{\substack{j=1 \\ j \neq s}}^{N_g} \sum_{\substack{j=1 \\ j \neq s}}^{N_g} P_i B_{ij} P_j + \sum_{\substack{j=1 \\ j \neq s}}^{N_g} B_{jo} P_j - \sum_{\substack{j=1 \\ j \neq s}}^{N_g} P_j \right) = 0$$

The positive, non-imaginary roots of the above quadratic equation give the value of dependent generator P_s .

2. **Inequality constraint:** The boundary conditions of slack generator is taken care by exterior penalty method, because the solution becomes infeasible on violation of limits. The augmented objective function becomes:

$$F(P) = f_k + r [\min(0, (P_s - P_s^{\max}))^2] + r_k [\min(0, (P_s^{\min} - P_s))^2] \quad (4.16)$$

where r_k is the penalty constant with sufficiently large value.

3. Ramp rate limit constraint

The generator ramp rate being an inequality constraint restricts the operating range of committed units between two adjacent periods. It is represented by Eq. (3.6) and (3.5). This constraint is undertaken by adjusting the minimum and maximum

limit of generator's power as follows [126]:

$$\max (P_j^{\min}, P_j^0 - DR_j) \leq P_j \leq \min (P_j^{\max}, P_j^0 + UR_j) \quad (j = 1, 2, \dots, N_g) \quad (4.17)$$

4. Prohibited operating zone constraint

The POZ constraints on the system are specified by the Eq. (3.7). The operation avoids the generation in these areas, for better performance of the system. A heuristic approach is applied to adjust the generator's power within a permitted range, in case the POZ limits are violated. The prohibited zone is avoided by updating the generation as described below:

$$P_j = \begin{cases} P_{j,NZ-1}^L + rand() \frac{(P_{j,NZ-1}^U - P_{j,NZ-1}^L)}{P_{j,NZ-1}^U} & ; P_j \leq P_{j,NZ}^L \\ P_{j,NZ+1}^L + rand() \frac{(P_{j,NZ+1}^U - P_{j,NZ+1}^L)}{P_{j,NZ+1}^U} & ; P_j \geq P_{j,NZ}^U \\ P_j & ; else \end{cases} \quad (j = 1, 2, \dots, N_g) \quad (4.18)$$

The power of the generator is updated repeatedly until there is no violation of prohibited zones.

4.5.5 Objective Function

Once the individual objective function's membership values are obtained (Section 4.1), the fuzzy max, min operator is employed to handle the conflicting objective functions as follows:

$$\mu_C = Max (Min\{\mu (f_k) \quad (k = 1, 2, \dots, M)\} \quad i = 1, 2, \dots, N_P) \quad (4.19)$$

The fuzzy operator evaluates the satisfaction/dominance level simultaneously achieved by all the objective functions during operation for each generated solution. Decision making process selects best-compromised solution, one having the maximum satisfaction level (μ_C) known as satisficing function value. The Eq. (4.19) is the objective function of the

problem under consideration, which is maximized by APPO algorithm.

4.5.6 Position Update of Prey and Predator

The positions of prey, represents the power generation of thermal generator and preference weights assigned to objective functions. In total there are n decision variables, such that $n = N_g + M - 2$. A population of size n is initialized stochastically. In an iterative process, the reinforcement and inhibition factors guide the movements of preys for improvement, which depends upon previous best experience and predator's presence (Section 4.2). Swarm's best position directs the predator's position. The iterative process continues until a termination criterion is satisfied. The termination criteria used in this paper is the number of swarm moves (IT^{max}). Figure 4.1 depicts a stepwise solution procedure.

4.6 SIMULATION CASES AND RESULTS

To prove the validity of the proposed solution approach to solve MOLD problem, benchmark problems related to power system operation are considered. Many researchers around the world have taken on these problems for standardized uses.

4.6.1 Test Power Systems

In order to illustrate the effectiveness of the proposed methodology, various test problems tabulated in Table 2.2 are considered for simulation and analysis. The problems considered have both equality and inequality constraints on decision variables as well as the nonlinearities due to valve point loading effect, ramp rate limits, prohibited operating zone and multiple fuels. Some case considers the transmission loss and presents the practical modeling of systems. The small power system is a multi-fuel power system with 10 generators while medium capacity power systems consist of 40 and high power generation capacity system has 140 generation units. The paper compares achieved MOLD results by the proposed method with other tested and reported population based optimization techniques in the literature.

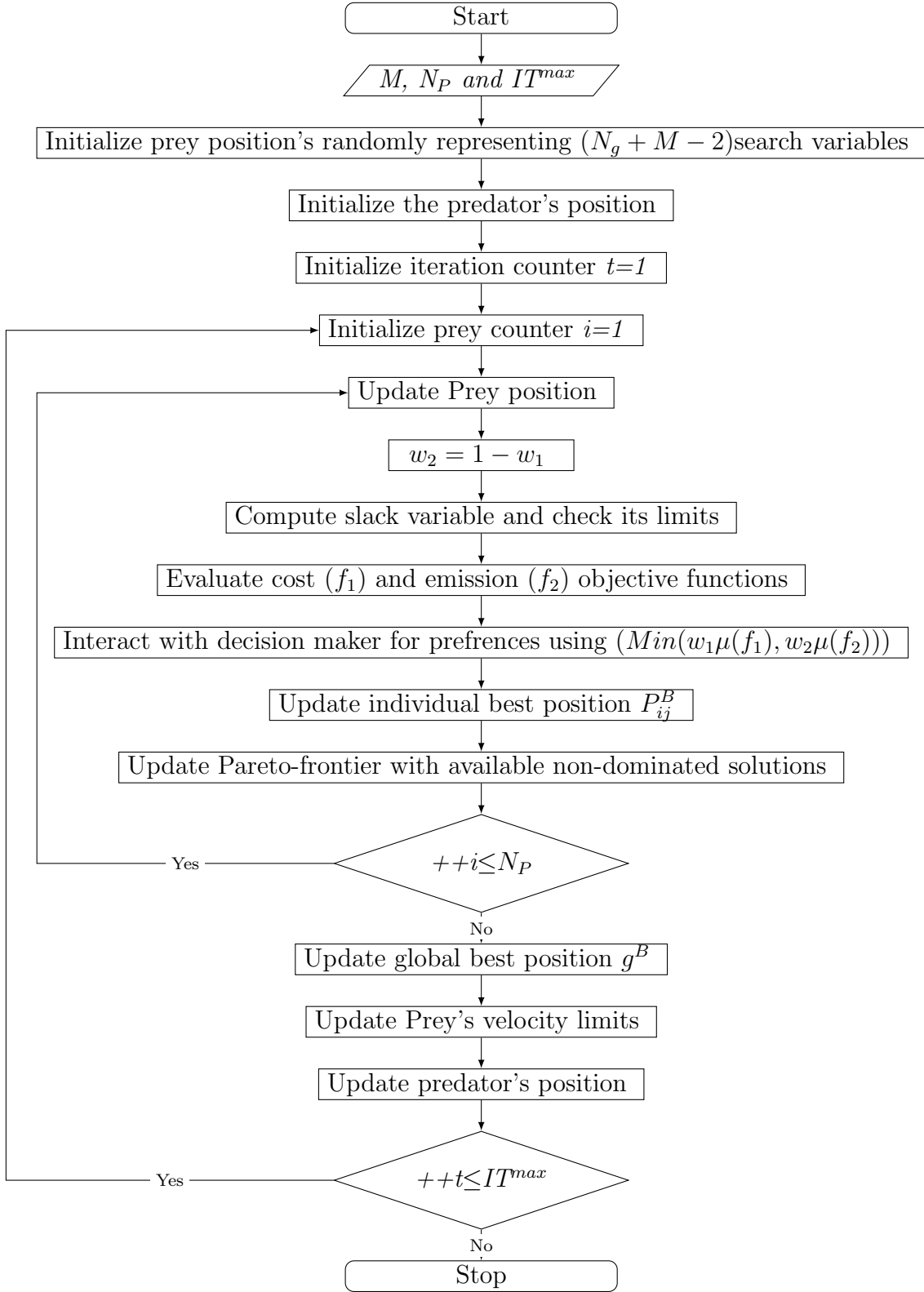


Figure 4.1: Flow chart of APPO based solution approach

4.6.2 Parameter Setting

The value of parameter c_1 and c_2 are taken as 2.0 as suggested by Kennedy *et al.* [114]. Similarly, on the basis of literature by Silva *et al.* [118] c_3 , c_p , a , b and P_f are taken 0.05, 2.0, 0.01, 100 and 0.36, respectively. These parameters remain same for all the problems undertaken to study. The swarm size and number of swarm moves are decided according to the size of the problem and are specified in the ensuing sections.

4.6.3 Test Power System-1

The first test power system has multiple fuel options for its generators. The study investigates six cases of multiple fuel power system for 2700MW power demand. To obtain the solution of the first three cases (ELD), the swarm size, N_P is set to 20 and number of swarm moves, IT^{max} is fixed at 200. The Table 4.1 shows the cost comparison of Case 1 to 3. To compare the performance, the results of *Biogeography based optimization* (BBO)

Table 4.1: Test power system-1, comparison of cost ($P_D = 2700MW$)

Algorithm	cost(\$/h)					
	Case-1	Case-2	Case-3	Case-4	Case-5	Case-6
BBO [306]	624.51	–	–			
CCPSO [321]	–	623.82	–	–	–	–
CGA-MU [373]	623.80	624.71	–	–	–	–
DE [373]	623.80	624.46	–	–	–	–
DEBBO [305]	624.51	–	–			
ELHN [369]	624.51	–	–			
IGA [163]	624.51	–	–			
IGA-MU [163]	623.80	624.51	–			
KHA [397]	624.51	–	–			
PSO[373]	623.80	624.24	–			
QP-ALHN [376]	623.80	–	624.32			
SPPO	623.80	623.82	624.32	700.29	700.77	700.48
CEPPS-1	623.75	623.87	623.76	699.70	699.54	704.94
CEPPS-2	623.75	623.88	623.77	699.77	699.73	700.60
APPO	623.80	623.82	624.32	699.30	699.40	700.41

[306], *Composite Particle Swarm Optimization* (CCPSO) [386], *Conventional Genetic Algorithm with Mutation Update* (CGA-MU) [163], *Differential Evolution* (DE) [373], *Improved Genetic Algorithm* (IGA) [163], *Improved Genetic Algorithm with multiplier update* (IGA-MU) [163], *Improved Particle Swarm Optimization* (IPSO) [373], *Hybrid of DE and*

Table 4.2: Test power system-1, Cost and emission for ELD and MED respectively

Case	ELD		MED	
	Min. Cost(\$/h)	Max. Emission(lb/h)	Min. Emission(lb/h)	Max. Cost (\$/h)
1	623.84	11834.1	105.8	691.32
2	623.85	12057.0	125.75	694.46
3	624.32	14645.46	112.3	701.0
4	699.3	39385.22	125.2	769.08
5	699.4	34766.98	125.2	762.69
6	700.41	25303.79	125.42	759.25

Table 4.3: Test power system-1, MOLD solution comparison ($P_D = 2700MW$)

Algorithm	Case-1			Case-2			Case-3		
	Cost(\$/h)	Emission(lb/h)	μ_C	Cost(\$/h)	Emission(lb/h)	μ_C	Cost(\$/h)	Emission(lb/h)	μ_C
CEPPS-1	629.49	1185.10	0.907	629.97	1263.65	0.904	629.62	1208.05	0.924
CEPPS-2	629.52	1192.03	0.907	630.15	1277.18	0.903	628.07	663.66	0.951
APPO	628.85	464.71	0.925	629.34	485.18	0.922	629.66	454.42	0.930

Table 4.4: Test power system-1, MOLD solution comparison ($P_D = 2700MW$)

Algorithm	Case-4			Case-5			Case-6		
	Cost(\$/h)	Emission(lb/h)	μ_C	Cost(\$/h)	Emission(lb/h)	μ_C	Cost(\$/h)	Emission(lb/h)	μ_C
CEPPS-1	701.31	1049.95	0.971	701.33	981.0	0.972	705.11	879.83	0.920
CEPPS-2	705.10	1129.55	0.916	704.97	972.46	0.920	704.97	3086.99	0.882
APPO	704.82	810.65	0.920	705.32	777.53	0.915	704.42	615.67	0.930

BBO (DEBBO) [386], *Krill herd optimization* (KHA) [397], *Quadratic Programming Augmented Hopfield Neural Network* (QP-ALHN) [376], *Particle Swarm Optimization* (PSO) [373], *Synergic Predator Prey Optimization* (SPPO) and *Chaotic Evolutionary Programming Pattern Search* (CEPPS) algorithms are considered. It is clear from the comparison table that the generator operating in case-1 is comparable with majority of algorithms, except CEPPS variants, which have resulted in lesser operational cost by 0.05 \$/h. In case-2, the generation cost is competitive with CCPSO, SPPO and CEPPS. Further, in case-3 the cost is comparable with SPPO and QP-ALHN and it is slightly higher than CEPPS variants. In the cases 4 to 6, the transmission losses of the power system network are considered. In these cases also APPO has obtained a better solution than its counterparts.

In order to obtain the solution of MOLD, the swarm size, N_P is set to 30 and number of swarm moves, IT^{max} is fixed at 200. Table 4.3 and 4.4 presents the comparison of the obtained compromised solution for MOLD problem using APPO for the different cases. The comparison of MOLD solution has been performed using satisficing function μ_c . The comparison reveals that the solution obtained by the APPO in case-1, case-2

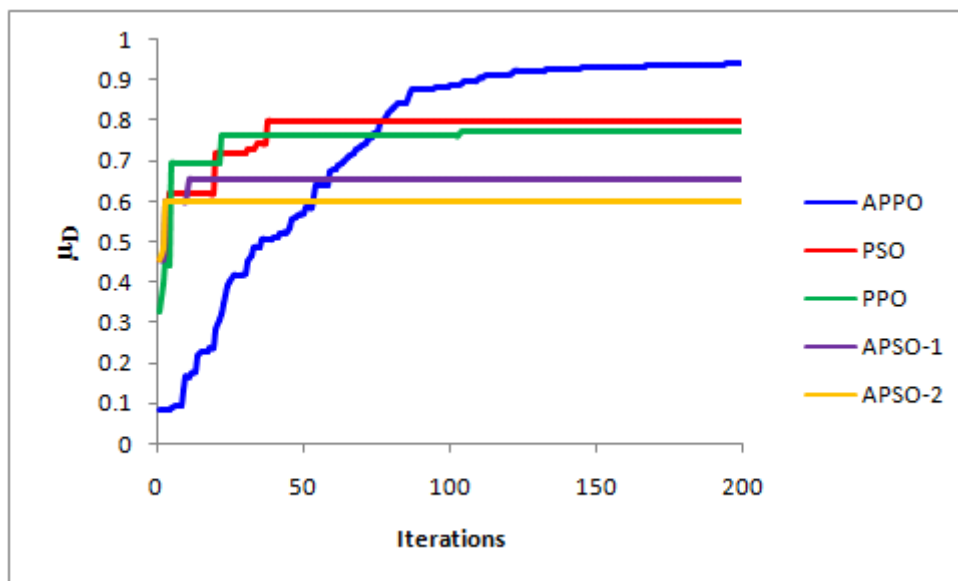


Figure 4.2: Test power system 1, case 2, Convergence characteristics comparison

and case-6 is better than the CEEPS variants, whereas in case-3, case-4 and case-5 the CEPPS-1 and CEPPS-1 respectively have provided better compromised solution. The corresponding power generation plan is presented in Table 4.5 and 4.6 respectively. The Test system-1 case-2 and case-5 being complex in nature have been used to investigate APPO's convergence behavior. The Fig. 4.2 shows the convergence plot of APPO and principally similar techniques such as PSO, PPO and APSO. The comparison of convergence characteristics shows that the APPO approaches better solution much faster than others. With regard to the robustness determination of APPO, 100 independent trials are performed to obtain the solution and the outcomes are plotted in Fig. 4.3. It is apparent from Fig. 4.3 that APPO is able to target the solution nearly 100 times. Similarly, the convergence characteristics comparison of Test system-1, Case 5 has been presented in Fig. 4.4. The plot reveals that the APPO is able to converge much faster and gives a better solution than other algorithms. Further, 100 trial runs are performed for this case to check the robustness of APPO. The Fig. 4.5 depicts the results and the success rate for APPO is 95%.

4.6.4 Test Power System-2

The Test power system-2 under investigation consists of forty generating units and 10050 MW power demand. In order to, obtain the solution N_P is set to 50 and IT^{max} is fixed at

Table 4.5: Test power system 1, MOLD Power distribution ($P_D = 2700MW$)

Generator	Case-1		Case-2		Case-3	
	Fuel type	Power(MW)	Fuel type	Power(MW)	Fuel type	Power(MW)
P_1	2	199.2408	2	201.2327	2	196.7394
P_2	1	204.5317	1	203.2297	1	203.5809
P_3	3	332.8842	3	334.7695	3	336.3827
P_4	3	239.4929	3	239.2826	3	241.8555
P_5	1	271.2275	1	267.5679	1	259.9478
P_6	3	240.5202	3	237.7867	3	242.8945
P_7	1	291.9733	1	294.4651	1	292.8182
P_8	3	237.7838	3	234.2989	3	239.4025
P_9	3	416.0605	3	415.609	3	416.3802
P_{10}	1	266.2851	1	271.7579	1	269.9986

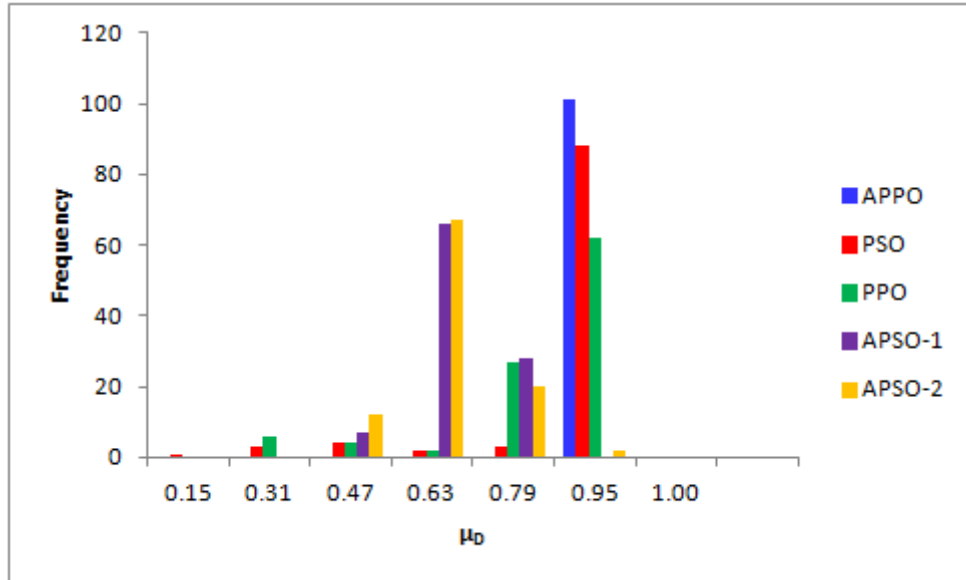


Figure 4.3: Test power system-1, case-2, Frequency plot of 100 trials

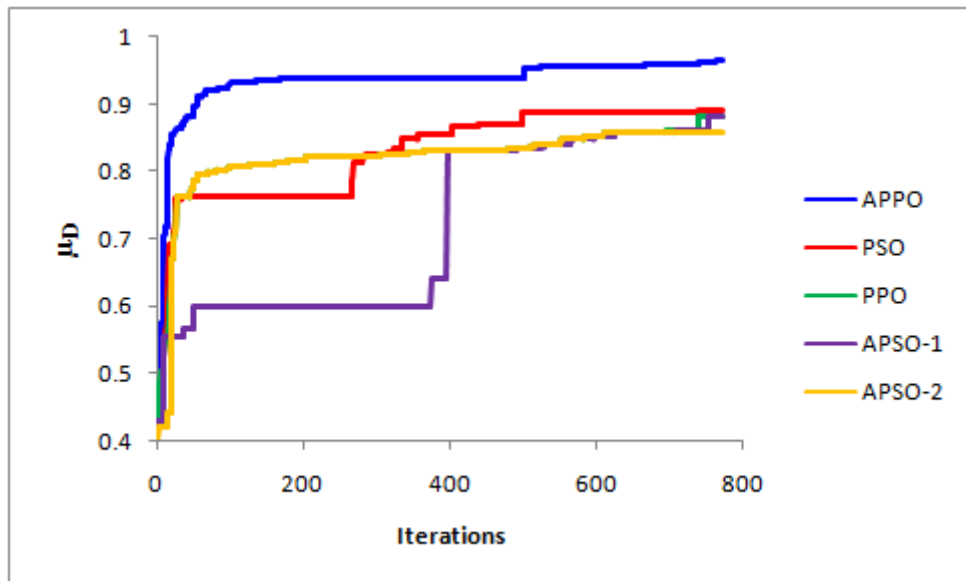


Figure 4.4: Test power system 1, case 5, Convergence characteristics comparison

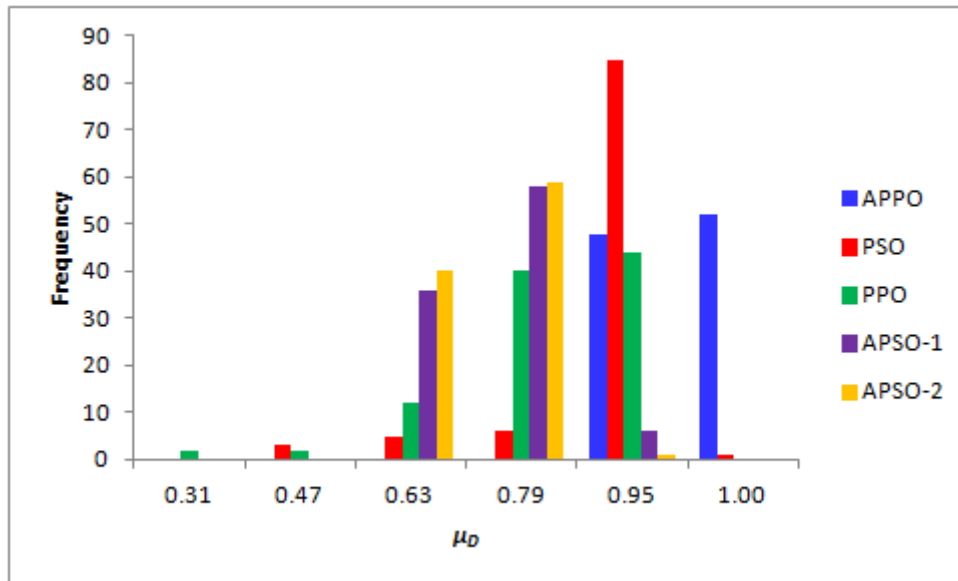


Figure 4.5: Test power system-1, Case 5 Frequency plot of 100 trials

Table 4.6: Test power system-1, MOLD Power distribution ($P_D=2700$)

Generator	Case-4		Case-5		Case-6	
	Fuel type	Power(MW)	Fuel type	Power (MW)	Fuel type	Power(MW)
P_1	2	210.8669	2	208.6998	2	205.1568
P_2	1	210.2619	1	208.415	1	213.0809
P_3	2	440.2899	2	440.6954	3	335.201
P_4	3	240.1868	3	238.749	3	241.5196
P_5	1	280.3992	1	284.9603	2	344.3734
P_6	3	240.2449	3	242.6113	3	245.4348
P_7	1	288.5124	1	285.503	1	308.2532
P_8	3	239.4828	3	239.3967	3	241.4599
P_9	3	420.6798	3	418.9945	3	436.4401
P_{10}	1	270.2317	1	273.0856	1	269.8253

Table 4.7: Test power system-2, Optimal solution (cost , emission) comparison ($P_D=10500$ MW)

Algorithm	ELD		MED		MOLD		μ_C
	Cost $\times 10^5$ \$/h	Emission $\times 10^5$ lb/h	Cost $\times 10^5$ \$/h	Emission $\times 10^5$ lb/h	Cost $\times 10^5$ \$/h	Emission $\times 10^5$ lb/h	
GSA [358]	–	–	–	–	1.25782	2.10932	0.486
LMPSO [417]	–	–	–	–	1.24668	2.19048	0.612
MABS I [424]	1.21412	3.59901	1.29995	1.766820	1.24490	2.5656	0.551
MABS II [424]	1.21412	3.59901	1.29996	1.766820	1.24491	2.56551	0.551
MODE [330]	1.2184	3.7479	1.2996	1.7668	1.25792	2.1119	0.484
NSGA II [330]	–	–	–	–	1.25825	2.10949	0.480
PDE [330]	–	–	–	–	1.25730	2.1177	0.429
SEPA II [330]	–	–	–	–	1.25807	2.1109	0.483
SMPSO [417]	–	–	–	–	1.24660	2.17256	0.617
TLBO [384]	1.21685	3.64593	1.29953	1.766835	1.26020	2.06648	0.458
PPO	1.21413	3.60420	1.29979	1.766843	1.25750	1.99125	0.417
SPPO	1.21402	3.56724	–	–	–	–	–
CEPPS-1	1.27180	3.74117	1.34516	2.46508	1.28124	3.14471	0.211
CEPPS-2	1.21487	3.63871	1.32889	2.25013	1.28515	3.27227	0.164
APPO	1.21402	3.56724	1.29924	1.76682	1.25667	1.95515	0.499

1000. The Table 4.7 presents the comparison of the solution obtained by applying APPO with its counterparts. The results are compared with the results of *Gravitational Search Algorithm* (GSA) [358], *Non-dominated sorting Genetic Algorithm* (NSGA) [330], *Modulated Artificial Bee Colony* (MABS) [424], *Multiobjective Differential Evolution* (MODE) [330], *Modulated Particle Swarm Optimization* (LMPSO) [417], *Pareto Differential Evo-*

Table 4.8: Test system-3, MOLD power schedule by APPO ($P_D = 10500$ MW)

Unit	Power (MW)	Unit	Power (MW)	Unit	Power (MW)	Unit	Power (MW)
P_1	114.0	P_{11}	318.39	P_{21}	434.28	P_{31}	190.0
P_2	114.0	P_{12}	318.39	P_{22}	436.45	P_{32}	189.9
P_3	120.0	P_{13}	394.35	P_{23}	434.84	P_{33}	190.0
P_4	179.73	P_{14}	394.27	P_{24}	436.25	P_{34}	200.0
P_5	97.0	P_{15}	394.28	P_{25}	435.18	P_{35}	200.0
P_6	140.0	P_{16}	394.28	P_{26}	434.66	P_{36}	200.0
P_7	300.0	P_{17}	489.27	P_{27}	10.0	P_{37}	110.0
P_8	300.0	P_{18}	489.27	P_{28}	10.0	P_{38}	110.0
P_9	300.0	P_{19}	425.54	P_{29}	11.65	P_{39}	110.0
P_{10}	130.0	P_{20}	423.93	P_{30}	97.0	P_{40}	422.88

lution (PDE) [330], *Sinusoidal Modulated Particle Swarm Optimization* (SMPSO) [417], *Strength Pareto Evolutionary algorithm* (SPEA) [330], *Teacher Learner Based Optimization* (TLBO) [384], *Predator Prey Optimization* (PPO) and *Synergic Predator Prey Optimization* (SPPO) and *Chaotic Evolutionary Programming Pattern Search* (CEPPS) as

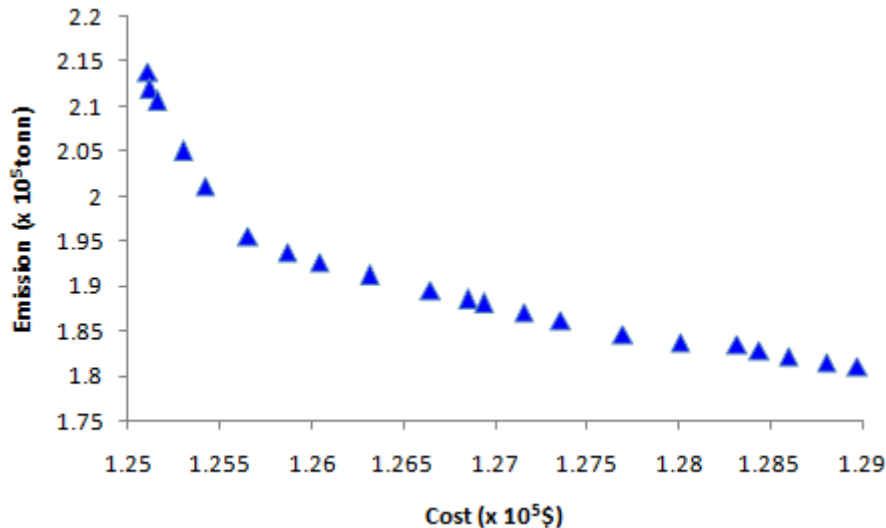


Figure 4.6: Test power system-2, Pareto front

shown in Table 4.7. The solutions obtained by using the proposed APPO approach is better than majority of algorithms considered for comparison. However, APPO has resulted in poor compromised solution in comparison to MABS, LMPSO and SMPSO. A well-distributed Pareto front has been obtained for this system and is depicted in Fig. 4.6. The MOLD power plan of compromised solution is presented in Table 4.8

4.6.5 Test Power System-3

This test power system, also called Korean power system, has 49342 MW power demand. In order to, obtain the solution swarm size, N_P is set to 60 and IT^{max} is fixed at 1000. The results of APPO have been compared with *Composite Particle Swarm Optimization* (CCPSO) [321], *Continuous Quick Group Search Optimizer* (CQGSO) [360], *Differential Harmony Search* (DHS) [388], *Group Search Optimizer* (GSO) [360], *Anti Predator Swarm Optimization* (APSO), *Predator Prey Optimization* (PPO) and *Chaotic Evolutionary Programming Pattern Search* (CEPPS) variants. The Table 4.9 presents the cost comparison of APPO and its counterparts. The comparison shows that APPO has resulted in a better quality solution than other algorithms except SPPO, which has resulted in lesser cost by 3 \$/h. Further, Table 4.10 presents comparison of ELD, MED and MOLD, from where it is evident that the APPO has obtained a better quality solution for MOLD. The MOLD compromised solution is compared using μ_c and its corresponding generation schedule has been presented in Table 4.11. For the convergence behavior comparison purpose,

the result is also obtained using principally similar techniques PSO, PPO, APSO-1 and APSO-2 and are plotted in Fig. 4.7. It is evident from plotted characteristics that APPO is capable of obtaining better optimal solution in less number of iterations. Further, from the frequency distribution table of 30 trial runs is presented in Fig. 4.8, from where it is evident that for most of the trails, the APPO algorithm is able to obtain the best solution ($\mu_D = 0.815$).

Table 4.9: Test power system 3, ELD solution comparison ($P_D = 49342MW$)

Algorithm	Cost (\$/h)
CCPSO [321]	1655685.0
CQGSO [360]	1655679.4
GSO [360]	1734405.7
PPO	1655684.0
APSO	1655684.5
SPPO	1655680.0
CEPPS-1	1778880.0
CEPPS-2	1778471.0
APPO	1655683.0

4.6.6 Robustness Analysis

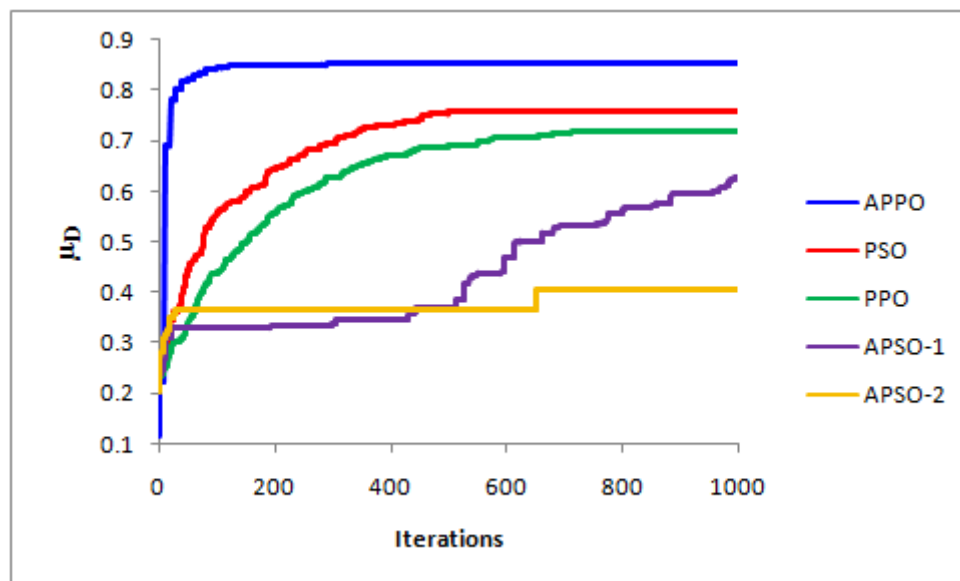
In order to, statistically verify the robustness of APPO, 30 independent trials have been performed on Test power system-3. The frequency distribution table of 30 trial runs has been plotted in Figure 4.8. It is evident that for most of the trails, the APPO algorithm is able to obtain the best solution ($\mu_D \approx 0.87$). The corresponding descriptive statistics are tabulated in Table 4.12. It is evident from the presented statistics that the APPO has obtained solutions with the minimum standard deviation.

Table 4.10: Test power system-3, MOLD solution comparison ($P_D=49342$ MW)

Algorithm	ELD		MED		MOLD		μ_C
	Cost(\$/h)	Emission(lb/h)	Cost(\$/h)	Emission(lb/h)	Cost(\$/h)	Emission(lb/h)	
SPPO	1655680	7447521	1993416	4699619	1722666	489825	0.801
CEPPS-1	1778880	6613403	1947279	5112745	1831802	5902779	0.478
CEPPS-2	1778471	6538250	1970072	5051650	1815388	5695509	0.527
APPO	1655683	7447800	1993416	4699619	1717937	4919461	0.815

Table 4.11: Test system 3, MOLD Power distribution by APPO ($P_D = 49342MW$)

Unit	Power(MW)	Unit	Power(MW)	Unit	Power(MW)	Unit	Power(MW)	Unit	Power(MW)
P_1	119.0	P_{29}	300.10	P_{57}	215.14	P_{85}	115.0	P_{113}	94.0
P_2	164.0	P_{30}	499.0	P_{58}	199.17	P_{86}	207.0	P_{114}	94.0
P_3	190.0	P_{31}	506.0	P_{59}	312.0	P_{87}	207.0	P_{115}	285.78
P_4	190.0	P_{32}	506.0	P_{60}	338.21	P_{88}	177.05	P_{116}	271.52
P_5	158.37	P_{33}	506.0	P_{61}	163.0	P_{89}	189.04	P_{117}	244.0
P_6	188.20	P_{34}	506.0	P_{62}	109.88	P_{90}	175.0	P_{118}	95.0
P_7	490.0	P_{35}	500.0	P_{63}	511.0	P_{91}	212.34	P_{119}	95.0
P_8	490.0	P_{36}	500.0	P_{64}	511.0	P_{92}	575.4	P_{120}	116.0
P_9	496.0	P_{37}	241.0	P_{65}	490.0	P_{93}	547.5	P_{121}	175.0
P_{10}	494.98	P_{38}	241.0	P_{66}	373.57	P_{94}	836.77	P_{122}	3.27
P_{11}	496.0	P_{39}	774.0	P_{67}	196.0	P_{95}	837.5	P_{123}	4.0
P_{12}	496.0	P_{40}	769.0	P_{68}	196.0	P_{96}	682.0	P_{124}	15.0
P_{13}	506.0	P_{41}	4.51	P_{69}	130.0	P_{97}	720.0	P_{125}	9.0
P_{14}	509.0	P_{42}	3.0	P_{70}	253.09	P_{98}	718.0	P_{126}	12.0
P_{15}	506.0	P_{43}	220.60	P_{71}	264.52	P_{99}	720.0	P_{127}	10.11
P_{16}	505.0	P_{44}	250.0	P_{72}	405.06	P_{100}	964.0	P_{128}	112.0
P_{17}	506.0	P_{45}	220.98	P_{73}	541.0	P_{101}	958.0	P_{129}	4.0
P_{18}	506.0	P_{46}	207.83	P_{74}	536.0	P_{102}	947.9	P_{130}	5.0
P_{19}	505.0	P_{47}	250.0	P_{75}	540.0	P_{103}	934.0	P_{131}	5.0
P_{20}	505.0	P_{48}	250.0	P_{76}	538.0	P_{104}	935.0	P_{132}	50.0
P_{21}	505.0	P_{49}	250.0	P_{77}	175.0	P_{105}	876.5	P_{133}	5.89
P_{22}	505.0	P_{50}	169.85	P_{78}	330.0	P_{106}	880.9	P_{134}	42.0
P_{23}	505.0	P_{51}	322.95	P_{79}	531.0	P_{107}	813.7	P_{135}	42.0
P_{24}	505.0	P_{52}	289.93	P_{80}	531.0	P_{108}	805.4	P_{136}	41.0
P_{25}	537.0	P_{53}	327.91	P_{81}	256.31	P_{109}	799.7	P_{137}	17.0
P_{26}	537.0	P_{54}	191.76	P_{82}	56.0	P_{110}	864.8	P_{138}	7.87
P_{27}	280.0	P_{55}	180.0	P_{83}	115.0	P_{111}	882.0	P_{139}	7.0
P_{28}	280.0	P_{56}	180.0	P_{84}	115.0	P_{112}	94.0	P_{140}	30.0

**Figure 4.7:** Test power system-3, Convergence characteristics comparison

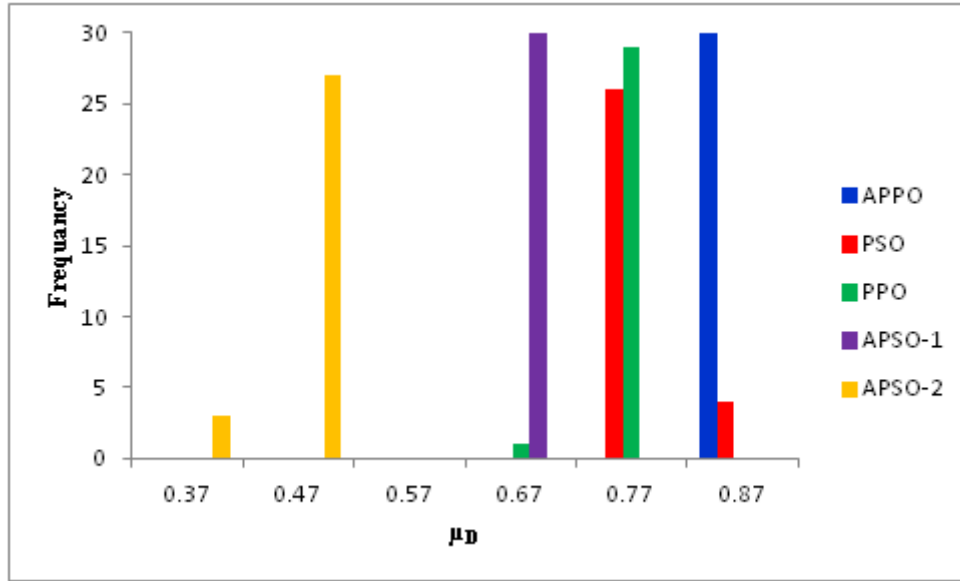


Figure 4.8: Test power system-3, Frequency plot of 30 trials

Table 4.12: Descriptive statistics Test power system-3

	PSO	PPO	APSO-1	APSO-2	APPO
Minimum	0.697660	0.669789	0.603231	0.288766	0.853623
Maximum	0.794423	0.738956	0.646727	0.458544	0.857773
Mean	0.746663	0.701053	0.626049	0.392013	0.856124
S.D.	0.026353	0.016688	0.012065	0.031905	0.001336

Table 4.13: Analysis of the parameter sensitivity Test power system-2

Parameter	C_1	C_2	C_3	C_P	a	b	P_f
Range	0 to 4.0	0 to 4.0	0 to 1.0	0 to 4.0	0 to 0.002	0 to 2.0	0 to 1
Step size	0.04	0.04	0.01	0.04	0.00002	0.02	0.01
Minimum	0.39	0.23	0.41	0.41	0.41	0.41	0.4
Maximum	0.44	0.43	0.44	0.44	0.44	0.44	0.44
Mean	0.41	0.4	0.42	0.42	0.42	0.42	0.43
S.D.	0.013	0.003	0.007	0.007	0.007	0.007	0.0073

4.6.7 Parameter sensitivity analysis

Parameter sensitivity analysis, for Test power system-2, is carried out by perturbing parameter value are by 2% of the used values. Table 4.13, presents the descriptive statistics for 30 independent trials. It is clear from the standard deviation of results that the performance of the proposed solution approach is least affected by parameter variation. An almost similar trend has been observed in statistical analysis of other investigated systems.

4.7 CONCLUSIONS

The MOLD is a constrained optimization problem having conflicting objectives *viz.* operating cost and pollutant emission. An interactive method has been applied to handle objective functions, where non-inferior solutions are generated by searching the weighting method, and decision-making has been done using fuzzy set theory. In order to avoid computational effort for obtaining complete Pareto surface, the preference weights have been searched along with generator's power using adaptive predator-prey optimization technique. The suggested method has been employed on three test power systems. The applied approach results in obtaining well distributed Pareto front in a single run. The comparison of the numerical results and the convergence characteristics of the overall membership function values confirm the effectiveness and the superiority of the proposed algorithm in majority of simulation studies. However, in some of the cases the APPO's solution is competitive with other algorithms. For smaller test power system, the best solution obtained is comparable with synergic predator prey optimization and chaotic evolutionary programming and Powell's method. Similarly, for medium power generation capacity system, the quality of compromised solution is little bit degraded in comparison to the modulated artificial bee colony and the modulated particle swarm optimization, respectively. While dealing with ELD of the high power generation system, the cost is a little bit higher than the synergic predator prey optimization, however the MOLD solution is better than its counterparts. The best solution obtained, gives better quality in terms of satisfaction of normalized membership function value. The APPO technique

shows great potential to escape from local optimum solutions due to the concept of mental ambiguity of prey. The adaptive velocity limits depending upon movements of swarm improves the convergence characteristics. Due to the robust mechanism, the method is capable of handling any number of objectives and it can be implemented on most of the practical multiobjective optimization problems.

CHAPTER 5

SURROGATE WORTH TRADE-OFF METHOD FOR MULTIOBJECTIVE LOAD DISPATCH

5.1 INTRODUCTION

In the sphere of power system generation scheduling, the *Multiobjective load dispatch* (MOLD) problem has drawn extensive attention of the research community. The optimization model and analyst's perception of a MOLD problem become more realistic with consideration of many objectives simultaneously and generally these objectives are in conflict with one another [229]. The practical MOLD problem is an optimization problem that is highly constrained with nonlinear, discontinuous and multi-modal objective functions/ goals. It is due to the valve-point loading effects, ramp-rate limits, *Prohibited Operating Zone* (POZ), spinning reserve requirements *etc.* So, the problem's solution procedure must effectively manage the system constraints and complexities while either minimizing or maximizing the objectives simultaneously.

The MOLD problem has been solved conventionally by applying golden section method [15], goal programming [24], Powell's method [27] and price penalty factor [30] *etc.* These techniques are unable to handle complexities of real world engineering problems. Alternatively, the *Multiobjective Evolutionary Algorithm* (MOEA) and hybrid algorithms have proven their superiority to solve multi-objective problems. The preferred solution approaches are *Nitshed Pareto Genetic Algorithm* (NPGA) [122], *Non-dominated sorting Genetic Algorithm* (NSGA) [335], NSGA-II [391], *Strength Pareto Evolutionary algorithm* (SPEA) [121]. Further, the MOEA based on *Particle Swarm Optimization* (PSO) [211],

Differential Evolution (DE) [328], *Harmony Search* (HS) [407], backtracking algorithm [398], chaotic bat algorithm [426], chaotic global best artificial bee colony [423] *etc.* are recently used. The supremacy of MOEA is credited to parallel processing nature, derivative autonomy and ability to handle contrary objectives simultaneously.

The Evolutionary Algorithm (EA) is an essential constituent of MOEA. Among the viable EAs, swarm based optimization technique, PSO is simple [114], memory effective [245] and computationally efficient algorithm [317]. Despite these advantages, the performance of basic PSO depends on initial swarms and tuning of parameter. PSO also lacks solution uniqueness and suffers partial optimism [197]. The population diversity is identified as the prime cause of PSO's limitations, and its remedies such as neighborhood topologies [245], parameter adaptation [276] and so on, fails to ensure universality of procedure. To overcome drawbacks of PSO, the PPO [197] consisting of preys and predator was proposed. It dynamically adjusts exploration and exploitation and has better ability to handle multi-modal problems [400]. It has been successfully applied to the engineering optimization [361] and designing problems [356]. Since, the PPO performance is parameter dependent, it is possible that during adaptive exploration-exploitation phase, PPO lacks useful diversity. A poor quality, but diverse population is less effective to locate the optimal solution [315]. The hybridization methodology merges a global search algorithm with refinement process and extracts the virtues of the parent approaches, thereby, materializing an efficient solution approach [203]. The exploration-exploitation aspects are regulated in a deterministic way, but these techniques require additional computational overhead. In short, a diverse population is able to generate better non-dominated solutions and hence ensures acceptable solution of problem in-hand.

The decision maker has to opt a solution from the noninferior solutions stipulated by MOEA, which can expeditiously attain the conflicting goals. It can be accomplished by assigning preference to the solutions using a surrogate function [93]. The *Surrogate worth trade-off* (SWT) provides an interface between the *Decision maker's* (DM) preferences and the mathematical models associated with non-dominated solutions [395]. SWT method has the advantage over weighting method proposed in the previous chapter that it does not require additional variables to be searched by the search algorithm. Among

the decision making categories, SWT belongs to the posteriori articulation of preferences. The progressive (interactive) DM iteratively analyzes the available data and update preferences and selection to accomplish conflicting objectives. The decision maker assures quality solution from available Pareto-frontier to cater conflicting ambiguous goals. The solution's quality can be better if Pareto frontier is uniformly dispersed. Among the Pareto-frontier quality measure approaches such as C-measure, D-measure [92] and U-measure, the latter is most promising grading criteria [129].

To summarize, an even-handed exploration and exploitation procedure, which deliberately maintains useful diversity in the population, is still a wide open field of research. Further, the algorithm must possess least mechanism complexity and parameter autonomy. At the same time, as the choice of objectives is ambiguous the SWT decision maker requires modification to handle uncertainty and proceed iteratively. Moreover, to verify that the algorithm has explored a wide range of feasible solutions in the objective space, the final Pareto-front needs to be investigated for uniformity in solution space.

In the light of these observations, the contributions to MOLD solution approach are: the foraging behavior decisions have been explored and a composite algorithm *Fly and Walk Predator Prey Optimization* (FWPO) is suggested as a solution approach. Among the constituents of FWPO, the fly and walk procedure progresses iteratively and deterministically ensures adaptive exploration and exploitation, resulting in a diversified population. Further, the problems' constraints are satisfied by a heuristic approach based on *change of direction*, originated from foraging behavior decisions. The fuzzy set theory oriented SWT decision making is suggested to handle non-commensurable objectives in ambiguous conditions. It advances as a progressive articulation and selects a suitable solution among noninferior solution. The performance of FWPO algorithm is analyzed for solution optimality using standard benchmark test function and hence implemented to solve MOLD problem. A quality measure procedure called the U-measure is exploited to indicate the superiority of the Pareto-frontier.

The chapter is organized into five sections. Section 5.2 presents constituents of FWPO algorithm. The proposed solution methodology is presented in section 5.3. Section 5.4 presents the performance analysis using standard test functions and MOLD problems.

Finally, the chapter is concluded in the section 5.5.

5.2 NATURAL FORAGING BEHAVIOR BASED PPO

The PSO is a popular search algorithm proposed by Kennedy and Eberhart [54] and it outperforms other evolutionary algorithms due to its simple mechanism and its computational simplicity. To further improve the performance of PSO, PPO is one of the well established approaches suitable for multi-modal problems [197]. It works well, if parameters are tuned properly [361]. In this paper, an important panorama of prey's natural foraging behavior called foraging decisions [468] constitutes the solution procedure. The important aspects of foraging behavior decisions: *fly*, *walk* and *direction of turn* are modeled as discussed in the following sections. The fly and walk are analogous to the exploration and exploitive aspects, whereas the direction of turn is related to change of foraging directions due to restrictions or constraints. The mathematical model of FWPO algorithm is elaborated as follows:

5.2.1 Prey Position Update: Fly, walk and Direction of turn model

Fly is the act of traveling a long distance (jumps) in search of food locations whereas, the walk is the activity to travel smaller distances around a location. This natural process is illustrated in Fig. 5.1. The mathematical model of FWPO algorithm is elaborated as

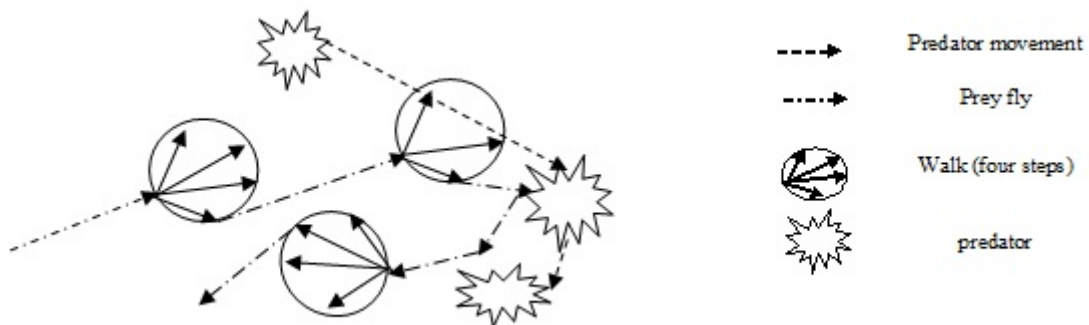


Figure 5.1: Natural foraging behavior: *fly* and *walk*

follows:

1. Fly means traveling fast over long distance. It includes prey movement guided by previous cognitive (P_{ij}^{best}) and social (g_j^{best}) experience as well as fear of predator

[197]. Mathematically, it can be represented as:

$$v_{ij}^{t+1} = wv_{ij}^t + c_1rand() (P_{ij}^{best} - P_{ij}^t) + c_2rand() (g_j^{best} - P_{ij}^t) + \gamma c_3rand() (ae^{-b\beta_i}) \quad (5.1)$$

$$P_{ij}^{t+1} = P_{ij}^t + v_{ij}^{t+1} \quad (j = 1, 2, \dots, N_g; i = 1, 2, \dots, N_P) \quad (5.2)$$

where v_{ij}^t and P_{ij}^t is prey's velocity and position at t^h interval; w is inertia weight; c_1, c_2 and c_3 are acceleration constant; $rand() \in [0, 1]$ is uniformly distributed random number; a represents the maximum amplitude of predator effect over prey at b distance; β is the euclidean distance between prey and predator; γ is a probabilistic fear controlled two state operator.

2. Walk is a movement over small distances. In the present work, it is assumed that the walk is a momentary unidirectional exploitation by an individual to attain better position. The walk is realized by perturbing the current position of prey as follows:

$$P_{ij}^t = P_{ij}^t + \begin{cases} +\Delta P_{ij}^t & ; (f(P_{ij}^t + \Delta P_{ij}^t) < \min[f(P_{ij}^t), f(P_{ij}^t - \Delta P_{ij}^t)]) \\ -\Delta P_{ij}^t & ; (f(P_{ij}^t - \Delta P_{ij}^t) < \min[f(P_{ij}^t), f(P_{ij}^t + \Delta P_{ij}^t)]) \\ 0 & ; otherwise \end{cases} \quad (5.3)$$

If the outcome of walk from current position P_j^t is a failure, perturbing factor is modified to

$$\Delta P_{ij}^t = \Delta P_{ij}^t / \alpha \quad ; |\Delta P_{ij}^t| > \epsilon \quad (5.4)$$

where α is the small pace; ϵ is the tolerance band (termination criteria).

On the other-hand, if the success being an outcome of the walk, from the current position of prey is achieved, then the learning behavior of prey is exploited to modify position.

So, the execution of the algorithm includes N_F number of flights and N_w number of walks. The fly and walk procedure switches automatically in an iterative process, thereby ensuring exploration phase shift to exploitation phase and vice-versa deterministically. This concept ascertains expedition of new potential search regions after exploitation of pre-explored areas and thus, the ultimate goal of useful diversity is assured. On the other hand predator's effect can be exploratory or exploitative, as it is changing adaptively during the fly process and ensuring faster convergence.

3. Direction of turn (constraint handling): The conventional constraint handling methods modify the problem under investigation, thereby changing its nature and hence can affect the performance of the solution procedure [461]. The alternative approach is the *direction of turn* decision from the natural foraging behavior. As per this approach, the trailing prey on recognizing a restriction in its path steers away. In general, the MOLD problem has constraints imposed by power generators and load power demand. The violation of generator boundary conditions is fixed to its respective limit. On the other hand, if the power balance constraint is unsatisfied, the unmet power is distributed among generators as per following rule:

$$\Delta = \begin{cases} \text{rand}() (P_{r_1} - P_{r_1}^{\min}) & ; (\delta < 0) \\ \text{rand}() (P_{r_1}^{\max} - P_{r_1}) & ; (\delta > 0) \end{cases} \quad (5.5)$$

$$P_{r_1} = P_{r_1} \begin{cases} +\Delta & ; (\delta < 0 \ \& \ |\delta| < \Delta \ \& \ P_{r_1} > P_{r_1}^{\min}) \\ -\Delta & ; (\delta > 0 \ \& \ |\delta| < \Delta \ \& \ P_{r_1} < P_{r_1}^{\max}) \end{cases} \quad (5.6)$$

where P_{r_1} is a randomly selected dedicated power generator, δ is power mismatch *i.e.* $|\sum_{j=1}^{N_g} P_j - P_L - P_D|$; $r_1 \in [1, N_g]$ is the direction of diversion.

4. Velocity squeezing: In order to explore the search area effectively, it becomes necessary that as the swarm moves ahead towards a solution, the complete search mechanism should help to balance exploration and exploitation. Therefore, the ve-

locity limits, which are responsible for the movement of a swarm, are adjusted in a certain fashion. It ensures that the swarm changes their position in variable step size within velocity restrictions. Therefore, velocity range reduces by the adaptive mechanism described below:

$$v_j^{max} = v_j^{max} - \delta_t \quad (j = 1, 2, \dots, N_g) \quad (5.7)$$

$$v_j^{min} = v_j^{min} + \delta_t \quad (j = 1, 2, \dots, N_g) \quad (5.8)$$

Where $\delta_t = \delta_t(m - MOD(t^{max}, m))/N_F$; $m < N_F$ and N_F is the maximum number of swarm flights.

5.2.2 Predator Position Modification

The principle approach of a predator is to chase swarm best position. The predator's velocity vp_j and position Pp_j are modified as:

$$\begin{aligned} vp_j^{t+1} &= c_p rand() (g_j^{best} - Pp_j^t) & (j = 1, 2, \dots, N_g) \\ Pp_j^{t+1} &= Pp_j^t + vp_j^{t+1} & (j = 1, 2, \dots, N_g) \end{aligned} \quad (5.9)$$

where c_P is the acceleration constant for predator.

5.3 Proposed Solution Approach

The solution approach to solve the MOLD problem, involves three main parts (i) The FWPO procedure to generate noninferior solutions (ii) Interaction with the SWT decision maker and (iii) The U-measure procedure to observe the quality of non-dominating solution set. The stepwise flowchart of solution approach is shown in Fig. 5.2 and is explained in the following sections:

5.3.1 Solution representation and initialization

The swarm position represents the power of a generator. The initial positions of the swarm and predator are randomly generated within respective limits. Each individual position in

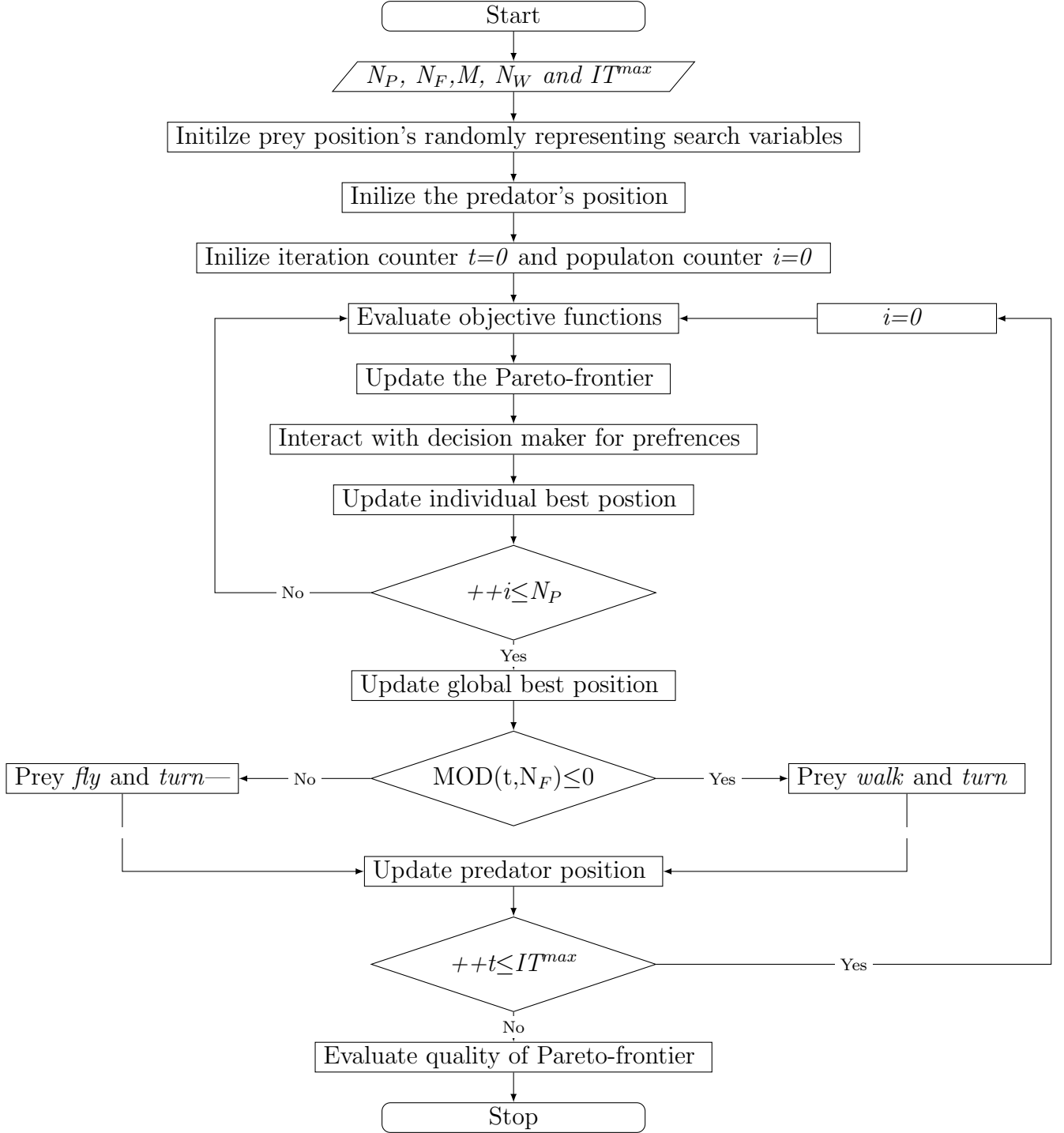


Figure 5.2: Flow chart of FWPPPO based solution approach

the population, presents the solution of the feasible space of MOLD problem. The position of i^{th} particle at t^{th} iteration is denoted by P_i^t , where $P_i^t = [P_{i1}^t, P_{i2}^t, \dots, P_{iNg}^t]$. The previous best position of the i^{th} prey particle is represented as $P_i^{best} = [P_{i1}^{best}, P_{i2}^{best}, \dots, P_{iNg}^{best}]$. The swarm's best position is represented by g_j^{best} . The predator's position is represented as $Pp^t = [Pp_1^t, Pp_2^t, \dots, Pp_{Ng}^t]$.

5.3.2 FWPO Components Update

The swarms' cognitive experience P_i^{best} is updated by comparing the presently assessed swarm positions with the previously memorized experience, based on the quality inferred by SWT function (Eq.(3.23)). The overall best position among individual's best positions is memorized as the global best experience g_j^{best} . The velocity of prey is updated using the *fly and walk* procedure described by Eqs. (5.3-5.4). In case the prey velocity violates the bounds it is fixed at respective limit. If the prey positions violate the equality constraint, the *direction of turn* starts corrective measure using Eq.(5.5-5.6), and compensate for constraint violation. In this paper, maximum number of swarm flights (N_F) is taken as the stopping criterion. On satisfying the termination criteria, the global best solution g_j^{best} is treated as solution of the problem.

5.3.3 Pareto Uniformity Measure

Leung *et al.* [96] has described the procedure used to measure the uniformity of solutions over the nonlinear hyper surface. Since the objective functions have different domains, the objectives are normalized, and the distance between any two noninferior solutions in the objective space is obtained. The discrepancy between two Pareto points can be interpreted as the deviation of distance (d_l) between two consecutive solutions from the ideal distance (d_{ideal}). The distance d_l between any two solutions in the objective space is defined as:

$$d_l(F_m, F_n) = \left[\sum_{l=1}^D \{G_l(P^m) - G_l(P^n)\}^2 \right]^{1/2} \quad l = 1, 2, \dots, D \quad (5.10)$$

where $G_l(P^m)$ and $G_l(P^n)$ are the normalized objective function values for P^m, P^n non-inferior solution, respectively.

The quality of Pareto-front indicated by (U) is the average discrepancy of all the distances *w.r.t.* ideal distance.

Mathematically,

$$U = \frac{1}{D} \sum_{l=1}^D \left| \frac{d_l}{d_{ideal}} - 1 \right| \quad (5.11)$$

where $d_{ideal} = \frac{1}{D} \sum_{l=1}^D d_l$ is the ideal distance between D pair of solutions in the frontier. The smaller the value of U , the better scattered are noninferior solutions in the problem's hyperspace.

5.4 SIMULATION CASES AND RESULTS

To prove the capability of the FWPO solution approach, the generalized benchmark test functions as well as standard real world problems related to power system operation are considered. The generalized benchmark problems specified in Table 2.1 are simulated. The performance investigation, aims to identify parameter dependency and suitable operating velocity conditions for optimality of FWPO algorithm. The small, medium and large capacity MOLD problems as shown in Table 2.2 are undertaken to verify the practicability of SWT method whereby search is performed by the FWPO algorithm to generate the non-dominated solutions.

5.4.1 Parameter Setting

The value of parameters c_1 and c_2 are taken as 2.0 [114]. Similarly c_3 , c_p , a , b and P_f are taken as 1.0, 2.0, 0.1, 10 and 0.5 [197] for all the problems. The swarm size N_P and number of swarm fly N_F are decided after performing trials and are specified in the ensuing sections. The walk operation is repeated once after every 5% of total swarm moves only, as it ensures the identification of potential foraging areas. After performing trials, the number of walks N_w is taken as 10. The comparison of obtained results performed with the results available in reported literature is presented in section 5.4.2.

5.4.2 Generalized Test Functions

Firstly, the FWPO solution approach is applied to solve seven benchmark test functions. The benchmark functions are unimodal, multi-modal, separable, non-separable and discontinuous. To investigate performance of FWPO, the best convergence behavior of PPO and FWPO *w.r.t* iterations out of 30 independent trial runs are obtained for the test function listed in Table 2.1 and are shown in Fig. 5.3 and 5.4, respectively, on logarithmic scale. The convergence behavior in Fig. 5.4 shows that in case of Griewank (f_1), Step

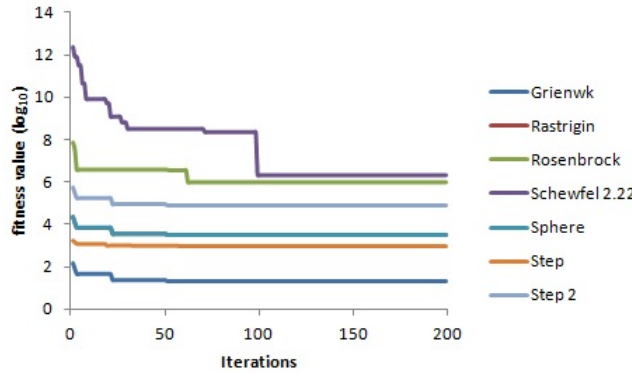


Figure 5.3: Convergence analysis of PPO using generalized benchmark problems

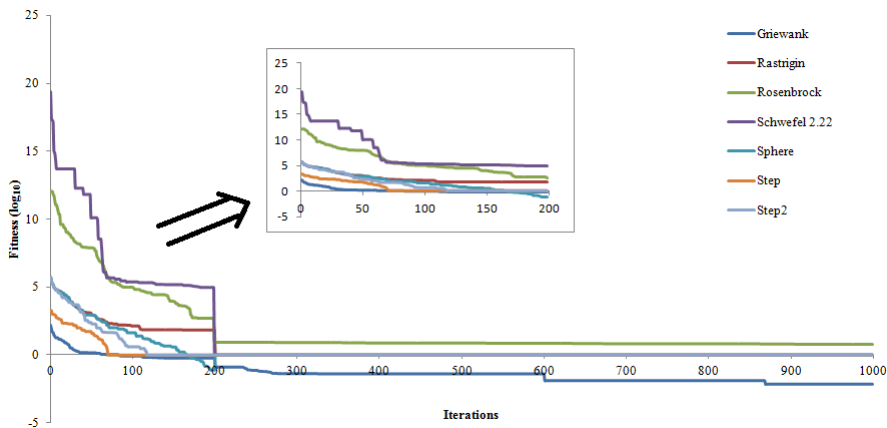


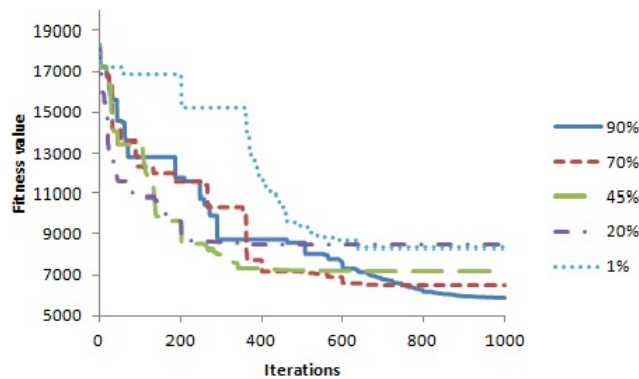
Figure 5.4: Convergence analysis of FWPO using generalized benchmark problems

function (f_6) and Step 2 function (f_7) the FWPO progresses swiftly, and achieves an optimal solution, whereas Sphere function (f_5) progresses towards the convergence point at moderate speed. For the rest of the functions FWPO is slow to converge, but achieves an optimal solution as the iterations proceed. The best, worst and mean experimental results after 1000 iterations for each test function are listed in Table 5.1. The Fig. 5.3 depicts the convergence behavior of functions obtained by implementing PPO for 200 iterations as it is unable to show any improvement after 200 iterations. It is evident from the convergence behavior comparison that the PPO is slow as well as experience premature convergence in comparison to FWPO. Further, in comparison to the synergic predator prey optimization algorithm (SPPO) results given in Table 2.4 and chaotic evolutionary programming and pattern search Table 3.1, the FWPO’s performance is better than CEPPS and comparable with SPPO.

Further, to observe the effects of prey velocity limits an experiment is performed considering Schwefel 2.26 function. The convergence behavior shown in Fig. 5.5, clearly exhibit

Table 5.1: Optimal values of generalized test functions using FWPO

Test function	Objective function value		
	Best	Worst	Mean
Griewank's function	6.7×10^{-3}	1.0	2.3×10^{-2}
Rastrigin's function	0.0	2.3×10^{-1}	5.6×10^{-2}
Rosenbrock's function	4.5×10^{-1}	6.3	1.6
Schwefel's 2.22 function	0.0	0.0	0.0
sphere function	0.0	0.0	0.0
Step function	0.0	0.0	0.0
Step-2 function	0.0	0.0	0.0


Figure 5.5: Effect of fly velocity limits variation on fitness value of Schwefel 2.26 function

dependency of behavior on velocity limits. The algorithm performs best when velocity limit is decreased by 10 percent *w.r.t.* its previous velocity after every 10 iterations. However the improvement in the solution quality has been inconsistent in all other cases. The reason of degradation is that narrower velocity limit,s restricts exploration of new foraging area by prey in fly operation. Therefore, in the rest of the experimentation the velocity limit reduction by a factor of 10% per 10 iterations is retained for prey.

5.4.3 Test Power Systems-1

Firstly, this system depicted in Table 2.1, is solved by applying FWPO for economic load dispatch (ELD). In order to obtain a solution of Test power system-1, the N_P and N_F are fixed at 15 and 200, respectively. The cost comparison is performed with other published work and is presented in Table 5.2. In order to, compare the performance, the results of *Biogeography based optimization* (BBO) [306], *Composite Particle Swarm Optimization* (CCPSO) [386], *Conventional Genetic Algorithm with Mutation Update* (CGA-MU) [163], *Differential Evolution* (DE) [373], *Improved Genetic Algorithm* (IGA) [163], *Improved Genetic Algorithm with multiplier update* (IGA-MU) [163], *Improved Particle*

Table 5.2: Test power system-1, comparison of cost ($P_D = 2700$)

Algorithm	Power generation cost(\$/h)					
	Case-1	Case-2	Case-3	Case-4	Case-5	Case-6
BBO [306]	624.51	–	–			
CCPSO [321]	–	623.82	–	–	–	–
CGA-MU [373]	623.80	624.71	–	–	–	–
DE [373]	623.80	624.46	–	–	–	–
DEBBO [305]	624.51	–	–			
ELHN [369]	624.51	–	–			
IGA [163]	624.51	–	–			
IGA-MU [163]	623.80	624.51	–			
KHA [397]	624.51	–	–			
PSO[373]	623.80	624.24	–			
QP-ALHN [376]	623.80	–	624.32			
SPPO	623.80	623.82	624.321	700.29	700.77	700.48
CEPPS-1	623.75	623.87	623.76	699.70	699.54	704.94
CEPPS-2	623.75	623.88	623.77	699.77	699.73	700.60
APPO	623.80	623.82	624.32	699.30	699.40	700.41
FWPO	623.75	623.85	624.32	699.30	699.40	700.48

Table 5.3: Test power system-1, Cost and emission for ELD and MED respectively

Case	ELD		MED	
	Min. Cost(\$/h)	Max. Emission(lb/h)	Min. Emission(lb/h)	Max. Cost (\$/h)
1	623.75	11834.1	89.17	730.31
2	623.85	12057.0	89.17	731.12
3	624.32	14645.46	107.18	701.85
4	699.32	39385.22	110.75	771.13
5	699.40	34766.98	110.75	772.02
6	700.48	25303.79	111.52	767.0

Table 5.4: Test power system-1, MOLD solution comparison ($P_D = 2700MW$)

Algorithm	Case-1				Case-2			
	Cost(\$/h)	Emission(lb/h)	μ_D	μ_C	Cost(\$/h)	Emission(lb/h)	μ_D	μ_C
CEPPS-1	629.49	1185.10	0.999	0.909	629.97	1263.65	0.999	0.90
CEPPS-2	629.52	1192.03	0.999	0.909	630.15	1277.18	0.999	0.90
APPO	628.85	464.71	0.999	0.951	629.34	485.18	0.999	0.94
FWPO	628.11	657.19	0.999	0.952	628.18	663.57	0.999	0.95

Table 5.5: Test power system-1, MOLD solution comparison ($P_D = 2700MW$)

Algorithm	Case-3				Case-4			
	Cost(\$/h)	Emission(lb/h)	μ_D	μ_C	Cost(\$/h)	Emission(lb/h)	μ_D	μ_C
CEPPS-1	629.62	1208.05	0.999	0.92	701.31	1049.95	0.999	0.97
CEPPS-2	628.07	663.66	0.999	0.95	705.10	1129.55	0.998	0.91
APPO	629.66	454.42	0.999	0.93	704.82	810.65	0.998	0.92
FWPO	628.54	671.42	0.999	0.95	702.24	2606.09	0.999	0.93

Swarm Optimization (IPSO) [373], Hybrid of DE and BBO (DEBBO) [386], Krill herd optimization (KHA) [397], Quadratic Programming Augmented Hopfield Neural Network (QP-ALHN) [376], Particle Swarm Optimization (PSO) [373], Synergic Predator Prey Optimization (SPPO), Chaotic Evolutionary Programming Pattern Search (CEPPS) and Adaptive Predator Prey Optimization (APPO) algorithms are considered. The comparison of results shows that the FWPO is able to obtain generation cost either better or equal to existing generation schedules in Case-1. While, in Case-2 the power generation cost is comparable with CCPSO, SPPO and APPO with negligible difference of 0.03 \$/h. In Case-3, the FWPO has generated competitive results with APPO, whereas the cost is slightly higher than CEPPS-1 and CEPPS-2 by 0.57 \$/h. In case 4-6, the results are comparable with APPO and SPPO.

Table 5.6: Test power system-1, MOLD solution comparison ($P_D = 2700MW$)

Algorithm	Case-5				Case-6			
	Cost(\$/h)	Emission(lb/h)	μ_D	μ_C	Cost(\$/h)	Emission(lb/h)	μ_D	μ_C
CEPPS-1	701.33	981.0	0.999	0.97	705.11	879.83	0.998	0.92
CEPPS-2	704.97	972.46	0.998	0.92	704.97	3086.99	0.999	0.90
APPO	705.32	777.53	0.997	0.91	704.42	615.67	0.998	0.93
FWPO	702.37	2616.66	0.999	0.92	703.83	3585.38	0.999	0.88

Further, the comparison of MOLD is performed with Chaotic Evolutionary Programming Pattern Search (CEPPS) and Adaptive Predator Prey Optimization (APPO) using the satisficing function μ_c (Eq. (4.19)) as shown in Table 5.4-5.6. In Case 1-3, The FWPO has resulted in better solution than its counterparts. Whereas, in case 4-5 the FWPO re-

Table 5.7: Test power system 1, MOLD Power distribution ($P_D = 2700MW$)

Case-1			Case-2		Case-3	
Generator	Fuel type	Power(MW)	Fuel type	Power(MW)	Fuel type	Power(MW)
P ₁	2	199.24	2	201.23	2	231.98
P ₂	1	204.53	1	203.22	1	209.92
P ₃	3	332.88	3	334.76	3	335.00
P ₄	3	239.49	3	239.28	3	237.53
P ₅	1	271.22	1	267.56	1	259.99
P ₆	3	240.52	3	237.78	3	238.18
P ₇	1	291.97	1	294.46	1	282.92
P ₈	3	237.78	3	234.29	3	238.88
P ₉	3	416.06	3	415.60	3	414.42
P ₁₀	1	266.28	1	271.75	1	269.14

Table 5.8: Test power system-1, MOLD Power distribution ($P_D=2700$)

Case-4			Case-5		Case-6	
Generator	Fuel type	Power(MW)	Fuel type	Power (MW)	Fuel type	Power(MW)
P ₁	2	210.86	2	208.69	2	205.15
P ₂	1	210.26	1	208.41	1	213.08
P ₃	2	440.28	2	440.69	3	335.20
P ₄	3	240.18	3	238.74	3	241.51
P ₅	1	280.39	1	284.96	2	344.37
P ₆	3	240.24	3	242.61	3	245.43
P ₇	1	288.51	1	285.50	1	308.25
P ₈	3	239.48	3	239.39	3	241.45
P ₉	3	420.67	3	418.99	3	436.44
P ₁₀	1	270.23	1	273.08	1	269.82
P _L (MW)		141.1565		141.1108		140.7454

sults are competitive with the CEPPS-1 and CEPPS-2. The corresponding MOLD power distribution plan is tabulated in Table 5.7 and 5.8 respectively.

5.4.4 Test Power System-2

This test system has 40 generators and to obtain the solution N_P and N_F are considered as 15 and 2000, respectively. The optimal solution comparison for ELD, MED and MOLD are presented in Table 5.9. The results are compared with the results of *Differential Evolution* (DE) [392], *Gravitational Search Algorithm* (GSA) [358], *Kinetic Gas Molecule Optimizer* (KGMO) [430], *Modulated Artificial Bee Colony* (MABS) [424], *Multiobjective Differential Evolution* (MODE) [330], *Modulated Particle Swarm Optimization* (LMPSO) [417], *Non-dominated sorting Genetic Algorithm* (NSGA) [330], *Pareto Differ-*

Table 5.9: Test power system-2, Optimal solution comparison ($P_D=10500$ MW)

Algorithm	ELD		MED		MOLD		μ_D
	Cost $\times 10^5$ \$/h	Emission $\times 10^5$ lb/h	Cost $\times 10^5$ \$/h	Emission $\times 10^5$ lb/h	Cost $\times 10^5$ \$/h	Emission $\times 10^5$ lb/h	
DE[392]	1.21442	–	–	–	–	–	–
GSA [358]	–	–	–	–	1.25782	2.10932	0.879
KGMO [430]	1.21378	–	–	–	–	–	–
LMPSO [417]	–	–	–	–	1.24668	2.19048	0.975
MABS I [424]	1.21412	3.59901	1.29995	1.766820	1.24490	2.5656	0.990
MABS II [424]	1.21412	3.59901	1.29996	1.766820	1.24491	2.56551	0.991
MODE [330]	1.21840	3.7479	1.2996	1.76682	1.25792	2.1119	0.879
MPSO [411]	1.21379	–	–	–	–	–	–
NSGA II [330]	–	–	–	–	1.25825	2.10949	0.874
PDE [330]	–	–	–	–	1.25730	2.1177	0.886
QTLBO [384]	1.21428	3.53417	1.29955	1.766825	1.25161	2.06648	0.922
SEPA II [330]	–	–	–	–	1.25807	2.1109	0.877
SMPSO [417]	–	–	–	–	1.24660	2.17256	0.973
TLBO [384]	1.21685	3.64593	1.29953	1.766835	1.26020	2.06648	0.885
PPO	1.21401	3.60420	1.29979	1.766843	1.25750	1.99125	0.849
SPPO	1.21402	3.56724	–	–	–	–	–
CEPPS-1	1.27180	3.74117	1.34516	2.46508	1.28124	3.14471	0.986
CEPPS-2	1.21487	3.63871	1.32889	2.25013	1.28515	3.27227	0.996
APPO	1.21402	3.56724	1.29924	1.76682	1.25667	1.95515	0.848
FWPO	1.21374	3.60018	1.29924	1.76682	1.24666	2.47257	1.0

ential Evolution (PDE) [330], Quasi Teacher Learner Based Optimization (QTLBO) [384], Real Coded Chemical Reaction Optimization (RCCRO), Sinusoidal Modulated Particle Swarm Optimization (SMPSO) [417], Strength Pareto Evolutionary algorithm (SPEA) [330], Teacher Learner Based Optimization (TLBO) [384], Predator Prey Optimization (PPO) and Synergic Predator Prey Optimization (SPPO). Synergic Predator Prey Optimization (SPPO) and Chaotic Evolutionary Programming Pattern Search (CEPPS) as shown in Fig. 5.9. The ELD solution comparison with other algorithm's results from the literature shows that FWPO is able to obtain comparable solution with others. In 30 independent trials, the minimum cost, mean cost and maximum cost for ELD is obtained as 121374.0, 121376.0 and 121379.0 respectively. Similarly, the MED solution is also of the same quality to its counterparts. Further, MOLD solution comparison on the basis of μ_D shows that the solution generated by FWPO is better than the solution obtained by the other approaches. The Pareto front comparison of FWPO and PPO is shown in Fig. 5.6. The FWPO Pareto front is uniformly distributed in the solution space. The optimal generation schedule for ELD, MED and MOLD is presented in Table 5.10.

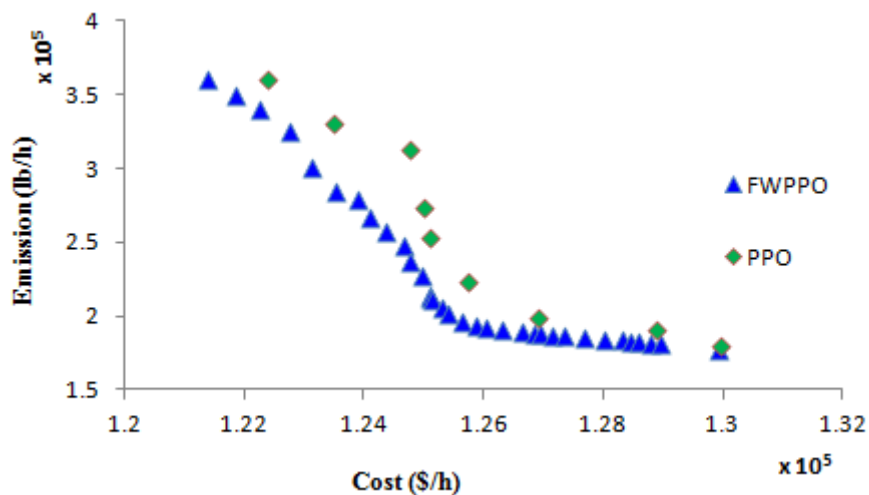


Figure 5.6: Test system-2, Pareto front using FWPO and PPO

5.4.5 Test Power System-3

This is a Korean power system having 140 power generators. As the system is of a complex nature, it is rigorously experimented under different conditions. To obtain the solution in all the cases N_F is fixed at 5000. Firstly, the ELD solution of this problem has been obtained using FWPO and is compared with other available solutions from literature as shown in Table 5.11. The results of FWPO are compared with *Composite Particle Swarm Optimization* (CCPSO) [321], *Continuous Quick Group Search Optimizer* (CQGSO) [360], *Group Search Optimizer* (GSO) [360], *Anti Predator Swarm Optimization* (APSO), *Predator Prey Optimization* (PPO), *Chaotic Evolutionary Programming Pattern Search* (CEPPS) and *Adaptive Predator Prey Optimization* (APPO). The cost comparison, shown in Table 5.11 proves the dominance of the FWPO algorithm in terms of minimum operating cost. The corresponding optimal generation schedule is tabulated in Table 5.13.

In order to, compare the MOLD solution obtained by FWPO, the problem is also solved with principally similar techniques such as PSO, PPO, APSO, APPO and CDEPS are considered. Surrogate worth trade-off value μ_D is used to compare the quality of comprised solution. The result comparison depicted in Table 5.12 shows the corresponding objective function values and μ_D . On basis of μ_D , it is concluded that FWPO has obtained a better compromised solution. The optimal power distribution plan for ELD, MED and MOLD cases are shown in Table 5.13.

Table 5.10: Test power system-2, Power schedule plan for ELD, MED and MOLD ($P_D=10500$ MW)

Unit	ELD	MED	MOLD	Unit	ELD	MED	MOLD
P_1	110.82	114.0	114.0	P_{21}	523.5298	439.38	435.06
P_2	110.80	114.0	114.0	P_{22}	523.28	439.65	434.51
P_3	97.40	120.0	120.0	P_{23}	523.32	439.04	442.42
P_4	179.73	169.73	179.69	P_{24}	523.28	439.44	438.47
P_5	87.79	97.0	97.0	P_{25}	523.27	440.11	502.47
P_6	140.0	123.96	140.0	P_{26}	523.3	440.95	496.37
P_7	259.6	300.0	300.0	P_{27}	10.0	28.98	10.0
P_8	284.6	297.88	300.0	P_{28}	10.0	28.63	10.0
P_9	284.6	297.05	300.0	P_{29}	10.0	28.34	10.0
P_{10}	130.0	130.0	130.0	P_{30}	87.8	97.0	97.0
P_{11}	94.0	298.63	187.0	P_{31}	190.0	172.26	184.0
P_{12}	94.0	297.89	171.99	P_{32}	190.0	172.30	190.0
P_{13}	214.75	433.03	305.11	P_{33}	190.0	172.59	185.59
P_{14}	394.27	423.54	394.95	P_{34}	164.8	200.0	200.0
P_{15}	394.28	422.61	394.2	P_{35}	194.36	200.0	200.0
P_{16}	394.27	421.61	395.23	P_{36}	200.0	200.0	200.0
P_{17}	489.28	440.05	489.18	P_{37}	109.97	100.96	110.0
P_{18}	489.27	439.52	489.05	P_{38}	109.99	101.19	110.0
P_{19}	511.27	438.71	507.91	P_{39}	109.82	100.49	110.0
P_{20}	511.27	440.27	503.72	P_{40}	511.28	438.97	500.98

Table 5.11: Test power system-3, Optimal cost comparison ($P_D=49342$ MW)

Algorithm	Cost(\$/h)		
	Minimum	Mean	Maximum
CCPSO [321]	1655685.0	1655685.0	1655985.0
CTPSO [321]	1655685.0	1655685.0	1655985.0
CQPSO [360]	1655679.4	1655679.4	1655679.4
GSO [360]	1734405.7	1739400.8	1745315.0
PPO	1655690	1655694.0	1657002.0
FWPO	1655674.0	1655677.2	1655683.0

Table 5.12: Test power system-3, MOLD compromised solution comparison

Algorithm	Cost (\$/h), $f_1(P_j)$	Emission (lb/h), $f_2(P_j)$	μ_D
PSO	1776898	5147131	0.612
APSO-1	1780385	5648351	0.601
APSO-2	1780385	5648351	0.601
PPO	1770809	4940244	0.632
CEPPS-1	1831802	5902779	0.965
CEPPS-2	1815388	5695509	0.962
APPO	1717937	4919461	0.988
FWPO	1704270	5160329	1.0

5.4.6 Quality Measure Analysis

The multi-objective problem solution procedure results in a set of non-dominated solutions. When these solutions are uniformly scattered over the Pareto-frontier in objective space, the quality of solution is better. An analysis of test system-3 to observe uniformity measure of the Pareto-frontier in independent trials is performed. The trial outcomes are shown in Fig. 5.7, which depicts that the quality of Pareto-front is closely maintained in

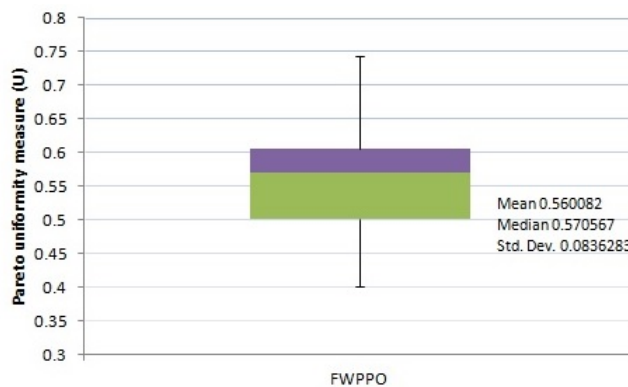


Figure 5.7: Test system-3, FWPO Pareto front's uniformity analysis

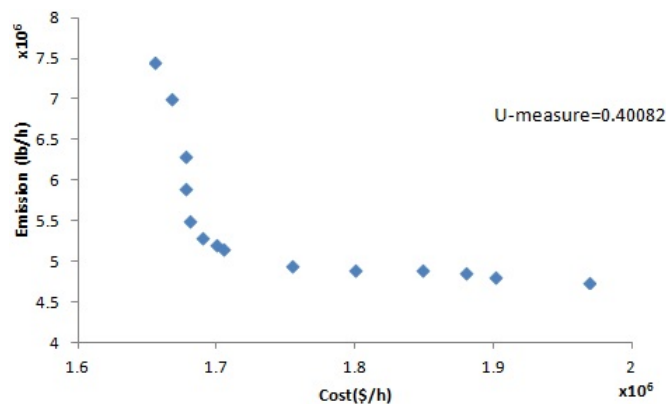


Figure 5.8: Test system-3, Pareto-frontier by applying FWPO

a small range by proposed SWT method and a search is performed by FWPO algorithm. The most uniformly distributed Pareto front is shown in Fig. 5.8. So, it can be concluded that the algorithm is able to retain the quality of Pareto-front and ensures the better optimal solution.

5.4.7 Sensitivity Analysis of FWPO Algorithm

1. Population size: To observe the effect of N_P on FWPO's performance, test power system-3 is investigated. Table 5.14 shows the outcomes of experimental study un-

Table 5.13: Test power system-3, Optimal power schedule plan ($P_D=49342$ MW)

Unit	ELD	MED	MOLD	Unit	ELD	MED	MOLD	Unit	ELD	MED	MOLD	Unit	ELD	MED	MOLD
1	119.0	88.6	119.0	36	500.0	465.1	500.0	71	137.0	313.6	204.8	106	880.9	825.3	880.9
2	164.0	164.0	164.0	37	241.0	131.1	241.0	72	365.7	454.1	386.6	107	873.7	813.7	813.7
3	190.0	175.1	190.0	38	241.0	175.5	241.0	73	195.0	535.6	493.6	108	877.4	805.4	805.4
4	190.0	125.2	190.0	39	774.0	423.0	774.0	74	229.9	524.1	536.0	109	871.7	799.7	799.7
5	190.0	174.1	190.0	40	769.0	526.8	769.0	75	230.5	540.0	540.0	110	864.8	795.7	864.8
6	190.0	139.6	190.0	41	3.0	19.0	3.0	76	253.3	523.3	538.0	111	882.0	880.7	882.0
7	490.0	344.4	490.0	42	3.0	22.1	3.0	77	390.5	245.1	198.3	112	94.0	202.0	94.0
8	490.0	465.2	490.0	43	250.0	169.5	250.0	78	330.0	344.2	330.0	113	94.0	203.0	94.0
9	496.0	478.2	496.0	44	250.0	245.3	250.0	79	531.0	213.4	530.9	114	94.0	200.0	94.0
10	496.0	260.0	496.0	45	250.0	216.8	250.0	80	531.0	394.1	531.0	115	244.0	310.8	244.0
11	496.0	286.9	496.0	46	250.0	160.7	250.0	81	542.0	486.6	508.0	116	244.0	365.5	244.2
12	496.0	287.7	496.0	47	250.0	250.0	250.0	82	56.0	60.5	56.0	117	244.0	270.3	244.0
13	506.0	495.3	506.0	48	250.0	165.2	250.0	83	115.0	238.5	115.0	118	95.0	109.7	95.0
14	509.0	371.1	509.0	49	250.0	216.4	250.0	84	115.0	233.5	115.0	119	95.0	124.8	95.0
15	506.0	323.7	506.0	50	250.0	160.0	204.2	85	115.0	190.7	115.0	120	116.0	188.4	116.0
16	505.0	318.8	505.0	51	165.0	490.5	300.8	86	207.0	222.9	207.0	121	175.0	321.0	175.0
17	506.0	494.8	506.0	52	165.0	504.0	238.3	87	207.0	299.0	207.0	122	2.0	10.2	2.0
18	506.0	397.5	506.0	53	165.0	265.8	239.5	88	175.0	345.0	175.0	123	4.0	22.9	4.0
19	505.0	497.3	505.0	54	165.0	307.7	165.0	89	175.0	331.9	175.0	124	15.0	82.9	15.0
20	505.0	479.9	505.0	55	180.0	381.3	180.0	90	178.8	175.0	175.0	125	9.0	53.0	9.0
21	505.0	378.4	505.0	56	180.0	294.7	180.0	91	175.0	345.0	175.0	126	12.0	27.1	12.0
22	505.0	297.2	505.0	57	103.0	314.3	107.1	92	575.4	541.6	575.4	127	10.0	25.8	10.0
23	505.0	505.0	505.0	58	198.0	608.9	198.0	93	547.5	517.2	547.5	128	112.0	373.0	112.0
24	505.0	498.9	505.0	59	312.0	312.0	312.0	94	836.8	795.0	836.8	129	4.0	6.9	4.0
25	537.0	518.9	537.0	60	308.1	471.0	316.1	95	837.5	796.0	837.5	130	5.0	36.1	5.0
26	537.0	530.0	537.0	61	163.0	500.0	163.0	96	682.0	578.0	682.0	131	5.0	19.0	5.0
27	549.0	280.0	347.2	62	95.0	302.0	95.0	97	720.0	621.9	720.0	132	50.0	98.0	50.0
28	549.0	280.0	347.4	63	511.0	511.0	511.0	98	718.0	663.8	718.0	133	5.0	6.5	5.0
29	501.0	300.1	356.68	64	511.0	511.0	511.0	99	720.0	706.8	720.0	134	42.0	52.2	42.0
30	499.0	350.9	499.0	65	490.0	476.7	490.0	100	964.0	884.4	964.0	135	42.0	53.6	42.0
31	506.0	506.0	506.0	66	256.1	439.3	287.5	101	958.0	906.7	958.0	136	41.0	105.0	41.0
32	506.0	464.6	506.0	67	490.0	196.0	196.0	102	947.9	870.8	947.9	137	17.0	49.1	17.0
33	506.0	506.0	506.0	68	490.0	196.0	196.0	103	934.0	934.0	934.0	138	7.0	11.4	7.0
34	506.0	493.8	506.0	69	130.0	130.0	130.0	104	935.0	935.0	935.0	139	7.0	16.3	7.0
35	500.0	499.1	500.0	70	294.9	130.8	295.1	105	876.5	816.8	876.5	140	26.0	27.0	26.8

Table 5.14: Swarm size (N_P) vs (cost and emission) analysis by FWPO for Test system-3

N_P	ELD		MED		MOLD		μ_D
	Cost (\$/h)	Emission (lb/h)	Cost (\$/h)	Emission (lb/h)	Cost (\$/h)	Emission (lb/h)	
5	1655677	7447880	1984933	4745941	1705836	5173913	0.839706
10	1655675	7446609	1968644	4740310	1704943	5166161	0.842572
15	1655674	7445358	1974565	4745446	1704637	5163486	0.843554
20	1655676	7446502	1971452	4747039	1704489	5162252	0.844017
25	1655675	7447630	1976630	4740598	1704301	5160611	0.844623
30	1655675	7444514	1965611	4745908	1704270	5160329	0.844727

dertaken for analysis. For ELD, MED and MOLD problem a solution is highlighted which offers minimum operating cost, minimum pollutant emission and maximum trade off (μ_D). So, the best solution for ELD is obtained for at $N_P=15$ with operating cost 1655674 \$/h. Whereas for MED the best solution is obtained at $N_P=10$ with minimum emission 4740310 lb/h. A compromised solution, for MOLD is highlighted that obtain maximum value of (μ_D) at $N_P=30$. The N_P greater than 30 is unable to show any improvement in the solution quality.

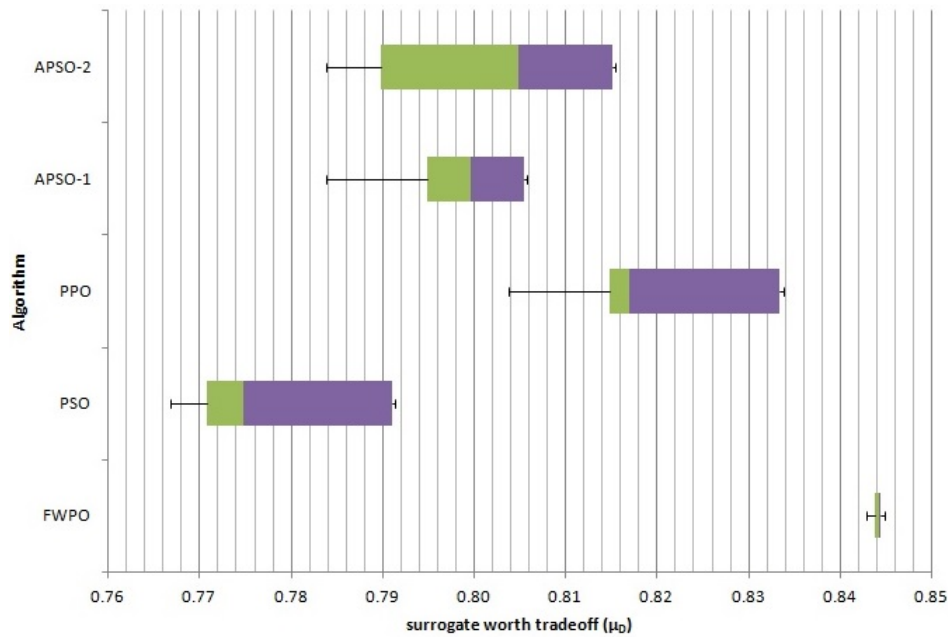


Figure 5.9: Performance comparison of FWPO algorithm

2. Robustness test: The robustness test is performed to verify the capability of the proposed FWPO approach to obtain solution. To observe parameter and initial population dependency, thirty independent trials are performed on Test system-3. The trial outcomes presented in Fig. 5.9 shows that the FWPO solutions are concentrated in a narrow region in comparison with similar nature algorithms PSO, PPO and APSO. It is observed that the mean solution is better in quality in terms of satisfaction level to both the objectives. So the outcomes of FWPO technique are more accurate, highly precise and repeatable as compared to others.

5.5 CONCLUSION

In this research work, a surrogate worth trade-off method has been implemented to decide the generation allocation of multiobjective load dispatch problem. Whereby, fly and walk predator-prey optimization (FWPO) has been implemented to generate the non-dominated solutions. The FWPO algorithm maintains useful diversity by switching exploration and exploitation phase in a cyclic process. The direction of turn which is another aspect of FWPO, bias the optimization process towards the feasible region without restricting the improvement in objective functions. The results obtained by FWPO are compared with other algorithms' result available in literature. The comparison of results

shows that in few cases of the small test power system, the FWPO solution is competitive with results of synergic predator prey optimization, adaptive predator prey optimization and chaotic evolutionary programming and Powell's method algorithm. However, for bigger systems, the FWPO results are better than its counterparts. The U-measure of Pareto-front indicates that the proposed approach is able to generate uniformly distributed non-dominated solutions for multiple objectives. The sensitivity analysis of experimental results shows that the FWPO performance is independent of the initial population and control parameters. The appropriate population size and proper velocity limits lead to a better convergence rate of FWPO. Overall, the experimental analysis shows that the SWT and FWPO algorithm reproduce better results in a narrow band, despite parameters and initial population abnormalities for single and multi-objective problems. However, a limitation of proposed surrogate worth trade-off function has been identified that in a particular situation, it is unable to convey conclusive information to select a compromised solution.

CHAPTER 6

MULTIOBJECTIVE LOAD DISPATCH USING CHAOTIC DIFFERENTIAL EVOLUTION AND POWELL'S METHOD

6.1 INTRODUCTION

The field of engineering design and planning and related areas are active field of optimization problems. These problems are often highly nonlinear and can have multiple optima. It is a challenge to solve such kind of multi-modal problems. In effect, the global optimization algorithms have been widely attempted as solution procedure. The traditional algorithms may not produce good results, and the latest trend is to use new meta-heuristic algorithms [479]. Effective and efficient optimization algorithms are always needed to tackle increasingly complex real world optimization problems.

Stochastic meta-heuristic algorithms attempt to mimic natural phenomena or social behavior so as to generate better solutions for the optimization problem by using a stochastic iterative process [475]. The commonly used meta-heuristic techniques are listed in Table 1.1. These swarm intelligence based algorithms are commonly inspired by observing natural swarming behavior or physical phenomenon. They have exhibited good performance to solve many engineering real-world problems. Despite many efforts invested so far, the aforementioned algorithms still face some challenges and disadvantages in their utilization. For example, *Particle Swarm Optimization* (PSO) exhibits slow convergence speed and it is liable to suffer from the premature convergence problem when addressing multi-modal problems. *Gravitational Search Algorithm* (GSA) and *Charged System*

Search (CSS) are often time-consuming. Some algorithms like *Artificial Bee Colony* (ABC) and *Bacterial Foraging Optimization* (BFO) need to tune several control parameters to maintain the balance between local search and global search. Others manage to achieve the preservation of population diversity at the cost of slow convergence or complicated algorithmic structures. So far, none of the heuristic search algorithms is capable of offering adequately high performance to solve all optimization problems. Consequently, it remains a challenge to develop a population-based heuristic search algorithm that is able to prevent premature convergence and meanwhile keep the fast converging feature.

The recent study concentrates on *Differential Evolution* (DE) based multiobjective solution approaches. DE, proposed by Storn and price (1995) [474], in a short span of time, has emerged as a simple and efficient solution technique for a variety of problems. Further, DE is extended to solve multiobjective optimization problems and the significant work includes *Pareto Differential Evolution* (PDE) [99], *Multiobjective Differential Evolution* (MODE) [137], DE for multiobjective optimization [180], multiobjective self-adaptive DE [281] and *Modified Differential Evolution* [353]. The ability of above summarized approaches, to generate non-inferior solutions depend upon exploration capability of DE. However, similar to other evolutionary algorithms, basic DE algorithm also faces some challenging problems. Despite, fast and reasonable solution, sometimes it does not guarantee a global optimal solution. The improved DE has also faster convergence, but often it has a probability of searching towards a local optimum or getting premature convergence. This drawback can be overcome by a larger population size, but it results in excessive memory requirements and large computational time [340]. The DE algorithm stagnates at local optimal even if the population is diverse [279].

It is a well-accepted fact that both exploration and exploitation are essential aspects of evolutionary algorithms. The two abilities of the algorithm must be in a perfect balance to ensure useful diversity in the population. In this respect, the hybrid methods consist of global search technique as exploratory phase and deterministic methods as exploitation phase, which helps to locate the global optimum solution. The Powell's conjugate direction method is the most successful and popular direction search method. It has the ability to solve problems with non-quadratic function and requires N^2 unidirectional searches to

minimize quadratic functions [462]. Nevertheless, these procedures results in an improvement in achieving the solution, but it is at the cost of additional complexity to the basic algorithm. Further, the selection of optimal parameters to improve the performance of hybrid algorithm is another critical issue to investigate [340].

On the other hand, the application of non-linear dynamics has recently drawn attention in the field of optimization. One of the application is the use of chaos map to select algorithm-dependent parameters [368]. In the literature, the chaotic sequence is used to tune parameters of PSO, DE, *Harmony Search* (HS), *Firefly Algorithm* (FA) *etc.* and these applications have shown positive improvement in the performance of algorithm [340, 271, 344, 279]. The empirical studies have shown that chaotic sequences have a high level of mixing capability and thus it is expected that when these, are used to replace fixed procedures, can result in a better balance of exploration and exploitation. So, in the context of improving DE's performance, a useful diversity can be ensured by combining deterministic chaotic sequence with DE [340].

The second important aspect of *Multiobjective Evolutionary Algorithm* (MOEA) is handling of multiple objective functions. As discussed in section 1.3 the non-interactive and interactive techniques are used to take care of the objectives. In this chapter, a scaling function is proposed that can be used as non-interactive and interactive procedure. A no-preference approach (non-interactive procedure) is adopted to reduce the computational complexities due to generation of Pareto-front and selection of satisficing solution. The proposed surrogate function is used as a satisficing function to resolve the conflict of non-commensurable objectives. The proposed function has the responsibility to select suitable solution, which is acceptable to all the objective functions.

In an interactive approach, the *Decision maker* (DM) has to select a solution from the non-inferior solutions stipulated by MOEA, which can expeditiously attain the conflicting goals. The scaling function assigns preference to the non inferior solutions [93], which helps to decide suitable non-inferior solution. The *Surrogate worth trade-off* (SWT) provides an interface between the DM's preferences and the mathematical model associated with non-dominated solutions [395]. Among the decision-making categories, SWT belongs to the posteriori articulation of preferences. The progressive DM iteratively analyzes the

available data, update preferences and selects quality solution to accomplish conflicting objectives. A uniformly dispersed Pareto frontier results in a better quality of solution.

The intent of the chapter is to use *Chaotic Differential Evolution and Powell's method* (CDEPS) approach to generate acceptable solution of the problem. The stochastic behavior of DE has been established using chaotic sequence available from the literature. The present work considers two maps, namely Gauss map and Tent map, to investigate its effect on CDEPS's performance and compatibility with SWT approach. The Powell's pattern search is applied to enhance the exploitation of proposed solution procedure. The conflicting objectives are handled by the scaling function used as non-interactive and interactive approach. The standard benchmark problems are considered to evaluate the performance of CDEPS variants. The case studies of electrical power system networks having small, medium and high power generation capacity as mentioned in Table 2.1 have been simulated.

The chapter is organized into five parts. Section 6.2, discuss conceptual details of CDEPS algorithm. In section 6.3 presents the CDEPS based solution approach to solve *Multiobjective load dispatch* (MOLD) problem is conversed. Section 6.4, presents numerical results by the proposed method and compare it with some of the most recently published work. Finally, Section 6.5 concludes the chapter.

6.2 PROPOSED CDEPS ALGORITHM

The proposed solution approach clubs *Chaotic Differential Evolution* (CDE) approach with *Powell's pattern search* (PS), to optimize the generation schedule of MOLD problem. The CDE aims to explore the global solution, whereas PS focuses on the exploitation of the search area. The PS phase remains normally in hibernation and is activated by chaotic process. These aspects are explained in the ensuing subsections:

6.2.1 Chaotic Differential Evolution

The principle idea of DE is to select a better quality solution among trial solutions, generated on the basis of the existing population. The mutation and the crossover are the main trial solution generation operators and the selection operation decides sustainability

of these solutions for future generations. Basically, all the operators of conventional DE are based on uniformly distributed random numbers, whereas in CDE the random numbers are generated by randomly initialized preselected chaotic sequence. The well accepted chaotic map namely, Gauss map and Tent map are used for randomization in the search process. So, in a N_g dimensional search space, an i^{th} individual vector of population at G^{th} generation is denoted by $x_{ij}^G; (j = 1, 2, \dots, N_g, i = 1, 2, \dots, N_P)$.

For each x_{ij}^G the mutation operator generates a mutant vector v_{ij}^G as per following relation:

$$v_{ij}^G = x_{\chi_1 j}^G + F(x_{\chi_2 j}^G - x_{\chi_3 j}^G) \quad (j = 1, 2, \dots, N_g, i = 1, 2, \dots, N_P) \quad (6.1)$$

where F is the mutation factor; $\chi_1 \neq \chi_2 \neq \chi_3 \neq i \in [1, N_P]$ index are selected by applying either Gauss or Tent map sequence represented by Eq. (3.12) and (3.13). Further, to increase the diversity of mutant individual binomial crossover operation is used, resulting in trial vector. Mathematically, it is written as

$$u_{ij}^{G+1} = \begin{cases} v_{ij}^G & ; \chi_j \leq C_R \text{ or } j = \chi_i \\ x_{ij}^G & ; \text{otherwise} \end{cases} \quad (j = 1, 2, \dots, N_g, i = 1, 2, \dots, N_P) \quad (6.2)$$

where $\chi_j \in [0, 1]$ is the j^{th} evaluation of random number. $C_R \in [0, 1]$ is the crossover constant. $\chi_i \in [1, N_G]$ is a randomly chosen index by exploiting chaotic sequence.

In order to decide, whether a trial vector is selected as a member of $G + 1^{th}$ generation, the survival of the fittest principle is followed. Mathematically,

$$x_{ij}^{G+1} = \begin{cases} u_{ij}^{G+1} & ; u_{ij}^{G+1} < x_{ij}^G \\ x_{ij}^G & ; \text{otherwise} \end{cases} \quad (j = 1, 2, \dots, N_g, i = 1, 2, \dots, N_P) \quad (6.3)$$

This ensures that the individual vectors in the next generation must be better or at least of the same quality to that of the present generation

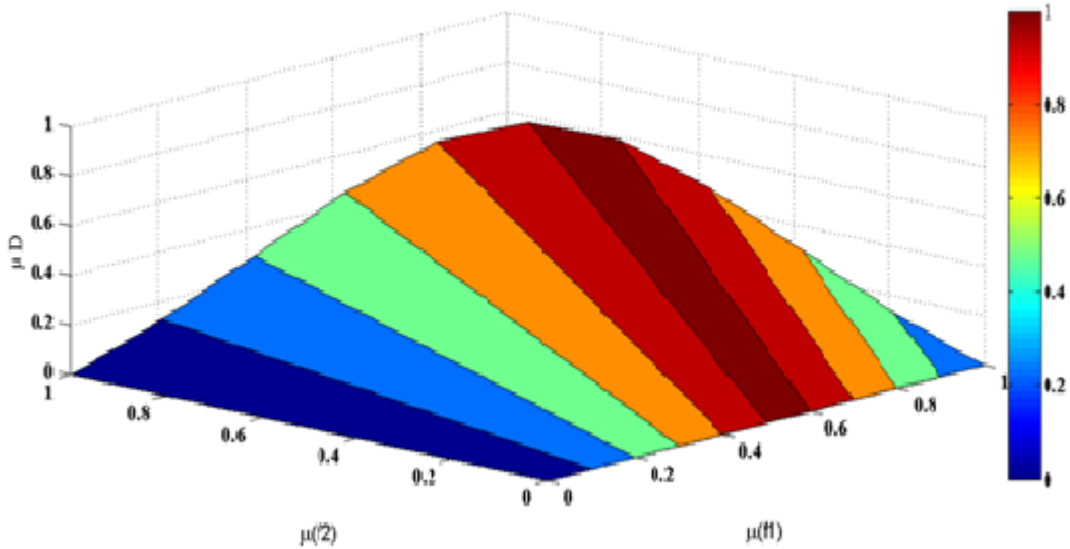


Figure 6.1: Scaling function behavior

6.3 SOLUTION METHODOLOGY

Multiobjective optimization problems can be solved by either by converting the problem involving multiple objectives into an optimization problem with a single objective function or by consideration of DM. The two approaches are classified as non-interactive and interactive approach respectively. In this chapter, the MOLD has been solved by consideration of the scaling function, which is applied, as no preference approach (non-interactive method) as well as an interactive method as discussed in following subsections:

6.3.1 Non-Interactive Approach

Multiobjective optimization problems can be solved by converting the problem involving multiple objectives into an optimization problem with a single objective function. As the new problem has a real-valued objective function that possibly depends on some parameters, it can be solved using appropriate single objective optimizers. This approach to solve a problem is beneficial, when either there is no DM or a justified Pareto front is not available. In such cases a no-preference approach is a better option. Then, the task is to find some neutral compromising solution without using any additional preference information. This means that instead of asking the DM for preference information, some assumptions are made about the nature of reasonable compromised between conflicting objectives. A

no-preference method to obtain compromise solution requires a scalar objective function. So, an inference function is proposed to club the two objective functions. The proposed scaling function (μ_S) is a quality index to indicate the level of satisfaction of both the objective functions to a particular solution. Considering the aggregation function μ_A (Eq. (3.22)) and satisficing function μ_c (Eq. (4.19)), the proposed scaling function to indicate the quality of compromised solution is mathematically written as follows:

$$\mu_S = \begin{cases} \mu_c \mu_A & ; \mu_A \leq 1 \\ \mu_c \mu_A^{-1} & ; \mu_A > 1 \end{cases} \quad (6.4)$$

The non-interactive approach to solve MOLD problem has two constituents: the CDEPS approach to generate the feasible solutions and a non-interactive scaling function to perform selection operation. The solution procedure is shown as a flowchart in Fig. 6.2 and is discussed in detail as follows:

6.3.1.1 Solution structure

In order to obtain the solution of MOLD problem by the proposed CDEPS approach, the set of output powers of generators is the decision variables. So, in a problem having N_g dedicated generating units, the decision variable P will be represented as $P = [P_1, P_2, \dots, P_{N_g}]$. Considering N_P as the population size, an i^{th} possible solution of problem at G^{th} iteration is represented as:

$$P_i^G = [P_{i1}^G, P_{i2}^G, \dots, P_{iN_g}^G]^T \quad (i = 1, 2, \dots, N_P) \quad (6.5)$$

6.3.1.2 Population initialization

In CDEPS based solution procedure the initial population is randomly generated within the feasible space. In the absence of any prior knowledge about solution and problem's hyperspace, chaotic sequence based population initialization process is adopted. Chaotic systems are nonlinear and dynamic in nature, and highly sensitive to initial conditions. Recently, these chaotic systems improve the performance of heuristic approaches. Given

the generator operational limits $[P_j^{min}, P_j^{max}]$ respectively as:

$$P_{ij} = P_j^{min} + \chi(P_j^{max} - P_j^{min}) \quad (j = 1, 2, \dots, N_g, i = 1, 2, \dots, N_P) \quad (6.6)$$

where $\chi \in [0, 1]$ is a chaotic sequence generated number.

6.3.1.3 Feasible solution update

The i^{th} individual in the population is modified under the effect of procedures, namely mutation, crossover, pattern search and selection as discussed in section 6.2. The diversification operators are Gauss map/Tent Map guided mutation and crossover whereas the selection operation is driven by SWT preferences, which reflects the satisfaction of a solution to the problem objectives. The PS improves solution quality and ensures deterministic exploitation in the search process. The PS is activated by a stochastic switch, operated by the threshold value IT^{HIT} and executes for IT^{PS} cycles. The overall best position among individual's best positions is memorized as the global best experience g^{best} . If the i^{th} individual violates the boundary conditions, it is fixed to the respective limit, whereas the equality constraint of MOLD problem is taken care by sharing violated power magnitude among other dedicated generators. In this paper, maximum number of iterations (IT^{MAX}) is taken as the stopping criterion. On satisfying the termination criteria, the global best solution g^{best} is treated as an optimal compromised solution of the problem.

6.3.2 Interactive Solution Approach

In interactive approaches, an iterative solution algorithm is used and repeated until a stopping criteria is satisfied. After each iteration, some information is given to the DM to specify preferences. The analyst aims to determine the preference structure of the DM in an interactive way. An important benefit is that the DM can specify and adjust one's preferences between each iteration and at the same time learn about the interdependencies in the problem as well as about one's own preferences. The interactive procedure to solve MOLD has been discussed in Section 3.4.2 and 3.4.3 respectively.

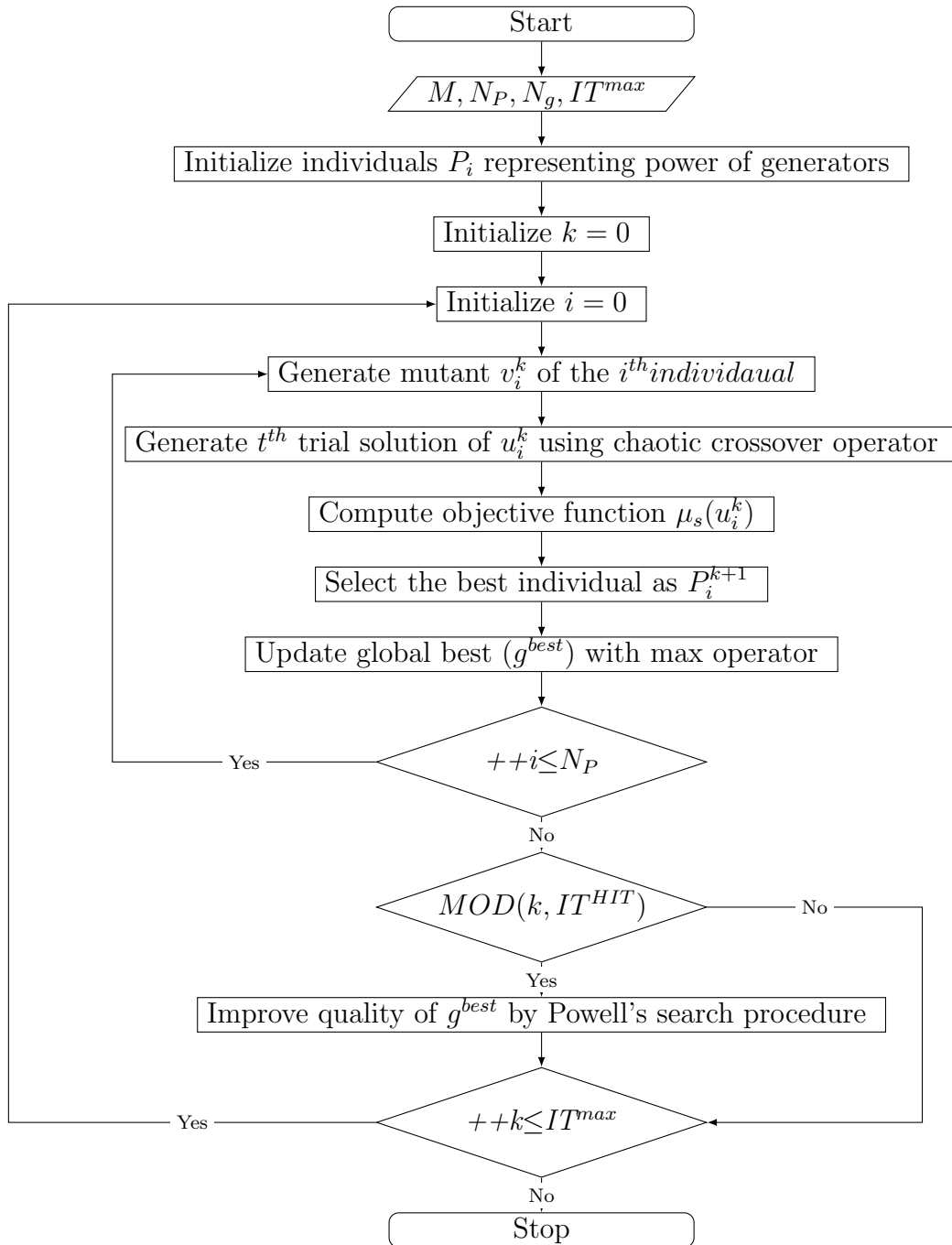


Figure 6.2: Non-interactive solution approach using CDEPS approach

The interactive approach used in this chapter to solve MOLD problem has three constituents: the CDEPS approach to generate the non-inferior solutions and an interactive method to assign preferences to Pareto-points. The solution procedure is shown as a flowchart in Fig. 6.3.

6.4 SIMULATION CASES AND RESULTS

To prove the capability of the CDEPS solution approach, the generalized benchmark test functions as well as standard real world problems related to power system operation are undertaken. The Gauss and Tent map sequences are applied to investigate their behavior while implementing CDE and CDEPS. On using Gauss map and Tent sequence the CDE method is termed as CDE-1 and CDE-2. A similar notation is used for CDEPS procedures. The generalized benchmark problems specified in Table 2.1 has been simulated and their results are presented. The performance investigation, aims to compare the convergence behavior of CDE and CDEPS. The small, medium and large capacity MOLD problems shown in Table 2.2 are undertaken to verify the practicability of SWT method. The search is performed by a CDEPS algorithm to generate the non-dominated solutions and SWT approach has been used as a non interactive and interactive approaches respectively.

6.4.1 Generalized Test Functions

In order to compare the performance of CDEPS variants with CDE, seven benchmark test functions listed in Table 2.1 are solved using CDEPS and CDE variants. The best, worst and mean experimental results after 2000 iterations for each test function performing 30 trial runs are also listed in Table 6.1. The chaotic process is used to adjust the crossover rate (C_R) and mutation factor (F). The results tabulated in Table 6.1 shows that among 30 trials, for Griewank's function the best solution is obtained by CDEPS-1, whereas the CDE-1 and CDEPS-2 has resulted in worst outcome. The average solution quality has been found to be same for CDE-1, CDEPS-1 and CDEPS-2. In case of Rastrigin's function, the best quality solution is obtained by CDE-1, followed by CDEPS-2. In case of Rosenbrock's function, both CDEPS variants have achieved same solution quality, followed by CDE variants. In case of Schwefel's function, the CDEPS-1 algorithm achieves

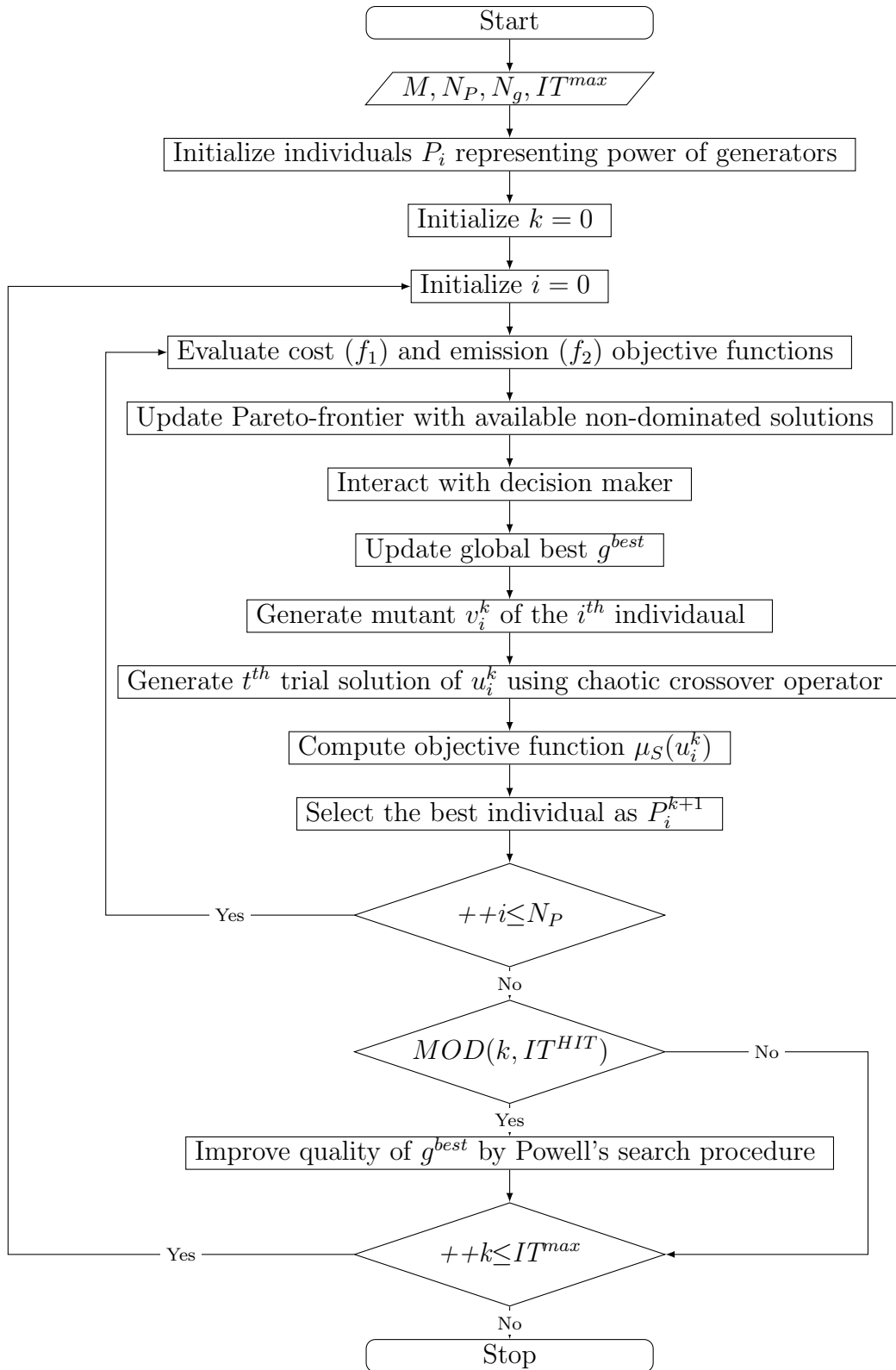
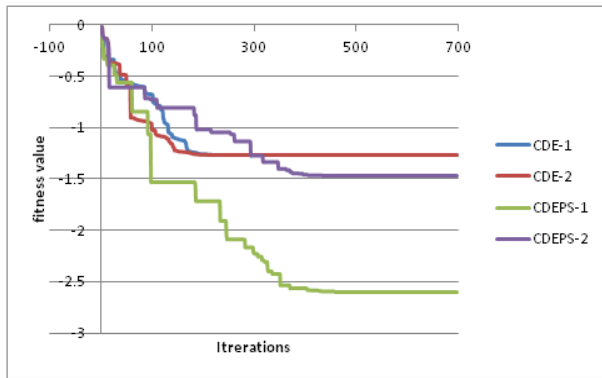
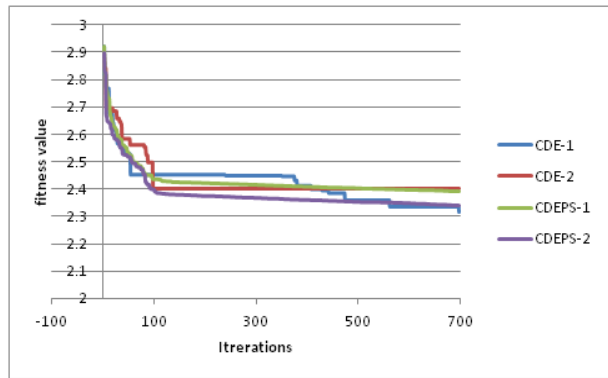


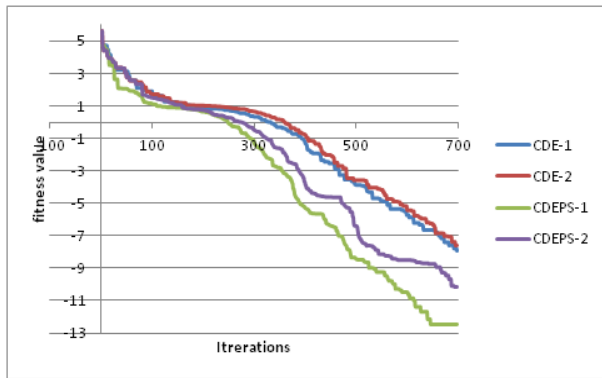
Figure 6.3: Interactive solution approach using CDEPS approach



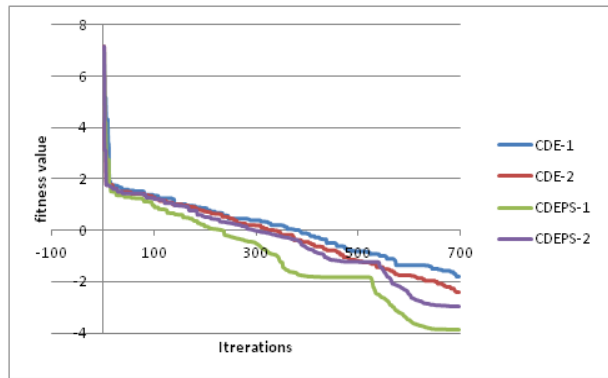
(a) Griewank function



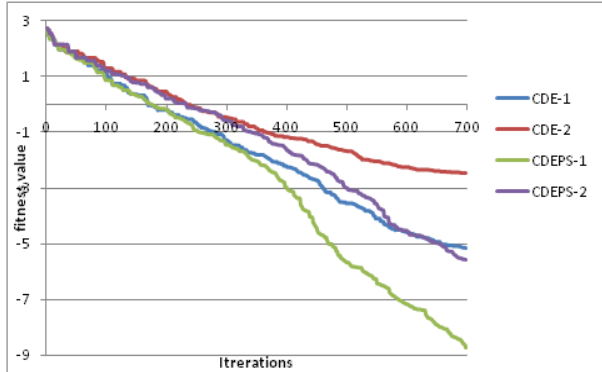
(b) Rastrigin function



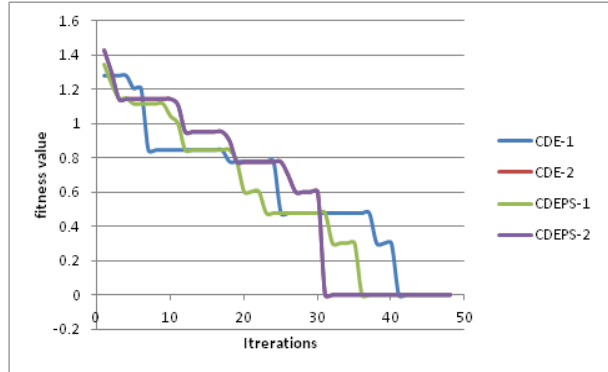
(c) Rosenbrock function



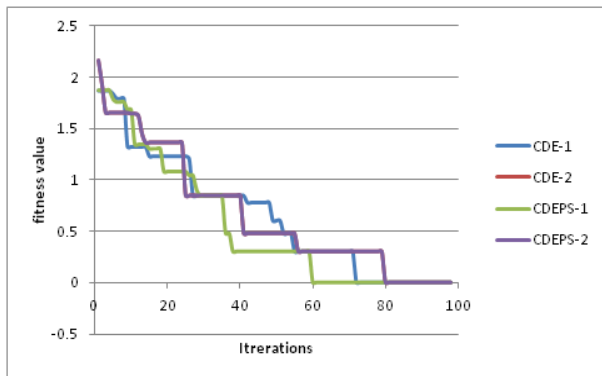
(d) Schwefel's 2.22 function



(e) Sphere function



(f) Step-1 function



(g) Step-2 function

Figure 6.4: Generalized benchmark test functions convergence comparison

Table 6.1: Performance analysis of generalized benchmark test functions

Test function	Fitness	Algorithm			
		CDE-1	CDE-2	CDEPS-1	CDEPS-2
Griewank function [380]	Worst	7.44×10^{-02}	5.14×10^{-02}	4.4×10^{-02}	7.44×10^{-02}
	Average	2.7×10^{-02}	4.17×10^{-02}	2.47×10^{-15}	2.47×10^{-02}
	Best	2.47×10^{-03}	3.44×10^{-02}	3.97×10^{-20}	5.42×10^{-02}
Rastrigin function [380]	Worst	166.2034	229.9046	206.5738	174.2419
	Average	166.2034	229.9046	206.5738	174.2419
	Best	166.2034	229.9046	206.5738	174.2419
Rosenbrock function [380]	Worst	1.55×10^{-10}	3.85×10^{-11}	0	0
	Average	7.55×10^{-11}	6.55×10^{-11}	0	0
	Best	3.55×10^{-13}	1.12×10^{-12}	0	0
Schwefel's 2.22 function [380]	Worst	1.52×10^{-09}	5.29×10^{-12}	2.41×10^{-16}	2.73×10^{-12}
	Average	1.52×10^{-09}	5.29×10^{-12}	2.41×10^{-16}	2.73×10^{-12}
	Best	1.52×10^{-09}	5.29×10^{-12}	2.41×10^{-16}	2.73×10^{-12}
Sphere function [380]	Worst	1.81×10^{-06}	6.75×10^{-04}	7.67×10^{-19}	3.97×10^{-20}
	Average	1.81×10^{-06}	6.75×10^{-04}	7.67×10^{-19}	3.97×10^{-20}
	Best	1.81×10^{-06}	6.75×10^{-04}	7.67×10^{-19}	3.97×10^{-20}
Step function [380]	Worst	0	0	0	0
	Average	0	0	0	0
	Best	0	0	0	0
Step-2 function [380]	Worst	0	0	0	0
	Average	0	0	0	0
	Best	0	0	0	0

a better quality solution, followed by CDEPS-2 and CDE variants. The experimentation of Sphere function shows that algorithm's performance follows a descending hierarchy as CDEPS-2, CDEPS-1, CDE-1 and CDE-2. In case of Step and Step-2 benchmark functions, the solution quality remains same for all the contenders. Further, in comparison to the synergic predator prey optimization algorithm (SPPO) results given in Table 2.4, chaotic evolutionary programming and pattern search in Table 3.1 and fy and walk predator prey optimization (FWPO) the CDEPS performance is better.

The convergence comparison plots for considered benchmark test functions are shown in Figure 6.4 on a logarithmic scale. Figure 6.4a presents the convergence behavior of the Griewank's function, where the CDEPS-1 algorithm is better in terms of solution quality and convergence rate. Figure 6.4b, depicts the behavior of the Rastrigin's function, where the CDE-1 variant is able to obtain the best solution for this function, whereas the CDEPS-2 is observed as a fastest performer. The appraisal of Figure 6.4c shows that in case of the Rosenbrock's function the CDEPS-1 has better convergence characteristics. In case of the Schwefel's function, CDEPS-1 exhibits best convergence behavior followed

by the CDEPS-2, and is shown in Figure 6.4d. Further, the experimental result of the Sphere function in Figure 6.4e, shows that the CDEPS-1 algorithm is better in terms of convergence rate. The convergence plot in Figure 6.4f and Figure 6.4g for step and step-1 function, presents CDEPS-2 as a better performing variant. Hence, the convergence plots clearly show that in comparison to CDE variants, the CDEPS variants perform better. The better performance is either in terms of solution quality or/ convergence rate. Moreover, it is observed that CDEPS-1 performs better on some of the benchmark, whereas CDEPS-2 shows superiority for other considered benchmark functions.

6.4.2 Test Power System-1

This system's power generators have the option of multiple-fuels. The system has been solved for economic load dispatch (ELD), minimum emission dispatch (MED) and multiobjective load dispatch (MOLD). In all the undertaken six cases, shown in Table 2.2, the value of population size (N_P) has been taken as 20 and the maximum number of iterations (IT^{MAX}) is set to 100. The cost comparison has been performed for ELD

Table 6.2: Test power system-1, comparison of cost ($P_D = 2700MW$)

Algorithm	cost(\$/h)					
	Case-1	Case-2	Case-3	Case-4	Case-5	Case-6
BBO [306]	624.51	–	–			
CCPSO [321]	–	623.82	–	–	–	–
CGA-MU [373]	623.80	624.71	–	–	–	–
DE [373]	623.80	624.46	–	–	–	–
DEBBO [305]	624.51	–	–			
ELHN [369]	624.51	–	–			
IGA [163]	624.51	–	–			
IGA-MU [163]	623.80	624.51	–			
KHA [397]	624.51	–	–			
PSO[373]	623.80	624.24	–			
QP-ALHN [376]	623.80	–	624.32			
SPPO	623.80	623.82	624.321	700.29	700.77	700.48
CEPPS-1	623.75	623.87	623.76	699.70	699.54	704.94
CEPPS-2	623.75	623.88	623.77	699.77	699.73	700.60
APPO	623.80	623.82	624.32	699.3	699.4	700.41
FWPO	623.75	623.85	624.32	699.33	699.40	700.48
CDEPS-1	623.75	623.82	623.75	699.26	699.36	699.26
CDEPS-2	623.75	623.80	623.75	699.26	699.36	699.35

with other published work such as the results of *Biogeography based optimization* (BBO)

[306], *Composite Particle Swarm Optimization* (CCPSO) [386], *Conventional Genetic Algorithm with Mutation Update* (CGA-MU) [163], *Differential Evolution* (DE)[373], *Improved Genetic Algorithm* (IGA) [163], *Improved Genetic Algorithm with multiplier update* (IGA-MU) [163], *Improved Particle Swarm Optimization* (IPSO) [373], *Hybrid of DE and BBO* (DEBBO) [386], *Krill herd optimization* (KHA) [397], *Quadratic Programming Augmented Hopfield Neural Network* (QP-ALHN) [376], *Particle Swarm Optimization* (PSO) [373], *Synergic Predator Prey Optimization* (SPPO), *Chaotic Evolutionary Programming Pattern Search* (CEPPS), *Adaptive Predator Prey Optimization* (APPO) and *Fly and Walk Predator Prey Optimization* (FWPO). The comparison of power generation cost is presented in Table 6.2. The comparison indicates that CDEPS-2 is able to obtain a generation schedule with operating cost either better or equal to its counterparts.

Table 6.3: Test power system-1, Cost and emission for ELD and MED respectively

Case	ELD		MED	
	Min. Cost(\$/h)	Max. Emission(lb/h)	Min. Emission(lb/h)	Max. Cost (\$/h)
1	623.76	12092.54	103.9	694.22
2	623.80	12071.62	103.9	695.17
3	623.75	12184.70	103.9	696.25
4	699.26	36678.18	110.74	770.0
5	699.36	38990.66	110.74	771.42
6	699.35	31427.20	110.74	770.78

To obtain a MOLD solution, the proposed scaling function (Eq. (6.4)) requires objective functions extrema. The best achieved minimum and maximum values of the conflicting objectives for all six cases, while implementing CDEPS variants are depicted in Table 6.3. These values are obtained from ELD and MED formulation's results and exploring the conflicting nature of objective functions.

The MOLD's compromise solution obtained using CDEPS approach have been compared with CEPPS, APPO and FWPO. The membership values are obtained by using the proposed satisficing function (Eq. (6.4)) and objective's extrema tabulated in Table 6.3. The comparison of the results shows that in case-1 the CDEPS-2 is able to obtain better solution, as the satisficing function value μ_S is better than CDEPS-1 and others except FWPO, where the solution is comparable. In case-2, both CDEPS-1 and CDEPS-2 works equally well as evident from the value of μ_S . In this case, the solution quality is

Table 6.4: Test power system-1, MOLD solution comparison ($P_D = 2700MW$)

Algorithm	Case-1			Case-2			Case-3		
	Cost(\$/h)	Emission(lb/h)	μ_S	Cost(\$/h)	Emission(lb/h)	μ_S	Cost(\$/h)	Emission(lb/h)	μ_S
CEPPS-1	629.49	1185.10	0.909	629.97	1263.65	0.903	629.62	1208.05	0.908
CEPPS-2	629.52	1192.03	0.909	630.15	1277.18	0.901	628.07	663.66	0.940
APPO	628.85	464.71	0.926	629.34	485.18	0.921	629.66	454.42	0.917
FWPO	628.11	657.19	0.938	628.18	663.57	0.938	628.54	671.42	0.933
CDEPS-1	629.35	1158.52	0.911	629.41	1168.05	0.911	629.74	1103.13	0.917
CDEPS-2	628.06	664.11	0.938	629.38	1163.80	0.911	628.07	663.07	0.938

Table 6.5: Test power system-1, MOLD solution comparison ($P_D = 2700MW$)

Algorithm	Case-4			Case-5			Case-6		
	Cost(\$/h)	Emission(lb/h)	μ_S	Cost(\$/h)	Emission(lb/h)	μ_S	Cost(\$/h)	Emission(lb/h)	μ_S
CEPPS-1	701.31	1049.95	0.971	701.33	981.0	0.972	705.11	879.83	0.917
CEPPS-2	705.10	1129.55	0.915	704.97	972.46	0.920	704.97	3086.99	0.904
APPO	704.82	810.65	0.919	705.32	777.53	0.915	704.42	615.67	0.927
FWPO	702.24	2606.09	0.931	702.37	2616.66	0.935	703.83	3585.38	0.887
CDEPS-1	701.07	953.35	0.974	701.22	931.65	0.974	701.60	904.64	0.974
CDEPS-2	701.16	904.64	0.973	701.1	1045.14	0.975	701.16	904.65	0.974

slightly degraded than FWPO. In case-3, CDEPS-2 gives a solution which is comparable with CEPPS-2 and better than all others. In case 4-6, both CDEPS-1 and 2 have resulted in a better compromised solution than its counterparts. In case-6, both CDEPS-1 and CDEPS-2 have provided the same quality of solution. Ultimately, in all six cases, the quality of solution obtained by CDEPS-2 is either superior or at par with the solution of CDEPS-1.

6.4.3 Test Power System-2

This Test power system consists of 40 generators and firstly it is solved for ELD. In all the undertaken dispatch problems, the value of population size N_P is taken as 50 and IT^{MAX} is set to 1000. The objective function values of the obtained optimal power allocation for ELD formulation are presented in Table 6.7. The results are compared with the results of *Differential Evolution* (DE) [392], *Gravitational Search Algorithm* (GSA) [358], *Improved Differential Evolution* (IDE) [392], *Kinetic Gas Molecule Optimizer* (KGMO) [430], *Modulated Artificial Bee Colony* (MABS) [424], *Multiobjective Differential Evolution* (MODE) [330], *Modulated Particle Swarm Optimization* (LMPSO) [417], *Non-dominated sorting Genetic Algorithm* (NSGA) [330], *Pareto Differential Evolution* (PDE) [330], *Quasi Teacher Learner Based Optimization* (QTLBO) [384], *Real Coded Chemical Reaction Optimization* (RCCRO), *Sinusoidal Modulated Particle Swarm Optimization* (SMPSO) [417], *Strength Pareto Evolutionary algorithm* (SPEA) [330], *Teacher*

Table 6.6: Test power system-2, Optimal solution comparison ($P_D=10500$ MW)

Algorithm	ELD		MED		MOLD		μ_S
	Cost $\times 10^5$ \$/h	Emission $\times 10^5$ lb/h	Cost $\times 10^5$ \$/h	Emission $\times 10^5$ lb/h	Cost $\times 10^5$ \$/h	Emission $\times 10^5$ lb/h	
DE[392]	1.21442	–	–	–	–	–	–
GSA [358]	–	–	–	–	1.25782	2.10932	0.425
IDE [392]	1.21370	–	–	–	–	–	–
KGMO [430]	1.21378	–	–	–	–	–	–
LMPSO [417]	–	–	–	–	1.24668	2.19048	0.598
MABS I [424]	1.21412	3.59901	1.29995	1.766820	1.24490	2.5656	0.551
MABS II [424]	1.21412	3.59901	1.29996	1.766820	1.24491	2.56551	0.551
MODE [330]	1.2184	3.7479	1.2996	1.7668	1.25792	2.1119	0.424
MPSO [411]	1.21379	–	–	–	–	–	–
NSGA II [330]	–	–	–	–	1.25825	2.10949	0.418
PDE [330]	–	–	–	–	1.25730	2.1177	0.434
SEPA II [330]	–	–	–	–	1.25807	2.1109	0.421
QTLBO [384]	1.21428	3.53417	1.29955	1.766825	1.25161	2.06648	0.513
SMPSO [417]	–	–	–	–	1.24660	2.17256	0.598
TLBO [384]	1.21685	3.64593	1.29953	1.766835	1.26020	2.06648	0.383
PPO	1.21401	3.60420	1.29979	1.766843	1.25750	1.99125	0.414
SPPO	1.21402	3.56724	–	–	–	–	–
CEPPS-1	1.27180	3.74117	1.34516	2.46508	1.28124	3.14471	0.207
CEPPS-2	1.21487	3.63871	1.32889	2.25013	1.28515	3.27227	0.163
APPO	1.21402	3.56724	1.29924	1.76682	1.25667	1.95515	0.421
FWPO	1.21374	3.60018	1.29924	1.76682	1.24666	2.47257	0.614
CDEPS-1	1.21368	3.59873	1.299252	1.766817	1.24341	2.4061	0.651
CDEPS-2	1.21368	3.59873	1.299209	1.766814	1.24313	2.40042	0.654

Learner Based Optimization (TLBO) [384], *Predator Prey Optimization* (PPO) and *Synergic Predator Prey Optimization* (SPPO), *Chaotic Evolutionary Programming Pattern Search* (CEPPS) and *Fly and Walk Predator Prey Optimization* (FWPO). The comparison of ELD results reveals that the results obtained by implementing CDEPS-1 and CDEPS-2 are better than the results available in the literature. In addition, the equality of results in terms of operating cost are competitive while implementing CDEPS-1 and

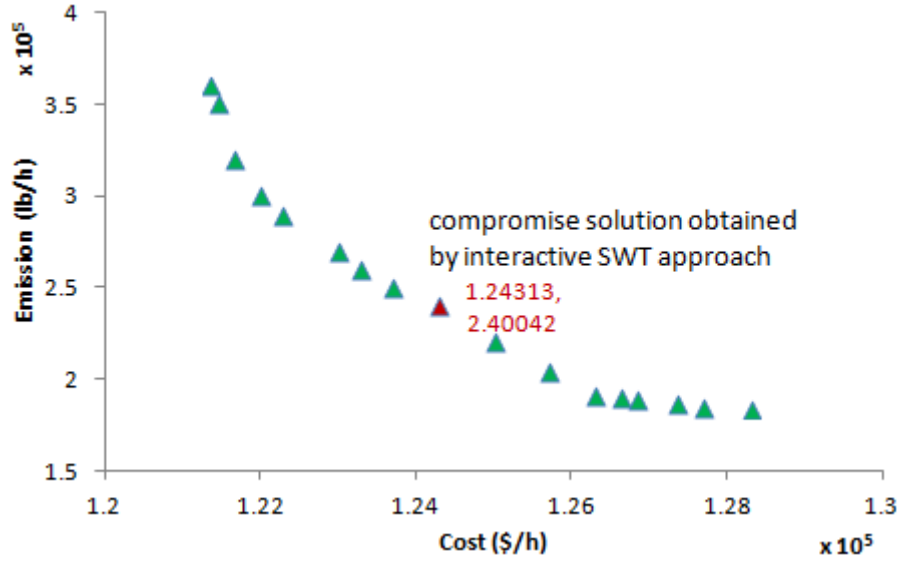


Figure 6.5: Test power system-2 Pareto-front using interactive SWT approach

Table 6.7: Test power system-2, Comparison of MOLD solution using interactive and non-interactive approach ($P_D = 10500MW$)

Algorithm	Interactive approach				Non-interactive approach			
	Cost $\times 10^5$ (\$/h)	Emission $\times 10^5$ (lb/h)	μ_D	μ_S	Cost $\times 10^5$ (\$/h)	Emission $\times 10^5$ (lb/h)	μ_D	μ_S
CDEPS-1	1.25677	1.95915	0.849	0.422	1.24341	2.40610	0.999	0.651
CDEPS-2	1.24313	2.40042	0.999	0.654	1.24313	2.40042	0.999	0.654

CDEPS-2. Further, in MED formulation the CDEPS-1 is able to generate better solution than CDEPS-2. The comparison of MOLD solution obtained by non-interactive SWT approach is presented in Table 6.6. The compromised solution obtained by CDEPS-1 and CDEPS-2 is better than other existing results in the literature, with satisficing function values, μ_S of 0.651 and 0.654, respectively. In addition, the CDEPS-2 has obtained a better quality solution as compared to CDEPS-1. The Table 6.8 presents the CDEPS-2's generated optimal generation schedule. The MOLD solution of test power system-2 is also obtained by interactive SWT approach. The Pareto front obtained by applying the

Table 6.8: Test system-3, MOLD power schedule by APPO ($P_D = 10500MW$)

Unit	Power (MW)	Unit	Power (MW)	Unit	Power (MW)	Unit	Power (MW)
P_1	113.43	P_{11}	243.51	P_{21}	491.44	P_{31}	164.94
P_2	113.43	P_{12}	174.19	P_{22}	522.17	P_{32}	189.35
P_3	120.0	P_{13}	394.27	P_{23}	518.09	P_{33}	175.11
P_4	179.69	P_{14}	394.35	P_{24}	433.78	P_{34}	200.0
P_5	96.82	P_{15}	394.28	P_{25}	433.53	P_{35}	200.0
P_6	139.98	P_{16}	394.32	P_{26}	433.52	P_{36}	199.99
P_7	299.33	P_{17}	488.52	P_{27}	9.99	P_{37}	109.99
P_8	284.59	P_{18}	489.20	P_{28}	9.99	P_{38}	105.09
P_9	284.59	P_{19}	421.51	P_{29}	9.99	P_{39}	109.90
P_{10}	130.0	P_{20}	509.30	P_{30}	96.2	P_{40}	421.49

CDEPS-2 approach is shown in Fig. 6.5, where the compromised solution obtained is highlighted. In comparison to the Pareto front of FWPO (Fig. 5.6), it can be clearly observed that FWPO has obtained more number of points on Pareto front, but CDEPS pareto front is more uniformly distributed.

The obtained compromised solution using interactive SWT is tabulated in Table 6.7, which is of the same quality as that of obtained by applying the non-interactive approach. In addition, in Table 6.7 the quality of compromised solution is also indicated by surrogate worth trade-off index μ_D , from where it is clearly indicated that μ_S is a credible index to represent the quality of compromised solution.

6.4.4 Test Power System-3

This is a Korean power system consisting of 140 power generators as shown in Table 2.1. This system is investigated in two cases. The case-1 considers the quadratic objective function with POZ constraints and ramp rate limits, whereas the case-2 considers valve-point loading effect on some of the generators Table 2.1. In order to obtain the solution, the value of population size N_P is taken as 200 and maximum iterations IT^{MAX} is set to 5000. Firstly, the problem is solved for ELD. The cost comparison of ELD results by CDEPS-1 and CDEPS-2, for case-1 and case-2 of Korean system is tabulated in Table 6.9. The results of CDEPS algorithm are compared with *Composite Particle Swarm Optimization* (CCPSO) [321], *Group Search Optimizer* (GSO) [360], *Continuous Quick Group Search Optimizer* (CQGSO) [360], *Differential Harmony Search* (DHS) [388], *Anti*

Predator Swarm Optimization (APSO), *Predator Prey Optimization* (PPO), *Chaotic Evolutionary Programming Pattern Search* (CEPPS), *Adaptive Predator Prey Optimization* (APPO) and *Fly and Walk Predator Prey Optimization* (FWPO). The cost comparison shows that in case-1, both CDEPS-1 and CDEPS-2 variants have obtained a better quality

Table 6.9: Test power system 3, ELD solution comparison ($P_D = 49342MW$)

Algorithm	Power generation cost (\$/h)	
	Case-1	Case-2
CCPSO [321]	1655685.0	1657962.75
CQGSO [360]	1655679.42	1657962.74
DHE [388]	–	1657944.86
GSO [360]	1734405.7	–
PPO	1655684	1657963.5
APSO	1655684.5	1657963.0
SPPO	1655680.0	1657962.0
CEPPS-1	1778880.0	1795760.0
CEPPS-2	1778471.0	1784659.0
APPO	1655683.0	–
FWPO	1655674.0	–
CDEPS-1	1655674.0	1657117.0
CDEPS-2	1655675.0	1659291.0

solution as compared to its counterparts. Further, in comparison to CDEPS-2, CDEPS-1 has obtained a solution with lesser power generation cost. However, for Case-2 of Test system-3, the CDEPS-2 was able to perform better than CEPPS variants. On the other hand, CDEPS-1 has resulted in lesser power generation cost than all other algorithms compared for comparison. Further, the Table 6.10 gives the operation cost corresponding to minimum emission achieved by solving MED problem. The table clearly shows that CDEPS-2 obtains better quality of solutions in terms of lesser emission of pollutants, as compared to the CDEPS-1.

The comparison of best compromised solution obtained by non-interactive SWT approach is performed with others on the basis of scaling function index, μ_s . The best compromised solution achieved by applying CDEPS-1 and CDEPS-2 is listed in Table 6.11. From the Table 6.11 it is observed, that CDEPS-2 gives better μ_s value as compared to CDEPS-1. The corresponding MOLD power schedule is presented in Table 6.12. Further, the test system-3, case-1 is solved for MOLD using an interactive approach. The Corresponding Pareto front is shown in Fig. 6.6, which shows the obtained compromise

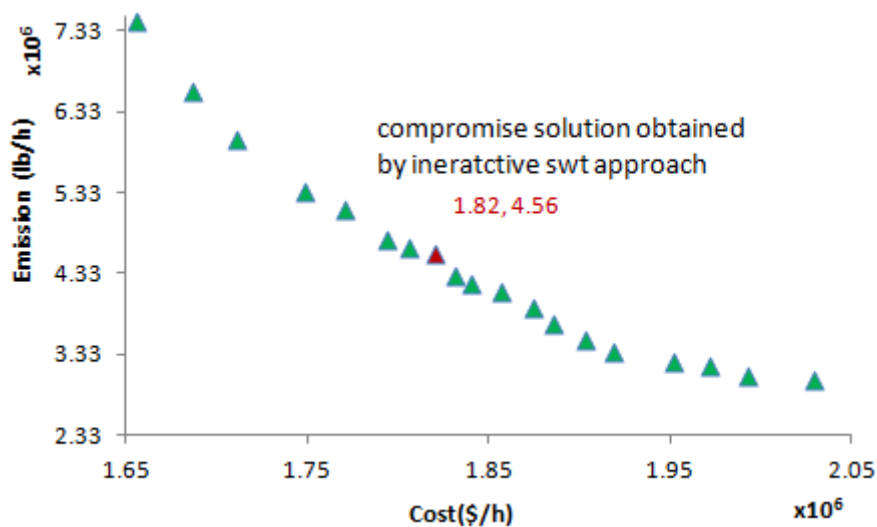
Table 6.10: Test power system 3, Extremum of cost and emission objective functions

Algorithm	Case-1				Case-2			
	ELD		MED		ELD		MED	
	Cost (\$/h)	Emission (lb/h)	Cost (\$/h)	Emission (lb/h)	Cost (\$/h)	Emission (lb/h)	Cost (\$/h)	Emission (lb/h)
CDEPS-1	1655674	7445358	2029025	3010234	1657117	7450891	2034347	3010234
CDEPS-2	1655675	7447630	2031090	3003420	1659291	7452072	2037315	3003420

Table 6.11: Test power system 3, MOLD solution comparison ($P_D = 49342MW$)

Algorithm	Case-1			Case-2		
	Cost(\$/h)	Emission(lb/h)	μ_S	Cost(\$/h)	Emission(lb/h)	μ_S
CEPPS-1	1831802	5902779	0.318	1832679	5932484	0.449
CEPPS-2	1815388	5695509	0.367	1821931	5790238	0.450
APPO	1717937	4919461	0.529	–	–	–
FWPO	1704270	5160329	0.450	–	–	–
CDEPS-1	1804164	4775084	0.602	1806903	4773480	0.601
CDEPS-2	1802892	4759965	0.605	1806699	4753203	0.606

solution. In comparison to the Pareto front of FWPO (Fig. 5.8), it can be clearly observed that the CDEPS Pareto front is more uniformly distributed.

**Figure 6.6:** Test power system-3, Pareto-front using interactive SWT approach

6.4.5 Convergence Behavior Analysis

To observe the convergence rate of CDEPS variants, a study is performed using test system-3. The experimental outcomes in both the cases 1 and 2 are shown in Fig. 6.7. The comparison of behavior clearly indicates that in case-1, the CDEPS-2 is able to achieve the compromised solution much ahead of its counterpart. Further, for case-2, which involves

Table 6.12: Test system 3, Optimal generation schedule for MOLD ($P_D=49342\text{MW}$)

Unit	Case-1	Case-2	Unit	Case-1	Case-2	Unit	Case-1	Case-2	Unit	Case-1	Case-2
P_1	118.99	117.38	P_{36}	499.99	150.51	P_{71}	452.77	379.04	P_{106}	880.60	880.89
P_2	123.32	139.77	P_{37}	231.58	451.15	P_{72}	448.27	202.25	P_{107}	812.25	805.73
P_3	167.38	161.82	P_{38}	224.47	537.75	P_{73}	539.94	477.29	P_{108}	804.50	800.58
P_4	183.92	181.14	P_{39}	437.39	178.36	P_{74}	524.69	535.99	P_{109}	799.15	796.92
P_5	128.21	180.84	P_{40}	763.63	178.39	P_{75}	539.99	476.65	P_{110}	804.63	825.23
P_6	183.39	185.76	P_{41}	8.46	495.91	P_{76}	518.81	534.24	P_{111}	881.99	881.99
P_7	489.37	473.43	P_{42}	23.48	174.86	P_{77}	178.32	183.39	P_{112}	94.72	84.97
P_8	489.74	486.90	P_{43}	249.11	568.13	P_{78}	329.63	328.61	P_{113}	175.18	98.20
P_9	495.99	495.42	P_{44}	159.91	530.99	P_{79}	162.23	215.73	P_{114}	100.47	106.53
P_{10}	259.97	265.27	P_{45}	243.79	530.99	P_{80}	528.89	520.09	P_{115}	378.83	272.97
P_{11}	495.76	493.73	P_{46}	159.93	541.75	P_{81}	219.47	244.33	P_{116}	264.38	378.40
P_{12}	494.67	467.11	P_{47}	245.39	55.82	P_{82}	63.06	74.32	P_{117}	242.93	312.24
P_{13}	502.52	498.34	P_{48}	230.22	115.89	P_{83}	115.85	224.09	P_{118}	132.71	134.64
P_{14}	507.98	397.82	P_{49}	249.04	114.93	P_{84}	159.79	178.43	P_{119}	97.54	95.51
P_{15}	502.13	505.99	P_{50}	159.93	114.85	P_{85}	133.03	208.71	P_{120}	124.37	127.33
P_{16}	504.30	408.87	P_{51}	495.70	206.76	P_{86}	206.78	213.39	P_{121}	174.87	248.60
P_{17}	484.74	489.37	P_{52}	503.85	207.09	P_{87}	212.86	306.62	P_{122}	7.74	3.55
P_{18}	503.45	505.99	P_{53}	329.62	174.79	P_{88}	274.59	258.93	P_{123}	19.79	10.85
P_{19}	504.58	505.0	P_{54}	169.46	175.50	P_{89}	336.31	344.99	P_{124}	53.97	26.48
P_{20}	505.0	495.69	P_{55}	180.40	183.70	P_{90}	174.78	176.42	P_{125}	52.37	10.29
P_{21}	505.0	504.99	P_{56}	181.79	175.15	P_{91}	286.93	292.96	P_{126}	11.92	18.68
P_{22}	505.0	505.0	P_{57}	326.08	575.4	P_{92}	575.39	575.40	P_{127}	12.97	27.77
P_{23}	504.99	502.39	P_{58}	197.98	547.5	P_{93}	547.49	547.49	P_{128}	120.0	247.50
P_{24}	505.0	505.0	P_{59}	295.89	836.8	P_{94}	836.39	836.79	P_{129}	19.99	4.81
P_{25}	534.92	532.25	P_{60}	392.19	837.5	P_{95}	837.49	837.49	P_{130}	6.10	14.74
P_{26}	536.99	530.61	P_{61}	302.71	682.0	P_{96}	681.32	681.99	P_{131}	11.99	5.31
P_{27}	279.49	277.6	P_{62}	156.85	720.0	P_{97}	719.99	719.99	P_{132}	52.0	57.24
P_{28}	279.54	277.65	P_{63}	446.79	718.0	P_{98}	717.99	711.79	P_{133}	8.13	9.58
P_{29}	300.43	297.92	P_{64}	328.97	720.0	P_{99}	719.99	719.99	P_{134}	64.27	61.63
P_{30}	356.88	497.88	P_{65}	485.48	964.0	P_{100}	962.17	963.99	P_{135}	48.17	62.30
P_{31}	505.99	505.10	P_{66}	481.75	958.0	P_{101}	939.83	957.96	P_{136}	47.12	48.64
P_{32}	505.97	505.99	P_{67}	195.46	947.90	P_{102}	946.43	911.61	P_{137}	21.09	16.94
P_{33}	503.45	479.11	P_{68}	195.49	934.0	P_{103}	930.41	933.99	P_{138}	7.39	8.0
P_{34}	503.12	505.99	P_{69}	129.21	935.0	P_{104}	934.99	935.0	P_{139}	18.71	11.46
P_{35}	499.99	499.99	P_{70}	171.48	876.50	P_{105}	865.79	876.49	P_{140}	39.92	36.20

complexity of valve-point loading effects the CDEPS-2 has performed much better.

6.4.6 Robustness Analysis

A statistical analysis is performed with 30 independent trials to compare the performances of CDEPS-1 and CDEPS-2. The Test power system-3 has been considered for the robustness analysis. Thirty independent trials are performed in both cases 1 and 2, for MOLD of the system using CDEPS-1 and CDEPS-2. The outcomes are shown as whisker plot in Fig. 6.8. The case-1 comparison clearly indicates that both CDEPS variants have nearly the same maximum value and quartiles, whereas the worst outcome in CDEPS-2 is

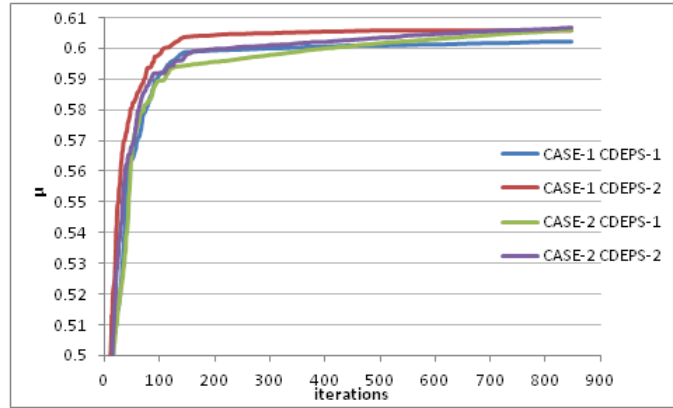


Figure 6.7: Convergence behavior, Test power system-3, case-1 and 2

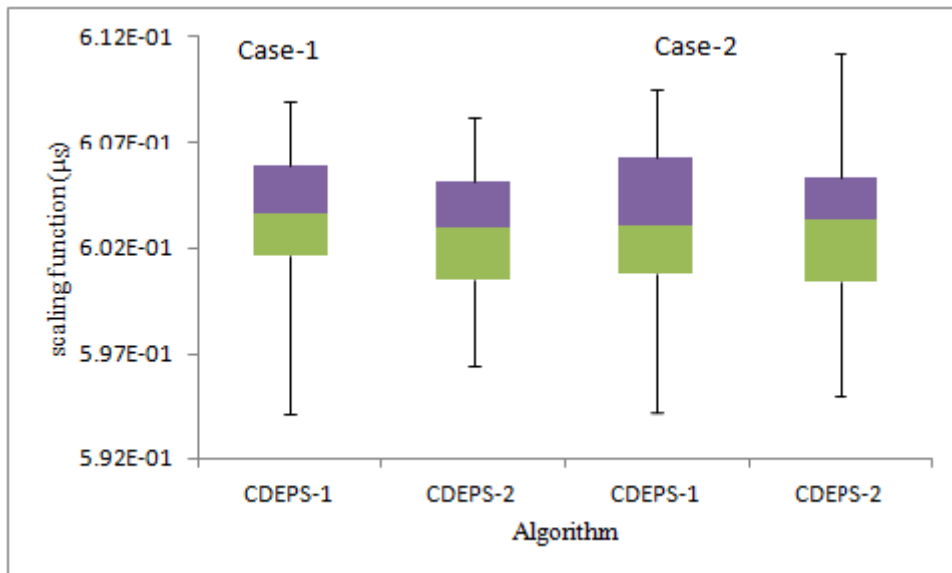


Figure 6.8: Robustness analysis test, Test power system-3, case-1 and 2

better in quality than the worst solution of CDEPS-1. Further, the comparison of case-2 outcomes clearly indicate that CDEPS-2 performs better in all aspects than CDEPS-1. The corresponding data statistics are tabulated in Table 6.13, from where it is evident that CDEPS-2 and CDEPS-1 respectively, have better robustness as compared to DE.

Table 6.13: Test power system-3, Descriptive statistics

	CASE-1			CASE-2		
	CDEPS-1	CDEPS-2	DE	CDEPS-1	CDEPS-2	DE
Mean (μ_s)	0.603215	0.60267	0.586978	0.602972	0.602757	0.584225
Min. (μ_s)	0.594165	0.596447	0.573614	0.594305	0.595061	0.540662
Max. (μ_s)	0.608933	0.608238	0.594608	0.609503	0.611225	0.595061
S.D.	0.003691	0.003101	0.004831	0.003899	0.003782	0.009821

6.4.7 Wilcoxon's Signed Rank Test

The Wilcoxon signed rank test has been applied to verify the effectiveness of CDEPS technique to obtain solution in comparison to its parent algorithm differential evolution, using Test power system-3. The test is performed to verify the significant difference at the 5 % level of confidence. The outcomes of the test are tabulated in Table 6.14. The test statistics clearly indicate that in both cases the p-value is lesser than the considered significance level, hence there is a significant difference between two samples. Hence, it is concluded that CDEPS techniques are able to achieve high quality results at every trial.

Table 6.14: Test power system 3, Wilcoxon's signed rank test outcomes

	Case-1		Case-2	
	CDEPS-1	CDEPS-2	CDEPS-1	CDEPS-2
p-value	3.34×10^{-11}	3.02×10^{-11}	4.08×10^{-11}	3.34×10^{-11}
z-score	6.63	6.645	6.6	6.63

6.5 CONCLUSION

In this research work, chaotic differential evolution (CDE) and Powell's patten search (PS) has been successfully hybridized for power scheduling of multiobjective thermal power load dispatch (MOLD) problem. The surrogate worth trade-off approach has been modified and a scaling function is proposed to solve multiobjective problem. The proposed

function has been investigated in two modes *viz.* Non-interactive and Interactive. The conflicting objectives being vague, the proposed scaling function implicates fuzzy theory. The CDE is responsible for diversification and PS deterministically improves the quality of feasible solutions. The two chaos maps, namely, Gauss map and Tent map are considered for performance analysis of chaotic differential evolution and Powell's search (CDEPS). The comparison of CDE and CDEPS using generalized benchmark problems with Gauss and Tent map chaotic sequences clearly indicates that the CDEPS is better in terms of convergence speed as well as solution quality. Further, the comparison of generation cost for small power generation capacity system shows that the solution obtained by CDEPS-1 and 2 is either equal to or better than the solutions of other algorithms. While, dealing with MOLD it has been observed that CDEPS-1 and CDEPS-2 have obtained a better solution than other algorithms, except in few cases where fly and walk predator prey optimization and chaotic evolutionary programming and pattern search have competitive results with the proposed approach. However, CDEPS-1 and 2 have resulted in better solution for ELD and MOLD of high power generation capacity systems. The capability of CDEPS to obtain a better MOLD solution, in comparison to DE has been verified using Wilcoxon's signed rank test. The Tent map chaotic sequence gives a faster convergence rate as compared to the Gauss map chaotic sequence, when CDEPS is applied to solve MOLD. The study shows that the Tent map based CDEPS has merit over the others to obtain a better compromised solution as indicated by proposed scaling function. In addition, it has been observed that proposed scaling function infers better information of compromised solution than surrogate worth trade-off function. Hence, it is more promising criteria to decide compromised solution.

CHAPTER 7

CONCLUSIONS

7.1 INTRODUCTION

This chapter presents the salient and significant contributions of the thesis as well as a brief statement concerning some recommendations for further research. The significant original contributions are listed and discussed.

7.2 SIGNIFICANT CONTRIBUTIONS

The population based search techniques have appeared as an effective tool for optimization. These techniques are based on search ideologies that simultaneously explore multiple sub-regions of the search space as well as the exploitative interaction of different search tracks. These evolutionary search processes are usually inspired by nature's principles. The process is usually independent of problem's domain knowledge and depends on the fitness value of population individuals. The trend in research on optimization is focused on algorithms that incorporate higher level of evolutionary and adaptive behavior. The stochastic approach of these algorithms relies on either collective behavior mechanism or foraging activity to generate a replica having better fitness value. The exchange of information between population members eventually helps their movement in multiple directions and thereby increases the probability of locating global optima. The contributions of this research work focus on synergic composition of constitutive elements of an algorithm, to enhance its search capabilities. Alternatively, the incorporation of specific heuristics or operators that comprise domain knowledge have been utilized for perfor-

mance improvement. The proposed search procedures having well balanced properties of exploration and exploitation have been used to solve real life multiobjective constrained optimization problems of power system. The following significant contributions result from this research work.

1. A *Synergic Predator Prey Optimization* (SPPO) algorithm has been proposed as a modification of Predator prey optimization algorithm for the solution of economic load dispatch. The search process of proposed approach starts with the opposition based initialization of population and proceeds towards the solution of problem by using collaborative effort. The intelligence of prey has been split up to best and not-best components, while the fear factor of prey is treated as a collaborative function of cognitive and social behavior of prey particle. The proposed modifications ensures diversity of solutions during the search process, thereby effectively exploring the search space. The proposed algorithm is presented as a solution approach to solve the *Economic Load Dispatch* (ELD). The operating cost functions of generating units are non-smooth and involve complexity because of additional constraints like multi-fuel options, ramp rate limits, and prohibited zones. The equality constraint has been taken care off by employing the concept of the variable elimination method. The applied approach ensures that prey guiding components do not stay at a particular position for a large number of iterations. The result obtained are compared with others available in literature and have been found better. However, the results of synergic predator prey optimization are competitive with composite particle swarm optimization and continuous quick group search algorithm. The numerical results reveal the better practicability and mature convergence capability of the proposed algorithm.
2. Another effort has been done with an objective to improve the search capability of the evolutionary programming algorithm. The proposed modification relies on chaotic map sequences instead of random numbers and is termed as chaotic evolutionary programming. Further, Powell's pattern search has been hybridized as an interleaved search process under stochastic control. The two chaotic map sequences

namely Gauss map and Tent map are considered for performance analysis of the proposed approach. The proposed algorithm *Chaotic Evolutionary Programming and Powell's Method* (CEPPS) has been investigated as a solution procedure to solve *Multiobjective Load Dispatch Problem* (MOLD). The conflicting objectives of MOLD have been taken care by an interactive surrogate worth trade-off approach. This method provides an interface between the decision maker and the mathematical model through surrogate worth trade-off function. Moreover, the surrogate worth trade-off method has an advantage of simple mechanism and requires few non-inferior solutions to arrive at the best solution, thus enhances computational efficiency. The experimentation with test power systems has shown that for a smaller power generation system, the CEPPS variants have obtained a better solution than the results available in the literature. In specific cases, the results are competitive with synergic predator prey optimization and composite particle swarm optimization algorithm. However, while handling the ELD of higher power generation capacity systems, it has resulted in premature convergence. Similar, are the observations while performing the MOLD of test power systems.

3. An interactive method has been applied to handle the conflicting objective functions, where non-inferior solutions are generated by implementing the weighting method, and decision-making has been done using fuzzy set theory. The computational effort to obtain complete Pareto surface has been reduced by searching the preference weights along with generator power using adaptive predator prey optimization technique. The *Adaptive Predator Prey Optimization* (APPO) has been constituted by reinforcement, inhabitation, discrete velocity variation and adaptive velocity limit concepts. The suggested method has been employed on three test power systems. The applied approach results in obtaining well distributed Pareto front in a single run. The comparison of the numerical results and the convergence characteristics of the overall membership function values confirm the effectiveness and the superiority of the proposed algorithm in majority of simulation studies. However, in some of the cases the APPO's solution is competitive with other algorithms. For smaller test power system the best solution obtained is comparable with the syner-

gic predator prey optimization and chaotic evolutionary programming and Powell's method. Similarly, for medium power generation capacity system, the quality of compromised solution is little bit degraded in comparison to the modulated artificial bee colony and modulated particle swarm optimization, respectively. While dealing with ELD of the high power generation system, the cost is a little bit higher than the synergic predator prey optimization, however the MOLD solution is better than its counterparts.

4. The fly and walk predator prey optimization presents a unique combination of fly, walk and direction of turn as a search process for solution of multiobjective load dispatch problem. The fly search process mimics the predator prey behavior whereas, walk process is motivated by Hooke-Jeeves method. The *Fly and Walk Predator Prey Optimization* (FWPO) algorithm maintains useful diversity by switching exploration and exploitation phase in a cyclic process. The direction of turn is another aspect of FWPO, which bias the optimization process towards the feasible region without restricting the improvement in objective functions. This approach is responsible for maintaining power balance constraint and decision boundary constraint. The comparison of the results shows that in a few cases of the small test power system, FWPO solution is competitive with results of synergic predator prey optimization, adaptive predator prey optimization and chaotic evolutionary programming and Powell's method algorithm. However, for bigger systems, the FWPO results are better than its counterparts. The U-measure of Pareto-front indicates that the proposed approach is able to generate uniformly distributed non-dominated solutions for multiple objectives. However, a limitation of proposed surrogate worth trade-off function has been identified that in a particular situation, it is unable to convey conclusive information to select a compromised solution.
5. The surrogate worth trade-off approach has been modified and a scaling function has been proposed to solve multiobjective problem. The use of scaling function as a non-interactive approach eliminates the additional computational overhead for sorting of Pareto-front. The conflicting objectives being vague, the proposed func-

tion implicates fuzzy theory. A *Chaotic Differential Evolution and Powell's method* (CDEPS) is implemented for power scheduling of multi-objective thermal power load dispatch problem. The proposed approach relies on chaotic sequence instead of uniformly distributed random numbers. In the search process chaotic differential evolution is responsible for diversification, whereas Powell's pattern search is used an interleaved operator to deterministically improve the quality of feasible solutions. The two chaos maps namely, Gauss map and Tent map are considered for performance analysis of chaotic differential evolution and Powell's search. The experimental studies prove that proposed approach performs better than its parent technique chaotic differential evolution. Further, the comparison of generation cost for small power generation capacity system shows that the solution obtained by CDEPS is either equal to or better than the solutions of other algorithms. While, dealing with MOLD it has been observed that CDEPS has obtained better solution than other algorithms, except in a few cases where fly and walk predator prey optimization and chaotic evolutionary programming and pattern search have competitive results with the proposed approach. However, CDEPS have resulted in a better solution for ELD and MOLD of high power generation capacity systems. In addition, it is observed that the Gauss map based approach generates a better quality solution on single objective problems. Whereas, the Tent map chaotic sequence gives a faster convergence rate as compared to the Gauss map chaotic sequence, when CDEPS is applied to solve MOLD. The study shows that the Tent map based CDEPS has merit over the others to obtain a better compromised solution as indicated by proposed scaling function. Moreover, it has been observed that proposed scaling function infers better information about the compromised solution than surrogate worth trade-off function. Hence, it is a more promising criteria to decide compromised solution.

6. In this research work, five modified versions of evolutionary programming, predator prey optimization and differential evolution have been proposed. The modified versions are namely SPPO, CEPPS, APPO, FWPO and CDEPS. These algorithms have been applied to solve the realistic ELD and MOLD problems. Three test

power system having small, medium and high power generation capacity has been considered in this research work. These systems have been further divided into cases depending on the complexity due to valve point loading effect, prohibited operating zone avoidance and ramp rate limits. The key observations on the basis of solution quality comparison among SPPO, CEPPS, APPO, FWPO and CDEPS respectively are summarized. The CDEPS has performed better on all the three test power systems, in all the cases undertaken to study. The CDEPS algorithm has resulted in a more uniformly distributed Pareto front, while solving the MOLD problems. FWPO has marginally lagged in performance than CDEPS algorithm. The performance of APPO and SPPO algorithm falls after the performance of CDEPS and FWPO. CEPPS algorithm provides good results for small power generation capacity system. The observations of research work finally recommends CDEPS algorithm to solve ELD and MOLD problem.

7. In this research work three different approaches *viz.* surrogate worth trade-off function, satisficing function and scaling function have been implemented to decide a compromised solution of the multi objective problem. It has been analyzed on the basis of the obtained results that the scaling function based decision approach provides unambiguous information to opt the solution, while solving MOLD using interactive as well as non-interactive approach.
8. Among the approaches to solve multiobjective problem, *i.e.* interactive and non-interactive approach, the interactive approach has advantage to generate a set of compromised solution. However, this advantage is at the cost of additional computation resources requirement. The non-interactive approach offers advantages of low computational complexity and provides a better quality single solution in lesser time.

7.3 SUGGESTIONS FOR FURTHER RESEARCH

There is no end of research work as end point of research in any field cannot be defined. The scope of improvement is always present. The following points are suggested as future

scope for the work presented.

- The incorporation of evolutionary and adaptive operators based on foraging activity can be a approach to universal solution procedure.
- In future, the walk process will be explored as modification of fly process by adjusting the velocity limits. Also, another important aspect of FWPPPO algorithm, a process responsible for maintaining constraints in specified limits, can be modified to operate as velocity perturbed mechanism.
- The use of other existing chaotic sequences in the search process is a subject of future research and investigation.
- To incorporate the concept of Pareto front uniformity at the level of the solution search process is an another interesting research area.
- The consideration of algorithm hybridization process from natural foraging behavior point of view is a big lacuna.
- Recently, a wide variety of nature based search algorithms have been materialized. The fusion of these algorithms among themselves or hybridization with local search techniques is a potential research area.

REFERENCES

- [1] John F Nash Jr. The bargaining problem. *Econometrica: Journal of the Econometric Society*, pages 155–162, 1950.
- [2] Abraham Charnes and William W Cooper. Management models and industrial applications of linear programming. *Management Science*, 4(1):38–91, 1957.
- [3] Robert Hooke and Terry A Jeeves. Direct search solution of numerical and statistical problems. *Journal of the Association for Computing Machinery*, 8(2):212–229, 1961.
- [4] RJ Ringlee and DD Williams. Economic system operation considering valve throttling losses ii-distribution of system loads by the method of dynamic programming. *Transactions of the American Institute of Electrical Engineers. Part III: Power Apparatus and Systems*, 81(3):615–620, 1962.
- [5] Roger Fletcher and Michael JD Powell. A rapidly convergent descent method for minimization. *The Computer Journal*, 6(2):163–168, 1963.
- [6] Lofti Zadeh. Optimality and non-scalar-valued performance criteria. *IEEE Transactions on Automation and Control*, 8(1):59–60, 1963.
- [7] William F Tinney and Clifford E Hart. Power flow solution by newton’s method. *IEEE Transactions on Power Apparatus and systems*, (11):1449–1460, 1967.
- [8] M Gent and John Lamont. Minimum-emission dispatch. *IEEE Transactions on Power Apparatus System*, 6(PAS-90):2650–2660, 1971.
- [9] Yacov Y Haimes, LS Ladson, and David A Wismer. Bicriterion formulation of problems of integrated system identification and system optimization. *IEEE Transactions on System Man Cybernetics Part A System Humans*, (3):296, 1971.
- [10] Arthur M Geoffrion, James S Dyer, and A Feinberg. An interactive approach for multi-criterion optimization, with an application to the operation of an academic department. *Management Science*, 19(4-part-1):357–368, 1972.
- [11] Gerald F Reid and Lawrence Hasdorff. Economic dispatch using quadratic programming. *IEEE Transactions on Power Apparatus System*, (6):2015–2023, 1973.
- [12] Po-Lung Yu. A class of solutions for group decision problems. *Management Science*, 19(8):936–946, 1973.
- [13] H H Happ. Optimal power dispatch. *IEEE Transactions on Power Apparatus System*, PAS-93(3):820–830, May 1974.
- [14] F Gembicki and Y Haimes. Approach to performance and sensitivity multiobjective optimization: The goal attainment method. *IEEE Transactions on Automatic Control*, 20(6):769–771, 1975.
- [15] Jacob Zahavi and Lawrence Eisenberg. Economic-environmental power dispatch. *IEEE Transactions on System Man Cybernetics*, (5):485–489, 1975.
- [16] Ralph L Keeney, Howard Raiffa, and David W Rajala. Decisions with multiple objectives: Preferences and value trade-offs. *IEEE Transactions on System Man Cybernetics*, 9(7):403–403, 1979.
- [17] WG Wood. Spinning reserve constrained static and dynamic economic dispatch. *IEEE Transactions on Power Apparatus System*, (2):381–388, 1982.
- [18] Scott Kirkpatrick, C Daniel Gelatt, and Mario P Vecchi. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983.

-
- [19] Ralph E Steuer and Eng-Ung Choo. An interactive weighted tchebycheff procedure for multiple objective programming. *Mathematical Programming*, 26(3):326–344, 1983.
- [20] J David Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. In *1st International Conference on Genetic Algorithms*, pages 93–100, 1985.
- [21] Juhani Koski and Risto Silvennoinen. Norm methods and partial weighting in multicriterion optimization of structures. *International Journal of Numerical Methods for Engineers*, 24(6):1101–1121, 1987.
- [22] A Monticelli, MVF Pereira, and S Granville. Security-constrained optimal power flow with post-contingency corrective rescheduling. *IEEE Transactions on Power System*, 2(1):175–180, 1987.
- [23] David E Goldberg and John H Holland. Genetic algorithms and machine learning. *Machine learning*, 3(2):95–99, 1988.
- [24] J Nanda, DP Kothari, and KS Lingamurthy. Economic-emission load dispatch through goal programming techniques. *IEEE Transactions on Energy Conversion*, 3(1):26–32, 1988.
- [25] RE Tyson and GT Heydt. An expert system and dynamic programming hybrid for unit commitment. In *21th Annual North-American Power Symposium*, pages 2–11, 1989.
- [26] Badrul H Chowdhury and Saifur Rahman. A review of recent advances in economic dispatch. *IEEE Transactions on Power Systems*, 5(4):1248–1259, 1990.
- [27] Bahman S Kermanshahi, Yue Wu, Keiichiro Yasuda, and Ryuichi Yokoyama. Environmental marginal cost evaluation by non-inferiority surface. *IEEE Transactions on Power System*, 5(4):1151–1159, 1990.
- [28] Donald F Specht. Probabilistic neural networks. *Neural networks*, 3(1):109–118, 1990.
- [29] Khalil S Hindi and MR Ab Ghani. Dynamic economic dispatch for large scale power systems: a lagrangian relaxation approach. *International Journal of Electrical Power and Energy System*, 13(1):51–56, 1991.
- [30] C Palanichamy and K Srikrishna. Economic thermal power dispatch with emission constraint. *Journal of Institute of Engineers (India)*, 72(April):11–18, 1991.
- [31] GP Granelli, M Montagna, GL Pasini, and P Marannino. Emission constrained dynamic dispatch. *Electrical Power System Research*, 24(1):55–64, 1992.
- [32] Z-X Liang and J Duncan Glover. A zoom feature for a dynamic programming solution to economic dispatch including transmission losses. *IEEE Transactions on Power System*, 7(2):544–550, 1992.
- [33] Chin E Lin, Shiuan Tai Chen, and C-L Huang. A direct newton-raphson economic dispatch. *IEEE Transactions on Power System*, 7(3):1149–1154, 1992.
- [34] Thomas Bäck and Hans-Paul Schwefel. An overview of evolutionary algorithms for parameter optimization. *Evolutionary Computation*, 1(1):1–23, 1993.
- [35] Chern-Lin Chen and Shun-Chung Wang. Branch-and-bound scheduling for thermal generating units. *IEEE Transactions on Energy Conversion*, 8(2):184–189, 1993.
- [36] Dipankar DasGupta. An overview of artificial immune systems and their applications. In *Artificial Immune Systems and their Applications*, pages 3–21. Springer, 1993.
- [37] JS Dhillon, SC Parti, and DP Kothari. Stochastic economic emission load dispatch. *Electrical Power System Research*, 26(3):179–186, 1993.
- [38] David B Fogel. Applying evolutionary programming to selected traveling salesman problems. *Cybernetics and systems*, 24(1):27–36, 1993.
- [39] Carlos M Fonseca and Peter J Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In *5th International Conference on Genetic Algorithm*, volume 93, pages 416–423, 1993.
- [40] G Klein and D Klinger. Natural decision making. *Human System IAC Gateway*, 2(1):16–19, 1993.
- [41] Fred N Lee and Arthur M Breipohl. Reserve constrained economic dispatch with prohibited operating zones. *IEEE Transactions on Power System*, 8(1):246–254, 1993.
-

-
- [42] JH Park, YS Kim, IK Eom, and KY Lee. Economic load dispatch for piecewise quadratic cost function using hopfield neural network. *IEEE Transactions on Power System*, 8(3):1030–1038, 1993.
- [43] Chunyan Wang and SM Shahidehpour. Effects of ramp-rate limits on unit commitment and economic dispatch. *IEEE Transactions on Power System*, 8(3):1341–1350, 1993.
- [44] KP Wong and CC Fung. Simulated annealing based economic dispatch algorithm. In *IEE Proceeding Generation Transmission Distribution*, volume 140, pages 509–515, 1993.
- [45] Peter J Angeline, Gregory M Saunders, and Jordan B Pollack. An evolutionary algorithm that constructs recurrent neural networks. *IEEE Transactions on Neural Networks*, 5(1):54–65, 1994.
- [46] AA El-Keib, H Ma, and JL Hart. Economic dispatch in view of the clean air act of 1990. *IEEE Transactions on Power System*, 9(2):972–978, 1994.
- [47] Lawrence J Fogel and David B Fogel. A preliminary investigation on extending evolutionary programming to include self-adaptation on finite state. *Informatica*, 18(4), 1994.
- [48] Jeffrey Horn, Nicholas Nafpliotis, and David E Goldberg. A niched pareto genetic algorithm for multiobjective optimization. In *IEEE World Congress Evolutionary Computation*, pages 82–87, 1994.
- [49] Mitchell A Potter and Kenneth A De Jong. A cooperative coevolutionary approach to function optimization. In *International Conference on Parallel Problem Solving from Nature*, pages 249–257. Springer, 1994.
- [50] Nidamarthi Srinivas and Kalyanmoy Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3):221–248, 1994.
- [51] JH Talaq, Ferial El-Hawary, and ME El-Hawary. A summary of environmental/economic dispatch algorithms. *IEEE Transactions on Power System*, 9(3):1508–1516, 1994.
- [52] Po-Hung Chen and Hong-Chan Chang. Large-scale economic dispatch by genetic algorithm. *IEEE Transactions on power systems*, 10(4):1919–1926, 1995.
- [53] Ahmed Farag, Samir Al-Baiyat, and TC Cheng. Economic load dispatch multiobjective optimization procedures using linear programming techniques. *IEEE Transactions on Power System*, 10(2):731–738, 1995.
- [54] James Kennedy and Russell Eberhart. Particle swarm optimization. In *Proceeding IEEE International Conference Neural Networks, 1995.*, volume 4, pages 1942–1948, Nov 1995.
- [55] T Denise King, ME El-Hawary, and Ferial El-Hawary. Optimal environmental dispatching of electric power systems via an improved hopfield neural network model. *IEEE Transactions on Power System*, 10(3):1559–1565, 1995.
- [56] J Kumar and Gerald B Sheblé. Clamped state solution of artificial neural network for real-time economic dispatch. *IEEE Transactions on Power System*, 10(2):925–931, 1995.
- [57] JW Lamont and EV Obessis. Emission dispatch models and algorithms for the 1990s. *IEEE Transactions on Power System*, 10(2):941–947, 1995.
- [58] Yasemin Aksoy, Timothy W Butler, and Elliott D Minor. Comparative studies in interactive multiple objective mathematical programming. *European Journal of Operational Research*, 89(2):408–422, 1996.
- [59] Timothy Ward Athan and Panos Y Papalambros. A note on weighted criteria methods for compromise solutions in multi-objective optimization. *Engineering Optimization*, 27(2):155–176, 1996.
- [60] Tobias Blickle and Lothar Thiele. A comparison of selection schemes used in evolutionary algorithms. *Evolutionary Computation*, 4(4):361–394, 1996.
- [61] Franklin Y Cheng and Dan Li. Multiobjective optimization of structures with and without control. *Journal of guidance, control and dynamics*, 19(2):392–397, 1996.
- [62] M Dorigo, V Maniezzo, and A Colorni. Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on System Man Cybernetics Part B Cybernetics*, 26(1):29–41, 1996.
- [63] Achille Messac. Physical programming-effective optimization for computational design. *The American Institute of Aeronautics and Astronautics Journal*, 34(1):149–158, 1996.
-

-
- [64] Zbigniew Michalewicz. Heuristic methods for evolutionary computation techniques. *Journal of Heuristics*, 1(2):177–206, 1996.
- [65] Hyun Myung and Jong-Hwan Kim. Hybrid evolutionary programming for heavily constrained problems. *BioSystems*, 38(1):29–43, 1996.
- [66] SO Orero and Malcolm R Irving. Economic dispatch of generators with prohibited operating zones: a genetic algorithm approach. In *IEE Proceedings on Generation, Transmission and Distribution*, volume 143, pages 529–534, 1996.
- [67] Hong-Tzer Yang, Pai-Chuan Yang, and Ching-Lien Huang. Evolutionary programming based economic dispatch for units with non-smooth fuel cost functions. *IEEE Transactions on Power System*, 11(1):112–118, 1996.
- [68] Xin Yao and Yong Liu. Fast evolutionary programming. *Evolutionary Programming*, 3:451–460, 1996.
- [69] Kumar Chellapilla and David B Fogel. Two new mutation operators for enhanced search and optimization in evolutionary programming. In *Optical Science, Engineering and Instrumentation*, pages 260–269, 1997.
- [70] Indraneel Das and John E Dennis. A closer look at drawbacks of minimizing weighted sums of objectives for pareto set generation in multicriteria optimization problems. *Structural and multi-disciplinary optimization*, 14(1):63–69, 1997.
- [71] James Kennedy and Russell C Eberhart. A discrete binary version of the particle swarm algorithm. In *IEEE International Conference Systems, Man, and Cybernetics*, volume 5, pages 4104–4108, 1997.
- [72] Jong-Hwan Kim and Hyun Myung. Evolutionary programming techniques for constrained optimization problems. *IEEE Transactions on Evolutionary Computing*, 1(2):129–140, 1997.
- [73] YH Song, GS Wang, AT Johns, and PY Wang. Distribution network reconfiguration for loss reduction using fuzzy controlled evolutionary programming. *IEE Proceeding Generation Transmission and Distribution*, 144(4):345–350, 1997.
- [74] YH Song, GS Wang, PY Wang, and AT Johns. Environmental/economic dispatch using fuzzy logic controlled genetic algorithms. *IEE Proceeding Generation Transmission and Distribution*, 144(4):377–382, 1997.
- [75] Rainer Storn and Kenneth Price. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359, 1997.
- [76] Ching-Tzong Su and Gwo-Jen Chiou. A fast-computation hopfield method to economic dispatch of power systems. *IEEE Transactions on Power System*, 12(4):1759–1764, 1997.
- [77] MP Walsh and MJ O’malley. Augmented hopfield network for unit commitment and economic dispatch. *IEEE Transactions on Power System*, 12(4):1765–1774, 1997.
- [78] David H Wolpert and William G Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computing*, 1(1):67–82, 1997.
- [79] Peter J Angeline. Evolutionary optimization versus particle swarm optimization: Philosophy and performance differences. In *International Conference on Evolutionary Programming*, pages 601–610. Springer, 1998.
- [80] Ji-Pyng Chiou and Feng-Sheng Wang. A hybrid method of differential evolution with application to optimal control problems of a bioprocess system. In *IEEE World Congress on Computational Intelligence*, pages 627–632, 1998.
- [81] Indraneel Das and John E Dennis. Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems. *Journal of Society for Industrial and Applied Mathematics*, 8(3):631–657, 1998.
- [82] Hisao Ishibuchi and Tadahiko Murata. A multi-objective genetic local search algorithm and its application to flowshop scheduling. *IEEE Transactions on System Man Cybernetics Part C Applications and Reviews*, 28(3):392–403, 1998.
- [83] Kwang Y Lee, Arthit Sode-Yome, and June Ho Park. Adaptive hopfield neural networks for economic load dispatch. *IEEE Transactions on Power System*, 13(2):519–526, 1998.
-

-
- [84] Ralf Salomon. Evolutionary algorithms and gradient search: similarities and differences. *IEEE Transactions on Evolutionary Computation*, 2(2):45–55, 1998.
- [85] Yuhui Shi and Russell Eberhart. A modified particle swarm optimizer. In *IEEE World Congress on Computational Intelligence*, pages 69–73, 1998.
- [86] David A Van Veldhuizen and Gary B Lamont. Evolutionary computation and convergence to a pareto front. In *Genetic programming conference*, pages 221–228, 1998.
- [87] T Yalcinoz and MJ Short. Neural networks approach for solving economic dispatch problem with transmission capacity constraints. *IEEE Transactions on Power System*, 13(2):307–313, 1998.
- [88] A Carlos and Coello Coello. An updated survey of evolutionary multiobjective optimization techniques: State of the art and future trends. In *Congress on Evolutionary Computation*, pages 3–13, 1999.
- [89] Furong Gao, Mingzhong Li, Fuli Wang, Baoguo Wang, and PoLock Yue. Genetic algorithms and evolutionary programming hybrid strategy for structure and weight learning for multilayer feedforward neural networks. *Industrial and Engineering chemistry research*, 38(11):4330–4336, 1999.
- [90] Joshua Knowles and David Corne. The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation. In *Congress on Evolutionary Computation*, volume 1, 1999.
- [91] Xin Yao, Yong Liu, and Guangming Lin. Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation*, 3(2):82–102, 1999.
- [92] Eckart Zitzler and Lothar Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computing*, 3(4):257–271, 1999.
- [93] JS Dhillon and DP Kothari. The surrogate worth trade-off approach for multiobjective thermal power dispatch problem. *Electric Power System Research*, 56(2):103–110, 2000.
- [94] PK Chattopadhyay P K. Hota, R. Chakrabarti. A simulated annealing-based goal-attainment method for economic emission load dispatch with nonsmooth fuel cost and emission level functions. *Electric Machines and Power Systems*, 28(11):1037–1051, 2000.
- [95] Sin-Jun Kang, Chun-Hee Woo, Hee-Soo Hwang, and Kwang B Woo. Evolutionary design of fuzzy rule base for nonlinear system modeling and control. *IEEE Transactions on Fuzzy Systems*, 8(1):37–45, 2000.
- [96] Yiu-Wing Leung and Yuping Wang. Multiobjective programming using uniform design and genetic algorithm. *IEEE Transactions on System Man Cybernetics Part C Applications and Reviews*, 30(3):293–304, 2000.
- [97] Robert Michael Lewis, Virginia Torczon, and Michael W Trosset. Direct search methods: then and now. *Journal of Computing and Applied Math*, 124(1):191–207, 2000.
- [98] Achille Messac, Cyriaque Puemi-Sukam, and Emanuel Melachrinoudis. Aggregate objective functions and pareto frontiers: required relationships and practical implications. *Optimization Engineers*, 1(2):171–188, 2000.
- [99] Hussein A Abbass, Ruhul Sarker, and Charles Newton. PDE: a pareto-frontier differential evolution approach for multi-objective optimization problems. In *Proceedings of the Congress on Evolutionary Computation*, volume 2, pages 971–978, 2001.
- [100] F Bergh and AP Engelbrecht. Effects of swarm size on cooperative particle swarm optimisers. In *3rd Annual Conference Genetic and Evolutionary Computation*, pages 892–899, 2001.
- [101] Zong Woo Geem, Joong Hoon Kim, and GV Loganathan. A new heuristic optimization algorithm: harmony search. *Simulation*, 76(2):60–68, 2001.
- [102] T Hendtlass and M Randall. A survey of ant colony and particle swarm meta-heuristics and their application to discrete optimization problems. In *Proceeding Inaugural Workshop on Artificial Life*, pages 15–25, 2001.
-

-
- [103] Tim Hendtlass. A combined swarm differential evolution algorithm for optimization problems. In *International Conference Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pages 11–18. Springer, 2001.
- [104] Whei-Min Lin, Fu-Sheng Cheng, and Ming-Tong Tsay. Nonconvex economic dispatch by integrated artificial intelligence. *IEEE Transactions on Power System*, 16(2):307–311, 2001.
- [105] Yong Liu, Xin Yao, Qiangfu Zhao, and Tetsuya Higuchi. Scaling up fast evolutionary programming with cooperative coevolution. In *Proceedings of Congress on Evolutionary Computation*, volume 2, pages 1101–1108, 2001.
- [106] Morten Lovbjerg, Thomas Kiel Rasmussen, and Thiemo Krink. Hybrid particle swarm optimizer with breeding and subpopulations. In *Genetic and evolutionary computation conference*, volume 2, pages 469–476, 2001.
- [107] CK Mohan and B Al-Kazemi. Discrete particle swarm optimization. In *Workshop on Particle Swarm Optimization, Indianapolis*, 2001.
- [108] Yog Raj Sood, S Verma, N Prasad Padhy, and HO Gupta. Evolutionary programming based algorithm for selection of wheeling options. In *Power Engineering Society Winter Meeting*, volume 2, pages 545–552, 2001.
- [109] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. SPEA2: Improving the strength pareto evolutionary algorithm. In *Eurogen*, volume 3242, pages 95–100, 2001.
- [110] Pathom Attaviriyapap, Hiroyuki Kita, Eiichi Tanaka, and Jun Hasegawa. A hybrid EP and SQP for dynamic economic dispatch with nonsmooth fuel cost function. *IEEE Transactions on Power System*, 17(2):411–416, 2002.
- [111] Maurice Clerc and James Kennedy. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computing*, 6(1):58–73, 2002.
- [112] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and Tamt Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computing*, 6(2):182–197, 2002.
- [113] Agoston E Eiben and Marc Schoenauer. Evolutionary computing. *Information Processing Letters*, 82(1):1–6, 2002.
- [114] James Kennedy and Rui Mendes. Population structure and particle swarm performance. In *Proceedings of Congress on Evolutionary Computation*, volume 2, pages 1671–1676, 2002.
- [115] Joshua D Knowles. *Local-search and hybrid evolutionary algorithms for Pareto optimization*. PhD thesis, Department of Computer Science, University of Reading, 2002.
- [116] Achille Messac and Christopher A Mattson. Generating well-distributed sets of pareto points for engineering design using physical programming. *Optimization Engineers*, 3(4):431–450, 2002.
- [117] Vladimiro Miranda and Nuno Fonseca. New evolutionary particle swarm algorithm applied to voltage control. In *14th power systems computation conference*, pages 1–6, 2002.
- [118] Arlindo Silva, Ana Neves, and Ernesto Costa. An empirical comparison of particle swarm and predator prey optimisation. In *Artificial Intelligence and Cognitive Science*, pages 103–110. Springer, 2002.
- [119] Chengjian Wei, Zhenya He, Yifeng Zhang, and Wenjiang Pei. Swarm directions embedded in fast evolutionary programming. In *Congress on Evolutionary Computation*, volume 9, page 11, 2002.
- [120] Kit Po Wong and Cameron Algie. Evolutionary programming approach for combined heat and power dispatch. *Electrical Power System Research*, 61(3):227–232, 2002.
- [121] MA Abido. Environmental/economic power dispatch using multiobjective evolutionary algorithms. *IEEE Transactions on Power System*, 18(4):1529–1537, 2003.
- [122] MA Abido. A niched pareto genetic algorithm for multiobjective environmental/economic dispatch. *International Journal of Electrical Power and Energy System*, 25(2):97–105, 2003.
- [123] Ş İlker Birbil and Shu-Chering Fang. An electromagnetism-like mechanism for global optimization. *Journal of Global Optimization*, 25(3):263–282, 2003.
-

-
- [124] MY El-Sharkh and AA El-Keib. Maintenance scheduling of generation and transmission systems using fuzzy evolutionary programming. *IEEE Transactions on Power System*, 18(2):862–866, 2003.
- [125] Hui-Yuan Fan and Jouni Lampinen. A trigonometric mutation operation to differential evolution. *Journal of Global Optimization*, 27(1):105–129, 2003.
- [126] Zue-Lee Gaing. Particle swarm optimization to solving the economic dispatch considering the generator constraints. *IEEE Transactions on Power System*, 18(3):1187–1195, 2003.
- [127] Natsuki Higashi and Hitoshi Iba. Particle swarm optimization with gaussian mutation. In *IEEE Swarm Intelligence Symposium*, pages 72–79, 2003.
- [128] James Kennedy. Bare bones particle swarms. In *IEEE Swarm Intelligence Symposium*, pages 80–87, 2003.
- [129] Yiu-Wing Leung and Yuping Wang. U-measure: a quality measure for multiobjective programming. *IEEE Transactions on System Man Cybernet Part A: Syst Humans*, 33(3):337–343, 2003.
- [130] Xiaodong Li. A non-dominated sorting particle swarm optimizer for multiobjective optimization. In *Genetic and Evolutionary Computation Conference*, pages 37–48. Springer, 2003.
- [131] Achille Messac, Amir Ismail-Yahaya, and Christopher A Mattson. The normalized normal constraint method for generating the pareto frontier. *Structural and multidisciplinary optimization*, 25(2):86–98, 2003.
- [132] Shigenori Naka, Takamu Genji, Toshiki Yura, and Yoshikazu Fukuyama. A hybrid particle swarm optimization for distribution state estimation. *IEEE Transactions on Power System*, 18(1):60–68, 2003.
- [133] CA Roa-Sepulveda and BJ Pavez-Lazo. A solution to the optimal power flow using simulated annealing. *International Journal of Electrical Power and Energy System*, 25(1):47–57, 2003.
- [134] A Sinha and DE Goldberg. A survey of hybrid genetic and evolutionary algorithms, 2003.
- [135] Nidul Sinha, R Chakrabarti, and PK Chattopadhyay. Evolutionary programming techniques for economic load dispatch. *IEEE Transactions on Evolutionary Computing*, 7(1):83–94, 2003.
- [136] B Venkatesh and Rakesh Ranjan. Optimal radial distribution system reconfiguration using fuzzy adaptation of evolutionary programming. *International Journal of Electrical Power and Energy System*, 25(10):775–780, 2003.
- [137] Feng Xue, Arthur C Sanderson, and Robert J Graves. Pareto-based multi-objective differential evolution. In *The Congress on Evolutionary Computation*, volume 2, pages 862–869, 2003.
- [138] Wen-Jun Zhang, Xiao-Feng Xie, et al. DEPSO: hybrid particle swarm with differential evolution operator. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 4, pages 3816–3821, 2003.
- [139] M Basu. An interactive fuzzy satisfying method based on evolutionary programming technique for multiobjective short-term hydrothermal scheduling. *Electrical Power System Research*, 69(2):277–285, 2004.
- [140] John E Bell and Patrick R McMullen. Ant colony optimization techniques for the vehicle routing problem. *Advanced Engineering Informatics*, 18(1):41–48, 2004.
- [141] Ji-Pyng Chiou, Chung-Fu Chang, and Ching-Tzong Su. Ant direction hybrid differential evolution for solving large capacitor placement problems. *IEEE Transactions on Power System*, 19(4):1794–1800, 2004.
- [142] Carlos A Coello Coello, Gregorio Toscano Pulido, and M Salazar Lechuga. Handling multiple objectives with particle swarm optimization. *IEEE Transactions on Evolutionary Computing*, 8(3):256–279, 2004.
- [143] V Gopalakrishnan, P Thirunavukkarasu, and R Prasanna. Reactive power planning using hybrid evolutionary programming method. In *IEEE Power Systems Conference and Exposition*, pages 1319–1323, 2004.
- [144] J Hao, LB Shi, and Ch Chen. Optimising location of unified power flow controllers by means of improved evolutionary programming. *IEE Proceeding Generation Transmission and Distribution*, 151(6):705–712, 2004.
-

-
- [145] Chia-Feng Juang. A hybrid of genetic algorithm and particle swarm optimization for recurrent network design. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 34(2):997–1006, 2004.
- [146] Chang-Yong Lee and Xin Yao. Evolutionary programming using mutations based on the lévy probability distribution. *IEEE Transactions on Evolutionary Computation*, 8(1):1–13, 2004.
- [147] R Timothy Marler and Jasbir S Arora. Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization*, 26(6):369–395, 2004.
- [148] Rui Mendes, James Kennedy, and José Neves. The fully informed particle swarm: simpler, maybe better. *IEEE Transactions on Evolutionary Computing*, 8(3):204–210, 2004.
- [149] R Naresh, J Dubey, and J Sharma. Two-phase neural network based modelling framework of constrained economic load dispatch. *IEE Proceeding Generation Transmission and Distribution*, 151(3):373–378, 2004.
- [150] C Christoper Asir Rajan and MR Mohan. An evolutionary programming-based tabu search method for solving the unit commitment problem. *IEEE Transactions on Power System*, 19(1):577–585, 2004.
- [151] Carlos Romero. A general structure of achievement function for a goal programming model. *European Journal of Operational Research*, 153(3):675–686, 2004.
- [152] Nidul Sinha and B Purkayastha. Pso embedded evolutionary programming technique for nonconvex economic load dispatch. In *IEEE Power Systems Conference and Exposition*, pages 66–71, 2004.
- [153] P Somasundaram, K Kuppusamy, and RP Kumudini Devi. Evolutionary programming based security constrained optimal power flow. *Electrical Power System Research*, 72(2):137–145, 2004.
- [154] Rene Thomsen. Multimodal optimization using crowding-based differential evolution. In *Congress on Evolutionary Computation*, volume 2, pages 1382–1389, 2004.
- [155] Frans Van den Bergh and Andries P Engelbrecht. A cooperative approach to particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 8(3):225–239, 2004.
- [156] T Aruldoss Albert Victoire and A Ebenezer Jeyakumar. Hybrid pso-sqp for economic dispatch with valve-point effect. *Electrical Power System Research*, 71(1):51–59, 2004.
- [157] Xiaozhan Xu. A note on the subjective and objective integrated approach to determine attribute weights. *European Journal of Operational Research*, 156(2):530–532, 2004.
- [158] Wei Yan, Shuai Lu, and David C Yu. A novel optimal reactive power dispatch method based on an improved hybrid evolutionary programming technique. *IEEE Transactions on Power System*, 19(2):913–918, 2004.
- [159] Eckart Zitzler and Simon Künzli. Indicator-based selection in multiobjective search. In *International Conference Parallel Problem Solving from Nature*, pages 832–842. Springer, 2004.
- [160] Farzaneh Afshinmanesh, Alireza Marandi, and Ashkan Rahimi-Kian. A novel binary particle swarm optimization method using artificial immune system. In *The International Conference on Computer as a Tool*, volume 1, pages 217–220, 2005.
- [161] RC Bansal. Optimization methods for electric power systems: An overview. *International Journal of Emerging Electrical Power Systems*, 2(1), 2005.
- [162] Paul K Bergey and Cliff Ragsdale. Modified differential evolution: a greedy random strategy for genetic recombination. *Omega*, 33(3):255–265, 2005.
- [163] Chao-Lung Chiang. Improved genetic algorithm for power economic dispatch of units with valve-point effects and multiple fuels. *IEEE Transactions on Power System*, 20(4):1690–1699, 2005.
- [164] Jiang Chuanwen and Etorre Bompard. A hybrid method of chaotic particle swarm optimization and linear interior for reactive power optimization. *Mathematics and computers in Simulation*, 68(1):57–65, 2005.
- [165] Nader Damavandi and Safieddin Safavi-Naeini. A hybrid evolutionary programming method for circuit optimization. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 52(5):902–910, 2005.
-

-
- [166] MY El-Sharkh, AKMM Rahman, and MS Alam. Evolutionary programming-based methodology for economical output power from pem fuel cell for micro-grid application. *Journal of power sources*, 139(1):165–169, 2005.
- [167] Ahmed AA Esmim, Germano Lambert-Torres, and AC Zambroni de Souza. A hybrid particle swarm optimization applied to loss power minimization. *IEEE Transactions on Power System*, 20(2):859–866, 2005.
- [168] Stefan Janson and Martin Middendorf. A hierarchical particle swarm optimizer and its adaptive variant. *IEEE Transactions on System Man Cybernetics Part B Cybernetics*, 35(6):1272–1282, 2005.
- [169] C Jiang and C Wang. Improved evolutionary programming with dynamic mutation and metropolis criteria for multi-objective reactive power optimisation. *IEE Proceeding Generation Transmission and Distribution*, 152(2):291–294, 2005.
- [170] Kwang Y Lee and Jong-Bae Park. Application of particle swarm optimization to economic dispatch problem: advantages and disadvantages. In *IEEE Power Systems Conference and Exposition*, pages 188–192, 2005.
- [171] Xiaodong Li. Efficient differential evolution using speciation for multimodal function optimization. In *7th annual conference on Genetic and evolutionary computation*, pages 873–880, 2005.
- [172] Derong Liu and Ying Cai. Taguchi method for solving the economic dispatch problem with nonsmooth cost functions. *IEEE Transactions on Power System*, 20(4):2006–2014, 2005.
- [173] Junhong Liu and Jouni Lampinen. A fuzzy adaptive differential evolution algorithm. *Soft Computing*, 9(6):448–462, 2005.
- [174] Mahamed GH Omran, Ayed Salman, and Andries P Engelbrecht. Self-adaptive differential evolution. In *International Conference on Computational and Information Science*, pages 192–199. Springer, 2005.
- [175] Gary Pampara, Nelis Franken, and Andries Petrus Engelbrecht. Combining particle swarm optimisation with angle modulation to solve binary problems. In *IEEE Congress on Evolutionary Computation*, volume 1, pages 89–96, 2005.
- [176] Jong-Bae Park, Ki-Song Lee, Joong-Rin Shin, and Kwang Y Lee. A particle swarm optimization for economic dispatch with nonsmooth cost functions. *IEEE Transactions on Power System*, 20(1):34–42, 2005.
- [177] A Pereira-Neto, C Unsuhay, and OR Saavedra. Efficient evolutionary strategy optimisation procedure to solve the nonconvex economic dispatch problem with generator constraints. *IEE Proceedings-Generation, Transmission and Distribution*, 152(5):653–660, 2005.
- [178] Riccardo Poli, Cecilia Di Chio, and William B Langdon. Exploring extended particle swarms: a genetic programming approach. In *7th annual conference on Genetic and evolutionary computation*, pages 169–176, 2005.
- [179] Carlo R Raquel and Prospero C Naval Jr. An effective use of crowding distance in multiobjective particle swarm optimization. In *7th annual conference on Genetic and evolutionary computation*, pages 257–264, 2005.
- [180] Tea Robič and Bogdan Filipič. Demo: Differential evolution for multiobjective optimization. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 520–533. Springer, 2005.
- [181] P Somasundaram and K Kuppusamy. Application of evolutionary programming to security constrained economic dispatch. *International Journal of Electrical Power and Energy System*, 27(5):343–351, 2005.
- [182] P Somasundaram, R Lakshmiramanan, and K Kuppusamy. Hybrid algorithm based on ep and lp for security constrained economic dispatch problem. *Electrical Power System Research*, 76(1):77–85, 2005.
- [183] T Victoire and A Ebenezer Jeyakumar. Reserve constrained dynamic dispatch of units with valve-point effects. *IEEE Transactions on Power Systems*, 20(3):1273–1282, 2005.
-

-
- [184] MR AlRashidi and ME El-Hawary. Economic dispatch with environmental considerations using particle swarm optimization. In *Large Engineering Systems Conference Power Engineering*, pages 41–46, 2006.
- [185] MR Alrashidi and ME El-Hawary. A survey of particle swarm optimization applications in power system operations. *Electric Power Components and Systems*, 34(12):1349–1357, 2006.
- [186] Ramesh C Bansal. Overview and literature survey of artificial neural networks applications to power systems (1992-2004). *Journal of Institution of Engineers*, 86(1):282–296, 2006.
- [187] Bahriye Basturk and Dervis Karaboga. An artificial bee colony (ABC) algorithm for numeric function optimization. In *IEEE swarm Intelligence Symposium*, volume 8, pages 687–697, 2006.
- [188] YS Brar, JS Dhillon, and DP Kothari. Multiobjective load dispatch based on genetic-fuzzy technique. In *IEEE Power Systems Conference and Exposition*, pages 931–937, 2006.
- [189] Janez Brest, Sao Greiner, Borko Boskovic, Marjan Mernik, and Viljem Zumer. Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. *IEEE Transactions on Evolutionary Computing*, 10(6):646–657, 2006.
- [190] CH Chen and SN Yeh. Particle swarm optimization for economic power dispatch with valve-point effects. In *Transmission and Distribution Conference and Exposition: Latin America*, pages 1–5, 2006.
- [191] Ji-Pyng Chiou, Chung-Fu Chang, and Ching-Tzong Su. Capacitor placement in large-scale distribution systems using variable scaling hybrid differential evolution. *International Journal of Electrical Power and Energy System*, 28(10):739–745, 2006.
- [192] Shu-Chuan Chu, Pei-Wei Tsai, and Jeng-Shyang Pan. Cat swarm optimization. In *Pacific International Conference on Artificial Intelligence*, pages 854–858. Springer, 2006.
- [193] Ld S Coelho and Viviana Cocco Mariani. Combining of chaotic differential evolution and quadratic programming for economic dispatch optimization with valve-point effect. *IEEE Transactions on power systems*, 21(2):989–996, 2006.
- [194] Leandro dos Santos Coelho and Viviana Cocco Mariani. Combining of chaotic differential evolution and quadratic programming for economic dispatch optimization with valve-point effect. *IEEE Transactions on Power System*, 21(2):989, 2006.
- [195] Leandro dos Santos Coelho and Viviana Cocco Mariani. Particle swarm optimization with quasi-newton local search for solving economic dispatch problem. In *IEEE International Conference Systems, Man and Cybernetics*, volume 4, pages 3109–3113, 2006.
- [196] Muzaffar Eusuff, Kevin Lansey, and Fayzul Pasha. Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization. *Engineering Optimization*, 38(2):129–154, 2006.
- [197] Mitsuharu Higashitani, Atsushi Ishigame, and Keiichiro Yasuda. Particle swarm optimization considering the concept of predator-prey behavior. In *IEEE Conference on Evolutionary Computing*, pages 434–437, 2006.
- [198] DN Jeyakumar, T Jayabarathi, and T Raghunathan. Particle swarm optimization for various types of economic dispatch problems. *International Journal of Electrical Power and Energy System*, 28(1):36–42, 2006.
- [199] P Kaelo and MM Ali. A numerical study of some modified differential evolution algorithms. *European Journal of Operational Research*, 169(3):1176–1184, 2006.
- [200] S Muralidharan, K Srikrishna, and S Subramanian. Emission constrained economic dispatcha new recursive approach. *Electric Power Components and Systems*, 34(3):343–353, 2006.
- [201] Jorge Nocedal and Stephen J Wright. Conjugate gradient methods. *Numerical Optimization*, pages 101–134, 2006.
- [202] Yew-Soon Ong, Meng-Hiot Lim, Ning Zhu, and Kok-Wai Wong. Classification of adaptive memetic algorithms: a comparative study. *IEEE Transactions on System Man Cybernetics Part B Cybernetics*, 36(1):141–152, 2006.
- [203] W Ongsakul and T Tantimaporn. Optimal power flow by improved evolutionary programming. *Electric Power Components and Systems*, 34(1):79–95, 2006.
-

-
- [204] Godfrey Onwubolu and Donald Davendra. Scheduling flow shops using differential evolution algorithm. *European Journal of Operational Research*, 171(2):674–692, 2006.
- [205] CK Panigrahi, PK Chattopadhyay, RN Chakrabarti, and M Basu. Simulated annealing technique for dynamic economic dispatch. *Electric Power Components and Systems*, 34(5):577–586, 2006.
- [206] Daniel Parrott and Xiaodong Li. Locating and tracking multiple dynamic optima by a particle swarm model using speciation. *IEEE Transactions on Evolutionary Computation*, 10(4):440–458, 2006.
- [207] Lakhwinder Singh and Jaspreet Singh Dhillon. Surrogate worth trade-off method for economic-emission dispatch. In *Power Systems Conference: Advanced Metering, Protection, Control*, pages 230–235, 2006.
- [208] KS Swarup and P Rohit Kumar. A new evolutionary computation technique for economic dispatch with security constraints. *International Journal of Electrical Power and Energy System*, 28(4):273–283, 2006.
- [209] TO Ting, MVC Rao, and CK Loo. A novel approach for unit commitment problem via an effective hybrid particle swarm optimization. *IEEE Transactions on Power System*, 21(1):411–418, 2006.
- [210] Frans Van den Bergh and Andries Petrus Engelbrecht. A study of particle swarm optimization particle trajectories. *Information Science*, 176(8):937–971, 2006.
- [211] MA Abido. Multiobjective particle swarm for environmental/economic dispatch problem. In *International Power Engineering Conference*, pages 1385–1390, 2007.
- [212] Fawaz S Al-Anzi and Ali Allahverdi. A self-adaptive differential evolution heuristic for two-stage assembly scheduling problem to minimize maximum lateness with setup times. *European Journal of Operational Research*, 182(1):80–94, 2007.
- [213] MR AlRashidi and ME El-Hawary. Hybrid particle swarm optimization approach for solving the discrete opf problem considering the valve loading effects. *IEEE Transactions on Power System*, 22(4):2030–2038, 2007.
- [214] Arijit Bisuias, Sambarta Dasgupta, and Swagatam Dasf Ajith Abraham. A synergy of differential evolution and bacterial foraging optimization for global optimization. *Neural Network World*, 6(07):607–626, 2007.
- [215] Daniel Bratton and James Kennedy. Defining a standard for particle swarm optimization. In *IEEE swarm Intelligence Symposium*, pages 120–127, 2007.
- [216] Jiejun Cai, Xiaoqian Ma, Lixiang Li, and Peng Haipeng. Chaotic particle swarm optimization for economic dispatch considering the generator constraints. *Energy Conversion and Management*, 48(2):645–653, 2007.
- [217] Donald Davendra and Godfrey Onwubolu. Enhanced differential evolution hybrid scatter search for discrete optimization. In *IEEE Congress on Evolutionary Computation*, pages 1156–1162, 2007.
- [218] Leandro dos Santos Coelho and Viviana Cocco Mariani. Improved differential evolution algorithms for handling economic dispatch optimization with generator constraints. *Energy Conversion and Management*, 48(5):1631–1639, 2007.
- [219] Shu-Kai S Fan and Ju-Ming Chang. A modified particle swarm optimizer using an adaptive dynamic weight scheme. In *Digital Human Modeling*, pages 56–65. Springer, 2007.
- [220] Yichao He, Xizhao Wang, and Yingzhan Kou. A binary differential evolution algorithm with hybrid encoding. *Computer Research and Development*, 44(9):1476–1484, 2007.
- [221] Chao-Ming Huang and Fu-Lu Wang. An rbf network with ols and epso algorithms for real-time power dispatch. *IEEE Transactions on Power System*, 22(1):96–104, 2007.
- [222] P Kaelo and MM Ali. Differential evolution algorithms using hybrid mutation. *Computational Optimization and Applications*, 37(2):231–246, 2007.
- [223] Dervis Karaboga and Bahriye Basturk. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of Global Optimization*, 39(3):459–471, 2007.
-

-
- [224] James Kennedy. Some issues and practices for particle swarms. In *IEEE Swarm Intelligence Symposium*, 2007.
- [225] Bo Liu, Ling Wang, and Yihui Jin. Advances in differential evolution. *Control Decisions*, 22(7):721, 2007.
- [226] C Christopher Asir Rajan and MR Mohan. An evolutionary programming based simulated annealing method for solving the unit commitment problem. *International Journal of Electrical Power and Energy System*, 29(7):540–550, 2007.
- [227] A Immanuel Selvakumar and K Thanushkodi. A new particle swarm optimization solution to nonconvex economic dispatch problems. *IEEE Transactions on Power System*, 22(1):42–51, 2007.
- [228] PS Shelokar, Patrick Siarry, Valadi K Jayaraman, and Bhaskar D Kulkarni. Particle swarm and ant colony algorithms hybridized for improved continuous optimization. *Applied mathematics and computation*, 188(1):129–142, 2007.
- [229] Lakhwinder Singh and JS Dhillon. Fuzzy satisfying interactive multiobjective thermal power dispatch: Swt approach. *Journal of System Science System Engineers*, 16(1):88–106, 2007.
- [230] M Sudhakaran, P Ajay-D-Vimalraj, and TG Palanivelu. Ga and pso culled hybrid technique for economic dispatch problem with prohibited operating zones. *Journal of Zhejiang University SCIENCE A*, 8(6):896–903, 2007.
- [231] M Fatih Tasgetiren, Quan-Ke Pan, Ponnuthurai Nagaratnam Suganthan, and Yun-Chia Liang. A discrete differential evolution algorithm for the no-wait flow shop scheduling problem with total flowtime criterion. In *IEEE Symposium on Computing Intelligence in Scheduling*, pages 251–258, 2007.
- [232] Lingfeng Wang and Chanan Singh. Environmental/economic power dispatch using a fuzzified multi-objective particle swarm optimization algorithm. *Electrical Power System Research*, 77(12):1654–1664, 2007.
- [233] Tankut Yalcinoz and Onur Köksoy. A multiobjective optimization method to environmental economic dispatch. *International Journal of Electrical Power and Energy System*, 29(1):42–50, 2007.
- [234] Bo Yang, Yunping Chen, and Zunlian Zhao. Survey on applications of particle swarm optimization in electric power systems. In *IEEE International Conference on Control and Automation*, pages 481–486, 2007.
- [235] Zhenyu Yang, Xin Yao, and Jingsong He. Making a difference to differential evolution. In *Advances in meta heuristics for hard optimization*, pages 397–414. Springer, 2007.
- [236] MR Alrashidi and ME El-Hawary. Impact of loading conditions on the emission-economic dispatch. *World Academy of Science, Engineering and Technology*, 29:148–151, 2008.
- [237] H Altun and T Yalcinoz. Implementing soft computing techniques to solve economic dispatch problem in power systems. *Expert System with Applications*, 35(4):1668–1678, 2008.
- [238] R Balamurugan and S Subramanian. Hybrid integer coded differential evolution–dynamic programming approach for economic load dispatch with multiple fuel options. *Energy Conversion and Management*, 49(4):608–614, 2008.
- [239] Alec Banks, Jonathan Vincent, and Chukwudi Anyakoha. A review of particle swarm optimization. part ii: hybridization, combinatorial, multi criteria and constrained optimization, and indicative applications. *Natural Computing*, 7(1):109–124, 2008.
- [240] M Basu. Optimal power flow with facts devices using differential evolution. *International Journal of Electrical Power and Energy System*, 30(2):150–156, 2008.
- [241] Xie Bin. Web mining based on chaotic social evolutionary programming algorithm. *Journal of System Engineers and Electronics*, 19(6):1272–1276, 2008.
- [242] HR Cai, CY Chung, and KP Wong. Application of differential evolution algorithm for transient stability constrained optimal power flow. *IEEE Transactions on Power System*, 23(2):719–728, 2008.
- [243] Krishna Teerth Chaturvedi, Manjaree Pandit, and Laxmi Srivastava. Self-organizing hierarchical particle swarm optimization for nonconvex economic dispatch. *IEEE Transactions on Power System*, 23(3):1079–1087, 2008.
-

-
- [244] Zhihua Cui, Xingjuan Cai, Jianchao Zeng, and Guoji Sun. Particle swarm optimization with fss and rws for high dimensional functions. *Applied Mathematics and Computation*, 205(1):98–108, 2008.
- [245] Yamille Del Valle, Ganesh Kumar Venayagamoorthy, Salman Mohagheghi, Jean-Carlos Hernandez, and Ronald G Harley. Particle swarm optimization: basic concepts, variants and applications in power systems. *IEEE Transactions on Evolutionary Computing*, 12(2):171–195, 2008.
- [246] Leandro dos Santos Coelho and Chu-Sheng Lee. Solving economic load dispatch problems in power systems using chaotic and gaussian particle swarm optimization approaches. *International Journal of Electrical Power and Energy System*, 30(5):297–307, 2008.
- [247] Dakuo He, Fuli Wang, and Zhizhong Mao. A hybrid genetic algorithm approach based on differential evolution for economic dispatch with valve-point effect. *International Journal of Electrical Power and Energy System*, 30(1):31–38, 2008.
- [248] Zhang Jie and Ju Hui. A novel evolutionary programming for adaptive filter based on learning mutation. In *International Conference on Communication, Circuits and System*, pages 851–853, 2008.
- [249] Wu Jiekang, Zhu Jianquan, Chen Guotong, and Zhang Hongliang. A hybrid method for optimal scheduling of short-term electric power generation of cascaded hydroelectric plants based on particle swarm optimization and chance-constrained programming. *IEEE Transactions on Power System*, 23(4):1570–1579, 2008.
- [250] L Lakshminarasimman and S Subramanian. A modified hybrid differential evolution for short-term scheduling of hydrothermal power systems with cascaded reservoirs. *Energy Conversion and Management*, 49(10):2513–2521, 2008.
- [251] Dioubate Mamady and Guanzheng Tan. Immune evolutionary programming-based locomotion control of autonomous mobile robot. In *7th World Congress on Intelligent Control and Automation*, pages 8990–8996, 2008.
- [252] Efrén Mezura-Montes, Margarita Reyes-Sierra, and Carlos A Coello Coello. Multi-objective optimization using differential evolution: a survey of the state-of-the-art. In *Advances in differential evolution*, pages 173–196. Springer, 2008.
- [253] Kaisa Miettinen. Introduction to multiobjective optimization: Noninteractive approaches. In *Multiobjective Optimization*, pages 1–26. Springer, 2008.
- [254] Nasimul Noman and Hitoshi Iba. Differential evolution for economic load dispatch problems. *Electrical Power System Research*, 78(8):1322–1331, 2008.
- [255] BK Panigrahi and V Ravikumar Pandi. Bacterial foraging optimisation: Nelder-mead hybrid algorithm for economic load dispatch. *IET Generation Transmission Distribution*, 2(4):556–565, 2008.
- [256] BK Panigrahi, V Ravikumar Pandi, and Sanjoy Das. Adaptive particle swarm optimization approach for static and dynamic economic load dispatch. *Energy Conversion and Management*, 49(6):1407–1415, 2008.
- [257] Millie Pant, Radha Thangaraj, and Ajith Abraham. Particle swarm optimization using adaptive mutation. In *19th International Workshop on Database and Expert Systems Applications*, pages 519–523, 2008.
- [258] TS Prasanna and P Somasundaram. Fuzzy mutated evolutionary programming based algorithm for combined economic and emission dispatch. In *IEEE Region 10th Conference (TENCON)*, pages 1–5, 2008.
- [259] Bin Qian, Ling Wang, Rong Hu, Wan-Liang Wang, De-Xian Huang, and Xiong Wang. A hybrid differential evolution method for permutation flow-shop scheduling. *International Journal of Advance Manufacture Technology*, 38(7-8):757–777, 2008.
- [260] Shahryar Rahnamayan, Hamid R Tizhoosh, and Magdy MA Salama. Opposition-based differential evolution. *IEEE Transactions on Evolutionary Computing*, 12(1):64–79, 2008.
- [261] Samir Sayah and Khaled Zehar. Modified differential evolution algorithm for optimal power flow with non-smooth cost functions. *Energy Conversion and Management*, 49(11):3036–3042, 2008.
-

-
- [262] A Immanuel Selvakumar and K Thanushkodi. Anti-predatory particle swarm optimization: solution to nonconvex economic dispatch problems. *Electrical Power System Research*, 78(1):2–10, 2008.
- [263] Dan Simon. Biogeography-based optimization. *IEEE Transactions on Evolutionary Computing*, 12(6):702–713, 2008.
- [264] Renato Tinós and Shengxiang Yang. Evolutionary programming with q-gaussian mutation for dynamic optimization problems. In *IEEE World Congress on Computational Intelligence*, pages 1823–1830, 2008.
- [265] P Venkatesh and Kwang Y Lee. Multi-objective evolutionary programming for economic emission dispatch problem. In *IEEE Power and Energy Society General Meeting-Conversion*, pages 1–8, 2008.
- [266] Guang Ya Yang, Zhao Yang Dong, and Kit Po Wong. A modified differential evolution algorithm with fitness sharing for power system planning. *IEEE Transactions on Power System*, 23(2):514–522, 2008.
- [267] QW Yang, Liang Cai, and YC Xue. A survey of differential evolution algorithms. *International Journal of Pattern Recognition Artificial Intelligence*, 21(4):506–513, 2008.
- [268] MA Abido. Multiobjective particle swarm optimization for environmental/economic dispatch problem. *Electrical Power System Research*, 79(7):1105–1113, 2009.
- [269] JA Adeyemo and FAO Otieno. Multi-objective differential evolution algorithm for solving engineering problems. *Journal of Applied Sciences*, 9(20):3652–3661, 2009.
- [270] Mohammed R AlRashidi and Mohamed E El-Hawary. A survey of particle swarm optimization applications in electric power systems. *IEEE Transactions on Evolutionary Computing*, 13(4):913–918, 2009.
- [271] Jiejun Cai, Xiaoqian Ma, Qiong Li, Lixiang Li, and Haipeng Peng. A multi-objective chaotic particle swarm optimization for environmental/economic dispatch. *Energy Conversion and Management*, 50(5):1318–1325, 2009.
- [272] A Chatterjee, SP Ghoshal, and V Mukherjee. A comparative study of single input and dual input power system stabilizer by hybrid evolutionary programming. In *World Congress on Nature and Biologically Inspired Computing*, pages 1047–1052, 2009.
- [273] Gonggui Chen. Combined economic emission dispatch using sfla. In *International Conference on Information Engineering and Computer Science*, pages 1–4, 2009.
- [274] J-P Chiou. A variable scaling hybrid differential evolution for solving large-scale power dispatch problems. *IET Generation Transmission Distribution*, 3(2):154–163, 2009.
- [275] Swagatam Das, Ajith Abraham, Uday K Chakraborty, and Amit Konar. Differential evolution using a neighborhood-based mutation operator. *IEEE Transactions on Evolutionary Computation*, 13(3):526–553, 2009.
- [276] Marco A Montes De Oca, Thomas Stützle, Mauro Birattari, and Marco Dorigo. Frankenstein’s pso: a composite particle swarm optimization algorithm. *IEEE Transactions on Evolutionary Computation*, 13(5):1120–1132, 2009.
- [277] JS Dhillon and DP Kothari. Economic-emission load dispatch using binary successive approximation-based evolutionary search. *IET Generation Transmission Distribution*, 3(1):1–16, 2009.
- [278] Leandro dos Santos Coelho and Viviana Cocco Mariani. Chaotic artificial immune approach applied to economic dispatch of electric energy using thermal units. *Chaos, Solitons and Fractals*, 40(5):2376–2383, 2009.
- [279] Leandro dos Santos Coelho, João Guilherme Sauer, and Marcelo Rudek. Differential evolution optimization combined with chaotic sequences for image contrast enhancement. *Chaos Solitons Fractals*, 42(1):522–529, 2009.
- [280] M Fesanghary and MM Ardehali. A novel meta-heuristic optimization methodology for solving various types of economic dispatch problem. *Energy*, 34(6):757–766, 2009.
-

-
- [281] Vicky Ling Huang, Shuguang Z Zhao, Rammohan Mallipeddi, and Ponnuthurai N Suganthan. Multi-objective optimization using self-adaptive differential evolution algorithm. In *IEEE Congress on Evolutionary Computation*, pages 190–194, 2009.
- [282] Antonio López Jaimes, Saúl Zapotecas Martínez, and Carlos A Coello Coello. An introduction to multiobjective optimization techniques. *Optimization in Polymer Processing*, pages 29–57, 2009.
- [283] Renato A Krohling and Eduardo Mendel. Bare bones particle swarm optimization with gaussian or cauchy jumps. In *IEEE Congress on Evolutionary Computation*, pages 3285–3291, 2009.
- [284] Mariano Luque, Kaisa Miettinen, Petri Eskelinen, and Francisco Ruiz. Incorporating preference information in interactive reference point methods for multiobjective optimization. *Omega*, 37(2):450–462, 2009.
- [285] Jose L Ceciliano Meza, Mehmet Bayram Yildirim, and Abu SM Masud. A multiobjective evolutionary programming algorithm and its applications to power generation expansion planning. *IEEE Transactions on System Man Cybernetics-Part A: Systems and Humans*, 39(5):1086–1096, 2009.
- [286] P Murugan, S Kannan, and S Baskar. Application of nsga-ii algorithm to single-objective transmission constrained generation expansion planning. *IEEE Transactions on Power System*, 24(4):1790–1797, 2009.
- [287] Ferrante Neri and Ville Tirronen. Scale factor local search in differential evolution. *Memetic Computing*, 1(2):153–171, 2009.
- [288] MS Osman, MA Abo-Sinna, and AA Mousa. An ϵ -dominance-based multiobjective genetic algorithm for economic emission load dispatch optimization problem. *Electrical Power System Research*, 79(11):1561–1567, 2009.
- [289] K Premalatha and AM Natarajan. Hybrid pso and ga for global maximization. *International Journal of Open Problems and Computational Mathematics*, 2(4):597–608, 2009.
- [290] A Kai Qin, Vicky Ling Huang, and Ponnuthurai N Suganthan. Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Transactions on Evolutionary Computing*, 13(2):398–417, 2009.
- [291] Bo-Yang Qu and Ponnuthurai N Suganthan. Multi-objective evolutionary programming without non-domination sorting is up to twenty times faster. In *IEEE Congress on Evolutionary Computation*, pages 2934–2939, 2009.
- [292] Esmat Rashedi, Hossein Nezamabadi-Pour, and Saeid Saryazdi. Gsa: a gravitational search algorithm. *Information Science*, 179(13):2232–2248, 2009.
- [293] Davoud Sedighizadeh and Ellips Masehian. Particle swarm optimization methods, taxonomy and applications. *International Journal of Computational Engineers*, 1(5):486, 2009.
- [294] A Immanuel Selvakumar and K Thanushkodi. Optimization using civilized swarm: solution to economic dispatch with multiple minima. *Electrical Power System Research*, 79(1):8–16, 2009.
- [295] S Chitra Selvi, RP Kumudini Devi, and C Christofer Asir Rajan. Hybrid evolutionary programming approach to multi-area unit commitment with import and export constraints. *Proceeding of International Journal on Recent Trends in Engineering*, 1(3), 2009.
- [296] Huang Shao-Rong. Survey of particle swarm optimization algorithm. *Computer Engineers and Design*, 30:1977–1980, 2009.
- [297] Lakhwinder Singh and JS Dhillon. Cardinal priority ranking based decision making for economic-emission dispatch problem. *International Journal of Engineering, Science and Technology*, 1(1):272–282, 2009.
- [298] KK Swarnkar, S Wadhvani, and AK Wadhvani. Optimal power flow of large distribution system solution for combined economic emission dispatch problem using partial swarm optimization. In *International Conference on Power Systems*, pages 1–5, 2009.
- [299] Nga Sing Teng, Jason Teo, and Mohd Hanafi A Hijazi. Self-adaptive population sizing for a tune-free differential evolution. *Soft Computing*, 13(7):709–724, 2009.
- [300] Zeld B Zabinsky. Random search algorithms. *Wiley Encyclopedia on Operation Research Management Science*, 2009.
-

-
- [301] Erwie Zahara and Yi-Tung Kao. Hybrid nelder–mead simplex search and particle swarm optimization for constrained engineering design problems. *Expert System with Applications*, 36(2):3880–3886, 2009.
- [302] Jingqiao Zhang and Arthur C Sanderson. Jade: adaptive differential evolution with optional external archive. *IEEE Transactions on evolutionary computation*, 13(5):945–958, 2009.
- [303] Qinghai Bai. Analysis of particle swarm optimization algorithm. *Computer and information science*, 3(1):180, 2010.
- [304] Javad Behnamian and SMT Fatemi Ghomi. Development of a pso–sa hybrid metaheuristic for a new comprehensive regression model to time-series forecasting. *Expert System with Applications*, 37(2):974–984, 2010.
- [305] Aniruddha Bhattacharya and Pranab Kumar Chattopadhyay. Hybrid differential evolution with biogeography-based optimization for solution of economic load dispatch. *IEEE Transactions on Power System*, 25(4):1955–1964, 2010.
- [306] Aniruddha Bhattacharya and Pranab Kumar Chattopadhyay. Solving complex economic load dispatch problems using biogeography-based optimization. *Expert Systems with Applications*, 37(5):3605–3615, 2010.
- [307] Jiejun Cai, Xiaoqian Ma, Qiong Li, Lixiang Li, and Haipeng Peng. A multi-objective chaotic ant swarm optimization for environmental/economic dispatch. *International Journal of International journal of Electric Power and Energy System*, 32(5):337–344, 2010.
- [308] CY Chung, CH Liang, KP Wong, and XZ Duan. Hybrid algorithm of differential evolution and evolutionary programming for optimal reactive power flow. *IET Generation Transmission and Distribution*, 4(1):84–93, 2010.
- [309] Leandro dos Santos Coelho and Viviana Cocco Mariani. An efficient cultural self-organizing migrating strategy for economic dispatch optimization with valve-point effect. *Energy Conversion and Management*, 51(12):2580–2587, 2010.
- [310] AA Abou El Ela, MA Abido, and S Rl Spea. Differential evolution algorithm for emission constrained economic power dispatch problem. *Electrical Power System Research*, 80(10):1286–1292, 2010.
- [311] AA Abou El Ela, MA Abido, and SR Spea. Optimal power flow using differential evolution algorithm. *Electrical Power System Research*, 80(7):878–885, 2010.
- [312] PK Hota, AK Barisal, and R Chakrabarti. Economic emission load dispatch through fuzzy based bacterial foraging algorithm. *International Journal of Electrical Power and Energy System*, 32(7):794–803, 2010.
- [313] T Warren Liao. Two hybrid differential evolution algorithms for engineering design optimization. *Applied Soft Computing*, 10(4):1188–1199, 2010.
- [314] Haiyan Lu, Pichet Sriyanyong, Yong Hua Song, and Tharam Dillon. Experimental study of a new hybrid pso with mutation for economic dispatch with non-smooth cost function. *International Journal of Electrical Power and Energy System*, 32(9):921–935, 2010.
- [315] Belkacem Mahdad, K Srairi, and T Bouktir. Improved parallel PSO solution to economic dispatch with practical generator constraints. In *15th IEEE Mediterranean Electro-technical Conference*, pages 314–319, 2010.
- [316] Tahir Nadeem Malik, Azzam ul Asar, Mudasser F Wyne, and Shakil Akhtar. A new hybrid approach for the solution of nonconvex economic dispatch problem with valve-point effects. *Electrical Power System Research*, 80(9):1128–1136, 2010.
- [317] Ke Meng, Hong Gang Wang, ZhaoYang Dong, and Kit Po Wong. Quantum-inspired particle swarm optimization for valve-point economic load dispatch. *IEEE Transactions on Power System*, 25(1):215–222, 2010.
- [318] Ferrante Neri and Ville Tirronen. Recent advances in differential evolution: a survey and experimental analysis. *Artificial Intelligence*, 33(1-2):61–106, 2010.
- [319] Taher Niknam. A new fuzzy adaptive hybrid particle swarm optimization algorithm for non-linear, non-smooth and non-convex economic dispatch problem. *Applied Energy*, 87(1):327–339, 2010.
-

-
- [320] BK Panigrahi, V Ravikumar Pandi, Sanjoy Das, and Swagatam Das. Multiobjective fuzzy dominance based bacterial foraging algorithm to solve economic emission dispatch problem. *Energy*, 35(12):4761–4770, 2010.
- [321] Jong-Bae Park, Yun-Won Jeong, Joong-Rin Shin, and Kwang Y Lee. An improved particle swarm optimization for nonconvex economic dispatch problems. *IEEE Transactions on Power Systems*, 25(1):156–166, 2010.
- [322] M Pilát. Evolutionary multiobjective optimization: A short survey of the state-of-the-art. *Proceeding the Contributed Papers Part I-Mathematics and Computer Sciences*, pages 1–4, 2010.
- [323] S Ali Pourmousavi, M Hashem Nehrir, Christopher M Colson, and Caisheng Wang. Real-time energy management of a stand-alone hybrid wind-microturbine energy system using particle swarm optimization. *IEEE Transactions on Sustainable Energy*, 1(3):193–201, 2010.
- [324] Bo-Yang Qu and Ponnuthurai Nagarathnam Suganthan. Novel multimodal problems and differential evolution with ensemble of restricted tournament selection. In *IEEE Congress on Evolutionary Computation*, pages 1–7, 2010.
- [325] C Christoper Asir Rajan. A solution to the economic dispatch using ep based sa algorithm on large scale power system. *International Journal of Electrical Power and Energy System*, 32(6):583–591, 2010.
- [326] Hamed Taheri, Zainal Salam, Kashif Ishaque, et al. A novel maximum power point tracking control of photovoltaic system under partial and rapidly fluctuating shadow conditions using differential evolution. In *IEEE Symposium on Industrial Electronics and Applications*, pages 82–87, 2010.
- [327] V Vahidinasab and S Jadid. Joint economic and emission dispatch in energy markets: a multiobjective mathematical programming approach. *Energy*, 35(3):1497–1504, 2010.
- [328] LH Wu, YN Wang, XF Yuan, and SW Zhou. Environmental/economic power dispatch problem using multi-objective differential evolution algorithm. *Electrical Power System Research*, 80(9):1171–1181, 2010.
- [329] Johannes Bader and Eckart Zitzler. Hype: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary Computation*, 19(1):45–76, 2011.
- [330] M Basu. Economic environmental dispatch using multi-objective differential evolution. *Applied Soft Computing*, 11(2):2845–2853, 2011.
- [331] Aniruddha Bhattacharya and Pranab Kumar Chattopadhyay. Solving economic emission load dispatch problems using hybrid differential evolution. *Applied Soft Computing*, 11(2):2526–2537, 2011.
- [332] Leticia Cecilia Cagnina, Susana Cecilia Esquivel, and Carlos A Coello Coello. A fast particle swarm algorithm for solving smooth and non-smooth economic dispatch problems. *Engineers Optimization*, 43(5):485–505, 2011.
- [333] Shyi-Ming Chen and Chih-Yao Chien. Solving the traveling salesman problem based on the genetic simulated annealing ant colony system with particle swarm optimization techniques. *Expert System with Applications*, 38(12):14439–14450, 2011.
- [334] Swagatam Das and Ponnuthurai Nagarathnam Suganthan. Differential evolution: a survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation*, 15(1):4–31, 2011.
- [335] S Dhanalakshmi, S Kannan, K Mahadevan, and S Baskar. Application of modified NSGA-II algorithm to combined economic and emission dispatch problem. *International Journal of Electrical Power and Energy System*, 33(4):992–1002, 2011.
- [336] Gaurav Prasad Dixit, Hari Mohan Dubey, Manjaree Pandit, and BK Panigrahi. Artificial bee colony optimization for combined economic load and emission dispatch. In *International Conference Sustainable Energy and Intelligent Systems*, pages 340–345, 2011.
- [337] Nagarjuna Duvvuru and KS Swarup. A hybrid interior point assisted differential evolution algorithm for economic dispatch. *IEEE Transactions on Power System*, 26(2):541–549, 2011.
- [338] AA Abou El Ela, MA Abido, and SR Spea. Differential evolution algorithm for optimal reactive power dispatch. *Electrical Power System Research*, 81(2):458–464, 2011.
-

-
- [339] Weifeng Gao and Sanyang Liu. Improved artificial bee colony algorithm for global optimization. *Information Processing Letters*, 111(17):871–882, 2011.
- [340] Dakuo He, Gang Dong, Fuli Wang, and Zhizhong Mao. Optimization of dynamic economic dispatch with valve-point effect using chaotic sequence based differential evolution algorithms. *Energy Conversion and Management*, 52(2):1026–1032, 2011.
- [341] S Jeyadevi, S Baskar, CK Babulal, and M Willjuice Iruthayarajan. Solving multiobjective optimal reactive power dispatch using modified nsga-ii. *International journal of Electric Power and Energy System*, 33(2):219–228, 2011.
- [342] Gopala Krishnan and A Krishnan. Study on techniques for combined economic and emission dispatch. *Global Journal of Research Electrical and Electronic Engineers*, 11(5), 2011.
- [343] Rajesh Kumar, Devendra Sharma, and Abhinav Sadu. A hybrid multi-agent based particle swarm optimization algorithm for economic power dispatch. *International Journal of Electrical Power and Energy System*, 33(1):115–123, 2011.
- [344] Youlin Lu, Jianzhong Zhou, Hui Qin, Ying Wang, and Yongchuan Zhang. Chaotic differential evolution methods for dynamic economic dispatch with valve-point effects. *Engineering Applications of Artificial Intelligence*, 24(2):378–387, 2011.
- [345] Youlin Lu, Jianzhong Zhou, Hui Qin, Ying Wang, and Yongchuan Zhang. A hybrid multi-objective cultural algorithm for short-term environmental/economic hydrothermal scheduling. *Energy Conversion and Management*, 52(5):2121–2134, 2011.
- [346] KK Mishra, Shailesh Tiwari, and AK Misra. A bio inspired algorithm for solving optimization problems. In *2nd International Conference on Computer and Communication Technology*, pages 653–659, 2011.
- [347] Taher Niknam, Hasan Doagou Mojarrad, and Hamed Zeinoddini Meymand. A novel hybrid particle swarm optimization for economic dispatch with valve-point loading effects. *Energy Conversion and Management*, 52(4):1800–1809, 2011.
- [348] V Ravikumar Pandi, BK Panigrahi, Ramesh C Bansal, Swagatam Das, and Ankita Mohapatra. Economic load dispatch using hybrid swarm intelligence based harmony search algorithm. *Electrical Power Components Systems*, 39(8):751–767, 2011.
- [349] Husam I Shaheen, Ghamgeen I Rashed, and SJ Cheng. Optimal location and parameter setting of upfc for enhancing power system security based on differential evolution algorithm. *International Journal of Electrical Power and Energy System*, 33(1):94–105, 2011.
- [350] Yuhui Shi. Brain storm optimization algorithm. In *International Conference on Swarm Intelligence*, pages 303–309. Springer, 2011.
- [351] S Sivasubramani and KS Swarup. Environmental/economic dispatch using multi-objective harmony search algorithm. *Electrical Power System Research*, 81(9):1778–1785, 2011.
- [352] Radha Thangaraj, Millie Pant, Ajith Abraham, and Pascal Bouvry. Particle swarm optimization: hybridization perspectives and experimental illustrations. *Applied Mathematics and Computation*, 217(12):5208–5226, 2011.
- [353] Musrrat Ali, Patrick Siarry, and Millie Pant. An efficient differential evolution based algorithm for solving multi-objective optimization problems. *European Journal of Operational Research*, 217(2):404–416, 2012.
- [354] Jiejun Cai, Qiong Li, Lixiang Li, Haipeng Peng, and Yixian Yang. A hybrid cpso–sqp method for economic dispatch considering the valve-point effects. *Energy Conversion and Management*, 53(1):175–181, 2012.
- [355] Irina Ciornei and Elias Kyriakides. A ga-api solution for the economic dispatch of generation in power system operation. *IEEE Transactions on Power System*, 27(1):233–242, 2012.
- [356] Marco Aurelio Costa e Silva, Leandro dos S Coelho, and Luiz Lebensztajn. Multiobjective biogeography-based optimization based on predator-prey approach. *IEEE Transactions on Mag-netics*, 48(2):951–954, 2012.
-

-
- [357] Amir Hossein Gandomi and Amir Hossein Alavi. Krill herd: a new bio-inspired optimization algorithm. *Communications in Nonlinear Science and Numerical Simulation*, 17(12):4831–4845, 2012.
- [358] U Güvenç, Y Sönmez, S Duman, and N Yörükeren. Combined economic and emission dispatch solution using gravitational search algorithm. *Scientia Iranica*, 19(6):1754–1762, 2012.
- [359] Rajesh Kumar, Abhinav Sadu, Rudesh Kumar, and SK Panda. A novel multi-objective directed bee colony optimization algorithm for multi-objective emission constrained economic power dispatch. *Electrical Power Energy System*, 43(1):1241–1250, 2012.
- [360] Mohammad Moradi-Dalvand, Behnam Mohammadi-Ivatloo, Arsalan Najafi, and Abbas Rabiee. Continuous quick group search optimizer for solving non-convex economic dispatch problems. *Electric Power Systems Research*, 93:93–105, 2012.
- [361] Nitin Narang, JS Dhillon, and DP Kothari. Multiobjective fixed head hydrothermal scheduling using integrated predator-prey optimization and powell search method. *Energy*, 47(1):237–252, 2012.
- [362] T Niknam and H Doagou-Mojarrad. Multiobjective economic/emission dispatch by multiobjective θ -particle swarm optimisation. *IET Generation Transmission Distribution*, 6(5):363–377, 2012.
- [363] Taher Niknam, Mohammad Rasoul Narimani, and Rasoul Azizipanah-Abarghooee. A new hybrid algorithm for optimal power flow considering prohibited zones and valve point effect. *Energy Conversion and Management*, 58:197–206, 2012.
- [364] Nicole Pandit, Anshul Tripathi, Shashikala Tapaswi, and Manjaree Pandit. An improved bacterial foraging algorithm for combined static/dynamic environmental economic dispatch. *Applied Soft Computing*, 12(11):3500–3513, 2012.
- [365] Kevin M Passino. Bacterial foraging optimization. *Innovations and Developments of Swarm Intelligence Applications*, pages 219–233, 2012.
- [366] Bo-Yang Qu, Ponnuthurai Nagaratnam Suganthan, and Jane-Jing Liang. Differential evolution with neighborhood mutation for multimodal optimization. *IEEE Transactions on Evolutionary Computing*, 16(5):601–614, 2012.
- [367] Binod Shaw, V Mukherjee, and SP Ghoshal. Solution of economic dispatch problems by seeker optimization algorithm. *Expert Systems with Applications*, 39(1):508–519, 2012.
- [368] Radha Thangaraj, Millie Pant, Thanga Raj Chelliah, and Ajith Abraham. Opposition based chaotic differential evolution algorithm for solving global optimization problems. In *Fourth World Congress on Nature and Biologically Inspired Computing*, pages 1–7, 2012.
- [369] Dieu Ngoc Vo and Weerakorn Ongsakul. Economic dispatch with multiple fuel types by enhanced augmented lagrange hopfield network. *Applied Energy*, 91(1):281–289, 2012.
- [370] Xin-She Yang, Seyyed Soheil Sadat Hosseini, and Amir Hossein Gandomi. Firefly algorithm for solving non-convex economic dispatch problems with valve loading effect. *Applied Soft Computing*, 12(3):1180–1186, 2012.
- [371] Yong Zhang, Dun-Wei Gong, and Zhonghai Ding. A bare-bones multi-objective particle swarm optimization algorithm for environmental/economic dispatch. *Information Science*, 192:213–227, 2012.
- [372] SM Abd-Elazim and ES Ali. A hybrid particle swarm optimization and bacterial foraging for optimal power system stabilizers design. *International Journal of Electrical Power and Energy System*, 46:334–341, 2013.
- [373] AK Barisal. Dynamic search space squeezing strategy based intelligent algorithm solutions to economic dispatch with multiple fuels. *International Journal of Electrical Power and Energy Systems*, 45(1):50–59, 2013.
- [374] M Basu. Hybridization of bee colony optimization and sequential quadratic programming for dynamic economic dispatch. *International Journal of Electrical Power and Energy System*, 44(1):591–596, 2013.
-

-
- [375] Wei-Neng Chen, Juyong Zhang, Yashen Lin, Ni Chen, Zhi-Hui Zhan, Henry Shu-Hung Chung, Yuhua Li, and Yu-Hui Shi. Particle swarm optimization with an aging leader and challengers. *IEEE Transactions on Evolutionary Computation*, 17(2):241–258, 2013.
- [376] Vo Ngoc Dieu and Peter Schegner. Augmented lagrange hopfield network initialized by quadratic programming for economic dispatch with piecewise quadratic cost functions and prohibited zones. *Applied Soft Computing*, 13(1):292–301, 2013.
- [377] AM Elaiw, X Xia, and AM Shehata. Hybrid de-sqp and hybrid pso-sqp methods for solving dynamic economic emission dispatch problem with valve-point effects. *Electrical Power System Research*, 103:192–200, 2013.
- [378] Iztok Fister Jr, Xin-She Yang, Iztok Fister, Janez Brest, and Dušan Fister. A brief review of nature-inspired algorithms for optimization. *Neural and Evolutionary Computing*, 2013.
- [379] Vahid Hosseinneshad and Ebrahim Babaei. Economic load dispatch using θ -pso. *International Journal of Electrical Power and Energy System*, 49:160–169, 2013.
- [380] Momin Jamil and Xin-She Yang. A literature survey of benchmark functions for global optimisation problems. *International Journal of Mathematical Modeling and Numerical Optimization*, 4(2):150–194, 2013.
- [381] Yuancheng Li, Yiliang Wang, and Bin Li. A hybrid artificial bee colony assisted differential evolution algorithm for optimal reactive power flow. *International Journal of Electrical Power and Energy System*, 52:25–33, 2013.
- [382] A Rezaee Jordehi and Jasronita Jasni. Parameter selection in particle swarm optimization: a survey. *Journal of Experimental and Theoretical Artificial Intelligence*, 25(4):527–542, 2013.
- [383] Umesh Kumar Rout, Rabindra Kumar Sahu, and Sidhartha Panda. Design and analysis of differential evolution algorithm based automatic generation control for interconnected power system. *Ain Shams Journal of Engineers*, 4(3):409–421, 2013.
- [384] Provas Kumar Roy and Sudipta Bhui. Multi-objective quasi-oppositional teaching learning based optimization for economic emission load dispatch problem. *International Journal of Electrical Power and Energy System*, 53:937–948, 2013.
- [385] G Giftson Samuel and C Christober Asir Rajan. Hybrid particle swarm optimizationgenetic algorithm and particle swarm optimizationevolutionary programming for long-term generation maintenance scheduling. In *International Conference on Renewable Energy and Sustainable Energy*, pages 227–232, 2013.
- [386] Samir Sayah and Abdellatif Hamouda. A hybrid differential evolution algorithm based on particle swarm optimization for nonconvex economic dispatch problems. *Applied Soft Computing*, 13(4):1608–1619, 2013.
- [387] Dieu N Vo, Peter Schegner, and Weerakorn Ongsakul. Cuckoo search algorithm for non-convex economic dispatch. *IET Generation Transmission Distribution*, 7(6):645–654, 2013.
- [388] Ling Wang and Ling-po Li. An effective differential harmony search algorithm for the solving non-convex economic load dispatch problems. *International Journal of Electrical Power and Energy System*, 44(1):832–843, 2013.
- [389] ZM Yasin, TKA Rahman, and Z Zakaria. Optimal undervoltage load shedding using quantum-inspired evolutionary programming. In *IEEE Spring Conference*, pages 337–341, 2013.
- [390] Ali R Yildiz. Hybrid taguchi-differential evolution algorithm for optimization of multi-pass turning operations. *Applied Soft Computing*, 13(3):1433–1439, 2013.
- [391] M Basu. Fuel constrained economic emission dispatch using nondominated sorting genetic algorithm-ii. *Energy*, 78:649–664, 2014.
- [392] M Basu. Improved differential evolution for economic dispatch. *International Journal of Electrical Power and Energy System*, 63:855–861, 2014.
- [393] Lahouaria Benasla, Abderrahim Belmadani, and Mostefa Rahli. Spiral optimization algorithm for solving combined economic and emission dispatch. *International Journal of Electrical Power and Energy System*, 62:163–174, 2014.
-

-
- [394] Kuntal Bhattacharjee, Aniruddha Bhattacharya, and Sunita Halder nee Dey. Solution of economic emission load dispatch problems of power systems by real coded chemical reaction algorithm. *International Journal of Electrical Power and Energy System*, 59:176–187, 2014.
- [395] Aria Iwasawa, Naohiko Kohtake, Nobuaki Minato, and William Crossley. Investigating the surrogate worth trade-off method to facilitate technology selection for new systems. In *Infranomics*, pages 339–349. Springer, 2014.
- [396] G Klein and D Klinger. Naturalistic decision making. *Human System IAC GATEWAY*, 10 (2014) 16-19, 10:16–19, 2014.
- [397] Barun Mandal, Provas Kumar Roy, and Sanjoy Mandal. Economic load dispatch using krill herd algorithm. *International Journal of Electrical Power and Energy System*, 57:1–10, 2014.
- [398] Mostafa Modiri-Delshad and Nasrudin Abd Rahim. Solving non-convex economic dispatch problem via backtracking search algorithm. *Energy*, 77:372–381, 2014.
- [399] Banaja Mohanty, Sidhartha Panda, and PK Hota. Controller parameters tuning of differential evolution algorithm and its application to load frequency control of multi-source power system. *International Journal of Electrical Power and Energy System*, 54:77–85, 2014.
- [400] Nitin Narang, JS Dhillon, and DP Kothari. Scheduling short-term hydrothermal generation using predator prey optimization technique. *Applied Soft Computing*, 21:298–308, 2014.
- [401] Nitin Narang, JS Dhillon, and DP Kothari. Weight pattern evaluation for multiobjective hydrothermal generation scheduling using hybrid search technique. *International Journal of Electrical Power and Energy Systems*, 62:665–678, 2014.
- [402] Qun Niu, Hongyun Zhang, Xiaohai Wang, Kang Li, and George W Irwin. A hybrid harmony search with arithmetic crossover operation for economic dispatch. *International Journal of Electrical Power and Energy System*, 62:237–257, 2014.
- [403] Jun Sun, Vasile Palade, Xiao-Jun Wu, Wei Fang, and Zhenyu Wang. Solving the power economic dispatch problem with generator constraints by random drift particle swarm optimization. *IEEE Transactions on Industrial Informatics*, 10(1):222–232, 2014.
- [404] Gai-Ge Wang, Amir H Gandomi, Amir H Alavi, and Guo-Sheng Hao. Hybrid krill herd algorithm with differential evolution for global numerical optimization. *Neural Computing Applied*, 25(2):297–308, 2014.
- [405] Qingzheng Xu, Lei Wang, Na Wang, Xinhong Hei, and Li Zhao. A review of opposition-based learning from 2005 to 2012. *Engineering Applications of Artificial Intelligence*, 29:1–12, 2014.
- [406] Zeyin Zhang, Ting Wang, and Xinggao Liu. Melt index prediction by aggregated rbf neural networks trained with chaotic theory. *Neurocomputing*, 131:368–376, 2014.
- [407] R Arul, S Velusami, and G Ravi. A new algorithm for combined dynamic economic emission dispatch with security constraints. *Energy*, 79:496–511, 2015.
- [408] M Balasubbareddy, S Sivanagaraju, and Chintalapudi V Suresh. Multi-objective optimization in the presence of practical constraints using non-dominated sorting hybrid cuckoo search algorithm. *An International Journal of Engineers Science and Technology*, 18(4):603–615, 2015.
- [409] Sumit Banerjee, Deblina Maity, and Chandan Kumar Chanda. Teaching learning based optimization for economic load dispatch problem considering valve point loading effect. *International Journal of Electrical Power and Energy System*, 73:456–464, 2015.
- [410] AK Barisal and RC Prusty. Large scale economic dispatch of power systems using oppositional invasive weed optimization. *Applied Soft Computing*, 29:122–137, 2015.
- [411] M Basu. Modified particle swarm optimization for nonconvex economic dispatch problems. *International Journal of Electrical Power and Energy System*, 69:304–312, 2015.
- [412] Marco A Contreras-Cruz, Victor Ayala-Ramirez, and Uriel H Hernandez-Belmonte. Mobile robot path planning using artificial bee colony and evolutionary programming. *Applied Soft Computing*, 30:319–328, 2015.
- [413] Hari Mohan Dubey, Manjaree Pandit, and BK Panigrahi. A biologically inspired modified flower pollination algorithm for solving economic dispatch problems in modern power systems. *Cognitive Computing*, 7(5):594–608, 2015.
-

-
- [414] Serhat Duman, Nuran Yorukeren, and Ismail H Altas. A novel modified hybrid psogsa based on fuzzy logic for non-convex economic dispatch problem with valve-point effect. *International Journal of Electrical Power and Energy System*, 64:121–135, 2015.
- [415] Wael T Elsayed and Ehab F El-Saadany. A fully decentralized approach for solving the economic dispatch problem. *IEEE Transactions on Power System*, 30(4):2179–2189, 2015.
- [416] Li Han, Carlos E Romero, and Zheng Yao. Economic dispatch optimization algorithm based on particle diffusion. *Energy Conversion and Management*, 105:1251–1260, 2015.
- [417] Vinay Kumar Jadoun, Nikhil Gupta, KR Niazi, and Anil Swarnkar. Modulated particle swarm optimization for economic emission dispatch. *International Journal of Electrical Power and Energy System*, 73:80–88, 2015.
- [418] Solmaz Kahourzade, Amin Mahmoudi, and Hazlie Bin Mokhlis. A comparative study of multi-objective optimal power flow based on particle swarm, evolutionary programming, and genetic algorithm. *Electrical Engineering*, 97(1):1–12, 2015.
- [419] Su Nguyen, Mengjie Zhang, Mark Johnston, and Kay Chen Tan. Automatic programming via iterated local search for dynamic job shop scheduling. *IEEE Transactions on cybernetics*, 45(1):1–14, 2015.
- [420] Arman Riaz Ochi, Mohammad Abdul Mannan, and Bashudeb Chandra Ghosh. Optimal allocation of facts devices using quantum inspired evolutionary programming. *Trends in Electrical Engineering*, 5(1):23–27, 2015.
- [421] Bijay Baran Pal and Mousumi Kumar. A hierarchical optimization method to solve environmental-economic power generation and dispatch problem with fuzzy data uncertainty. In *Information Systems Design and Intelligent Applications*, pages 419–434. Springer, 2015.
- [422] Manjaree Pandit, Laxmi Srivastava, and Manisha Sharma. Environmental economic dispatch in multi-area power system employing improved differential evolution with fuzzy selection. *Applied Soft Computing*, 28:498–510, 2015.
- [423] Dinu Calin Secui. The chaotic global best artificial bee colony algorithm for the multi-area economic/emission dispatch. *Energy*, 93:2518–2545, 2015.
- [424] Dinu Calin Secui. A new modified artificial bee colony algorithm for the economic dispatch problem. *Energy Conversion and Management*, 89:43–62, 2015.
- [425] AY Abdelaziz, ES Ali, and SM Abd Elazim. Combined economic and emission dispatch solution using flower pollination algorithm. *International Journal of Electrical Power and Energy System*, 80:264–274, 2016.
- [426] BR Adarsh, T Raghunathan, T Jayabarathi, and Xin-She Yang. Economic dispatch using chaotic bat algorithm. *Energy*, 96:666–675, 2016.
- [427] Ashik Ahmed and Md Shahid Ullah. Optimal design of proportional–integral controllers for stand-alone solid oxide fuel cell power plant using differential evolution algorithm. *SpringerPlus*, 5(1):1, 2016.
- [428] WA Augusteen, R Kumari, and R Rengaraj. Economic and various emission dispatch using differential evolution algorithm. In *3rd International Conference Electrical Energy Systems*, pages 74–78, 2016.
- [429] M Balamurugan, S Narendiran, Sarat Kumar Sahoo, Raja Das, and Ashwin Kumar Sahoo. Application of particle swarm optimization for maximum power point tracking in pv system. In *3rd International Conference on Electrical Energy Systems*, pages 35–38, 2016.
- [430] M Basu. Kinetic gas molecule optimization for nonconvex economic dispatch problem. *International Journal of Electrical Power and Energy System*, 80:325–332, 2016.
- [431] M Basu. Multi-objective optimal reactive power dispatch using multi-objective differential evolution. *International Journal of Electrical Power and Energy System*, 82:213–224, 2016.
- [432] M Basu. Quasi-oppositional differential evolution for optimal reactive power dispatch. *International Journal of Electrical Power and Energy System*, 78:29–40, 2016.
-

-
- [433] K Bavithra, S Charles Raja, and P Venkatesh. Optimal setting of facts devices using particle swarm optimization for atc enhancement in deregulated power system. *IFAC-PapersOnLine*, 49(1):450–455, 2016.
- [434] Kusum Kumari Bharti and Pramod Kumar Singh. Chaotic gradient artificial bee colony for text clustering. *Soft Computing*, 20(3):1113–1126, 2016.
- [435] F Chen, GH Huang, YR Fan, and RF Liao. A nonlinear fractional programming approach for environmental-economic power dispatch. *International Journal of Electrical Power and Energy System*, 78:463–469, 2016.
- [436] Zeyu Chen, Rui Xiong, and Jiayi Cao. Particle swarm optimization-based optimal power management of plug-in hybrid electric vehicles considering uncertain driving conditions. *Energy*, 96:197–208, 2016.
- [437] Vun Jack Chin, Zainal Salam, and Kashif Ishaque. An accurate modelling of the two-diode model of pv module using a hybrid solution based on differential evolution. *Energy Conversion and Management*, 124:42–50, 2016.
- [438] Narges Daryani, Mehrdad Tarafdar Hagh, and Saeed Teimourzadeh. Adaptive group search optimization algorithm for multi-objective optimal power flow problem. *Applied Soft Computing*, 38:1012–1024, 2016.
- [439] Maria Jose del Jesus and J Aguilera. Estimating the maximum power delivered by concentrating photovoltaics technology through atmospheric conditions using a differential evolution approach. In : 11th *International Conference on Hybrid Artificial Intelligent Systems*, volume 9648, page 273. Springer, 2016.
- [440] Zohar Drach, Shany Hershkovitz, Domenico Ferrero, Pierluigi Leone, Andrea Lanzini, Massimo Santarelli, and Yoed Tsur. Impedance spectroscopy analysis inspired by evolutionary programming as a diagnostic tool for soec and sofc. *Solid State Ionics*, 288:307–310, 2016.
- [441] Sotirios K Goudos, Margot Deruyck, David Plets, Luc Martens, and Wout Joseph. Optimization of power consumption in wireless access networks using differential evolution with eigenvector based crossover operator. In 10th *European Conference on Antennas and Propagation*, pages 1–4, 2016.
- [442] Xiangzhu He, Yunqing Rao, and Jida Huang. A novel algorithm for economic load dispatch of power systems. *Neurocomputing*, 171:1454–1461, 2016.
- [443] JQ James and Victor OK Li. A social spider algorithm for solving the non-convex economic load dispatch problem. *Neurocomputing*, 171:955–965, 2016.
- [444] C Jena, M Basu, and CK Panigrahi. Differential evolution with gaussian mutation for combined heat and power economic dispatch. *Soft Computing*, 20(2):681–688, 2016.
- [445] Yacine Labbi, Djilani Ben Attous, Hossam A Gabbar, Belkacem Mahdad, and Aboelsood Zidan. A new rooted tree optimization algorithm for economic dispatch with valve-point effect. *International Journal of Electrical Power and Energy System*, 79:298–311, 2016.
- [446] Faa-Jeng Lin, Kuang-Hsiung Tan, and Chia-Hung Tsai. Improved differential evolution-based elman neural network controller for squirrel-cage induction generator system. *IET Renewable Power Generation*, 2016.
- [447] Xiao-Fang Liu, Zhi-Hui Zhan, Jun-Hao Lin, and Jun Zhang. Parallel differential evolution based on distributed cloud computing resources for power electronic circuit optimization. In *Genetic and Evolutionary Computation Conference Companion*, pages 117–118, 2016.
- [448] Akanksha Mishra and Gundavarapu Venkata Nagesh Kumar. A fast contingency management approach with optimal placement and tuning of interline power flow controller using differential evolution algorithm. *Electric Power Components and Systems*, pages 1–15, 2016.
- [449] Mostafa Modiri-Delshad and Nasrudin Abd Rahim. Multi-objective backtracking search algorithm for economic emission dispatch problem. *Applied Soft Computing*, 40:479–494, 2016.
- [450] Mahaletchumi Morgan, Nor Rul Hasma Abdullah, Mohd Herwan Sulaiman, Mahfuzah Mustafa, and Rosdiyana Samad. Benchmark studies on optimal reactive power dispatch based multi-objective evolutionary programming using mutation based on adaptive mutation operator and polynomial mutation operator. *Journal of Electrical Systems*, 12(1), 2016.
-

-
- [451] Mei Shan Ngan and Chee Wei Tan. Photovoltaic multiple peaks power tracking using particle swarm optimization with artificial neural network algorithm. In *Advances in Solar Photovoltaic Power Plants*, pages 107–138. Springer, 2016.
- [452] K Rajesh, A Bhuvanesh, S Kannan, and C Thangaraj. Least cost generation expansion planning with solar power plant using differential evolution algorithm. *Renewable Energy*, 85:677–686, 2016.
- [453] K Rajesh, S Kannan, and C Thangaraj. Least cost generation expansion planning with wind power plant incorporating emission using differential evolution algorithm. *International Journal of Electrical Power and Energy System*, 80:275–286, 2016.
- [454] Roberto Soto-Villalobos, Fernando Velasco-Tapia, Francisco Javier, Sóstenes Méndez-Delgado Almaguer-Matrinez, and Francisco Gerardo Benavides-Bravo. Partial melting inverse modeling of earth upper mantle by evolutionary programming optimization. *Main Themes*, page 236, 2016.
- [455] Amiyne Zakouni, Jiawei Luo, and Fouad Kharroubi. Solving the static manycast RWA problem in optical networks using evolutionary programming. In *International Conference on Intelligent Computing*, pages 147–157. Springer, 2016.
- [456] Science Direct, 25.8.2016.
- [457] IEEE Xplore Digital Library, 25.8.2016.
- [458] Percy Williams Bridgman. *Dimensional analysis*. Yale University Press, 1922.
- [459] Carlos Coello, Gary B Lamont, and David A Van Veldhuizen. *Evolutionary algorithms for solving multi-objective problems*. Springer Science and Business Media, 2007.
- [460] Lawrence Davis. *Handbook of genetic algorithms*. 1991.
- [461] Kalyanmoy Deb. *Multi-objective optimization using evolutionary algorithms*, volume 16. John Wiley and Sons, 2001.
- [462] Kalyanmoy Deb. *Optimization for engineering design: Algorithms and examples*. PHI Learning Pvt. Ltd., India, 2012.
- [463] JS Dhillon and DP Kothari. *Power system optimization*. PHI Learning Pvt. Ltd., India, 2004.
- [464] Matthias Ehrgott. *Multicriteria optimization*. Springer Science and Business Media, 2006.
- [465] David B Fogel. *Artificial Intelligence through simulated evolution*. Wiley-IEEE Press, 2009.
- [466] Tomas Gal, Theodor Stewart, and Thomas Hanne. *Multicriteria decision making: advances in MCDM models, algorithms, theory, and applications*, volume 21. Springer Science and Business Media, 2013.
- [467] David E Goldberg. *Genetic algorithms*. Pearson Education India, New Delhi, 2006.
- [468] Donald L. Karner. *Evolutionary Ecology: Concepts and Case studies*. Oxford University Press, New York, 2001.
- [469] Alexander Lotov, Vladimir A Bushenkov, and Georgy K Kamenev. *Interactive decision maps: Approximation and visualization of Pareto frontier*, volume 89. Springer Science and Business Media, 2013.
- [470] Kaisa Miettinen. *Nonlinear multiobjective optimization*, volume 12. Springer Science and Business Media, 2012.
- [471] Kenneth Price, Rainer M Storn, and Jouni A Lampinen. *Differential evolution: a practical approach to global optimization*. Springer Science and Business Media, 2006.
- [472] Singiresu S Rao and SS Rao. *Engineering optimization: theory and practice*. John Wiley and Sons, 2009.
- [473] Ralph E Stever. *Multiple Criteria Optimization: Theory, Computing and Application*. John Wiley and Sons, 1986.
- [474] Rainer Storn and Kenneth Price. *Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces*, volume 3. International Computer Science Institute, Berkeley, 1995.
-

-
- [475] El-Ghazali Talbi. *Metaheuristics: from design to implementation*, volume 74. John Wiley and Sons, 2009.
- [476] Thomas Lange Vincent and Walter Jervis Grantham. *Optimality in parametric systems*. John Wiley and Sons, 1981.
- [477] Bo Xing and Wen-Jing Gao. *Invasive Weed Optimization Algorithm*, pages 177–181. Springer International Publishing, Cham, 2014.
- [478] Xin-She Yang. *Flower Pollination Algorithm for Global Optimization*, pages 240–249. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
- [479] Xin-She Yang. *Nature-Inspired Metaheuristic Algorithms*. Luniver Press, 2008.
- [480] Xin-She Yang. *A New Metaheuristic Bat-Inspired Algorithm*, pages 65–74. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

APPENDICES

APPENDIX A

RANDOM SEARCH TECHNIQUES

A.1 PARTICLE SWARM OPTIMIZATION (PSO)

In particle swarm optimization each individual in PSO flies in the search space with a velocity which is dynamically adjusted according to its own flying experience and its companions flying experience. Each individual is treated as a volume-less particle (a point) in the N_g dimensional search space. The i^{th} particle is represented as $X_i = (x_{i1}, x_{i2}, \dots, x_{iN_g})$. The best previous position (the position giving the best fitness value) of the i^{th} particle is recorded and represented as $P_i = (p_{i1}, p_{i2}, \dots, p_{iN_g})$. The index of the best particle among all the particles in the population is represented by the symbol g^B . The velocity update for i^{th} particle is given by the following equation [54]:

$$v_{ij}^{t+1} = wv_{ij}^t + c_1rand()(x_{ij}^B - x_{ij}^t) + c_2rand()(g_j^B - x_{ij}^t) \quad (\text{A.1})$$
$$(j = 1, 2, \dots, N_g; i = 1, 2, \dots, N_P)$$

where $w \in [0.4, 0.9]$ is an inertia weight factor; c_1 and c_2 are acceleration constant. $rand() \in [0, 1]$ is a uniformly distributed random number. x_{ij}^B and g_j^B are the individual's best and swarm's best experienced positions respectively. The position of i^{th} particle will be updated as follows:

$$x_{ij}^{t+1} = x_{ij}^t + v_{ij}^{t+1} \quad (j = 1, 2, \dots, N_g; i = 1, 2, \dots, N_P) \quad (\text{A.2})$$

The above stated steps are repeated in an iterative process. Each X_i^t is evaluated using objective function and on basis of objective function value the individual best experience x_{ij}^B and global best experience g^B is updated. The iterative process continues until termination criteria is satisfied.

A.2 PREDATOR PREY OPTIMIZATION (PPO)

PPO is a method which takes a cue from the behavior of schools of sardines and pods of killer whales. In this method, particles are divided into two categories, predator and prey. Predators show the behavior of chasing the center of preys' swarm; it looks like chasing preys. And preys escape from predators in multidimensional solution space. This contributes to make the particles avoid the local optimal solutions and find the global optimal solution. The mathematical expressions governing the behavior of PPO are as follows [118]:

$$v_{ij}^{t+1} = wv_{ij}^t + c_1rand()(x_{ij}^B - x_{ij}^t) + c_2rand()(g_j^B - x_{ij}^t) + c_3rand()(ae^{-b\beta}) \quad (\text{A.3})$$

$$(j = 1, 2, \dots, N_g; i = 1, 2, \dots, N_P)$$

where a represents the maximum amplitude of the predator's effect over prey. b controls the distance at which the effect is still significant. β is the Euclidean distance between predator and prey. P_f and F_f are probability of fear and fear factor, respectively. β is the Euclidean distance between predator and prey given by $|g_j^B - PP_j|$.

The position of i^{th} particle will be updated as follows:

$$x_{ij}^{t+1} = x_{ij}^t + v_{ij}^{t+1} \quad (j = 1, 2, \dots, N_g; i = 1, 2, \dots, N_P) \quad (\text{A.4})$$

The objective of predator is to pursue the best individual in the prey swarm and has disruptive effect on swarm. The predators velocity vp_j and position, xp_j are updated as:

$$vp_j^{t+1} = vp_j^t + c_p rand()(g_j^B - xp_j^t) \quad (j = 1, 2, \dots, N_g) \quad (\text{A.5})$$

$$xp_j^{t+1} = xp_j^t + vp_j^{t+1} \quad (j = 1, 2, \dots, N_g) \quad (\text{A.6})$$

where c_p is acceleration constant. The above stated steps are repeated in an iterative process. Each X_i^t is evaluated using objective function and on basis of objective function value the individual best experience x_{ij}^B and global best experience g^B is updated. The iterative process continues until termination criteria is satisfied.

A.3 ANTI PREDATOR PREY OPTIMIZATION (APSO)

The anti-predatory property, which is natural among birds, helps the swarm to escape from the predators. In the original PSO, food locations are modeled as the best result points of the optimization problem. In this model the predators are the worst result points. The APSO is developed by splitting both the cognitive and social behaviors of the classical PSO. The cognitive behavior is divided into good experience and bad experience components. The former refers to a bird's memory about previously visited best position and the later refers to a bird's memory about the previously visited worst position (location of predator). Similarly, the social behavior of the classical PSO is split into global good experience and global bad experience components. To calculate the velocity of a particle, the proposed APSO considers the particle's bad experience along with good experience. Then, the velocity-update equation for APSO model is given by [262]:

$$v_{ij}^{t+1} = wv_{ij}^t + c_1rand()(x_{ij}^B - x_{ij}^t) + c_2rand()(g_j^B - x_{ij}^t) + c_3rand()(x_{ij}^t - x_{ij}^W) + c_4rand()(x_{ij}^t - g_j^W) \quad (j = 1, 2, \dots, N_g; i = 1, 2, \dots, N_P) \quad (\text{A.7})$$

where $w \in [0.4, 0.9]$ is an inertia weight factor; c_3 and c_4 are constants. x_{ij}^W and g_j^W are the individual's best and swarm's best experienced positions respectively. The position of i^{th} particle will be updated as follows:

$$x_{ij}^{t+1} = x_{ij}^t + v_{ij}^{t+1} \quad (\text{A.8})$$

The above stated steps are repeated in an iterative process. Each X_i^t is evaluated using objective function and on basis of objective function value the individual best experience

x_{ij}^B , global best experience g^B , individual worst experience x_{ij}^W , global worst experience g^W is updated. The iterative process continues until termination criteria is satisfied.

A.4 DIFFERENTIAL EVOLUTION (DE)

DE is a parallel direct search method. It utilizes N_P number of parameter vectors x_i^t in N_g dimensional search space. The initial vector population is chosen randomly and should cover the entire parameter space. As a rule, a uniform probability distribution is followed. In case a preliminary solution is available, the initial population might be generated by adding normally distributed random deviations to the nominal solution. DE generates new parameter vectors by adding the weighted difference between two population vectors to a third vector using mutation operation. The mutated vector's are then mixed with the another predetermined vector, the target vector, to yield a trial vector. Parameter mixing is often referred to as crossover. If the trial vector are better in quality than the target vector, the trial vector replaces the target vector in the following generation. This last operation is called selection. Each population vector has to serve once as the target vector, so that N_P competitions take place in one generation. For each x_{ij}^t the mutation operator generates a mutant vector v_{ij}^t as per following relation [474, 75]:

$$v_{ij}^t = x_{\chi_1 j}^t + F(x_{\chi_2 j}^t - x_{\chi_3 j}^t) \quad (j = 1, 2, \dots, N_g, i = 1, 2, \dots, N_P) \quad (\text{A.9})$$

where F is the mutation factor; $\chi_1 \neq \chi_2 \neq \chi_3 \neq i \in [1, N_P]$ index are selected by uniformly distributed random number. Further, to increase the diversity of mutant individual binomial crossover operation is used, resulting in trial vector. Mathematically, it is written as

$$u_{ij}^{t+1} = \begin{cases} v_{ij}^t & \text{if } (\chi_j \leq C_R \text{ or } j = \chi_i) \\ x_{ij}^t & \text{otherwise} \end{cases} \quad (j = 1, 2, \dots, N_g, i = 1, 2, \dots, N_P) \quad (\text{A.10})$$

where $\chi_j \in [0, 1]$ is the j^{th} evaluation of random number. $C_R \in [0, 1]$ is the crossover constant. $\chi_i \in [1, N_g]$ is a randomly chosen index by exploiting chaotic sequence.

In order to decide, whether a trial vector is selected as a member of $G+1$ generation, the survival of the fittest principle is followed. Mathematically,

$$x_{ij}^{t+1} = \begin{cases} u_{ij}^{t+1} & \text{if } u_{ij}^{t+1} < x_{ij}^t \\ x_{ij}^t & \text{otherwise} \end{cases} \quad (j = 1, 2, \dots, N_g, i = 1, 2, \dots, N_P) \quad (\text{A.11})$$

The above stated process of mutation, crossover and selection is repeated in an iterative way to update each X_i^t . The iterative process continues until termination criteria is satisfied and upon termination best solution available in population is treated as final solution of the problem.

A.5 EVOLUTIONARY PROGRAMMING (EP)

Evolutionary Programming is initially proposed as an alternative for artificial intelligence. EP follows the two step evolution concept *viz.* mutate the current population and selection of solution from the current and mutated population. Each individual in the population is a real valued vector (x_i, η_i) , where x_i is the objective variable and η_i is the associated strategy parameter. Mutation is responsible for generation of new solutions whereas selection operation decides the survival in next population. Mathematically, the working concept of EP is illustrated as follows [38]:

$$x_{ij}^{t+1} = x_{ij}^t + \eta_{ij}^t N_j(0, 1) \quad (\text{A.12})$$

$$\eta_{ij}^{t+1} = \eta_{ij}^t \exp(\tau^{t+1} N(0, 1) + \tau N_j(0, 1)) \quad (j = 1, 2, \dots, N_g, i = 1, 2, \dots, N_P) \quad (\text{A.13})$$

where $N_j(0, 1)$ is the normally distributed random number with mean 0 and 1 standard deviation, generated for j^{th} individual. The parameters τ and τ^{t+1} are given by $(\sqrt{2\sqrt{n}})^{-1}$ and $(\sqrt{2n})^{-1}$.

The iterative process to update strategy parameter η_{ij}^t and offspring x_{ij}^t generation is repeated along with selection operation to decide parents for $k + 1^{\text{th}}$ generation. The iterative process continues till termination criteria is satisfied.

APPENDIX B

DIRECT SEARCH METHODS

The direct search methods are classified under the category of local search techniques and have better exploitation capabilities than global search techniques. A direct search algorithm does not require partial derivatives of the objective function and are called non-gradient methods. Among the various direct search methods, Powell's method and Hooke Jeeve's method are one of the promising search techniques. The detailed discussion of these methods is as follows:

B.1 POWELL'S METHOD

This local search method is proposed by Michael J.D. Powell [462]. Powell's method requires, a set of directions $[d_1, \dots, d_{N_g}]^T$ that are iteratively updated to approximate the direction pointing to x^* . Given an initial point x^0 , at k^{th} iteration, n steps are taken using the n directions with step size λ for a new point x_i^k . Mathematically,

$$x_{i+1}^k = x_i^k + \lambda d_i \quad (i = 1, 2, \dots, N_g) \quad (\text{B.1})$$

The search directions are updated as: $d = x_{n+1}^k - x_1^k$. The old directions are shifted, $d_i = d_{i+1}, i = 1, \dots, N_g - 1$ and iterations continue with the updated directions until $|f(x_1^{k+1}) - f(x_1^k)| < \epsilon$. The search process is shown in algorithm B.1.

B.2 HOOKES JEEVE'S METHOD

This direct search method is propose by Hooke and Jeeves [3]. In this method a combi-

Algorithm B.1 Powell's algorithm

Input: x^0 and ϵ $k \leftarrow 0, (d_0, \dots, d_n) \leftarrow (e_0, \dots, e_n), x_1^1 \leftarrow x^0$ **repeat**

++k

for $i = 1$ to n **do** $\lambda = \operatorname{argmin}_{\mu} f(x_i^k + \mu d_i)$ $x_1^{k+1} = x_{n+1}^k + \lambda d_i$ **end for** $d = x_{n+1}^k - x_1^k$ $x_1^{k+1} = x_{n+1}^k + \lambda d; \lambda = \operatorname{argmin} f(x_{n+1}^k + \mu d)$ **until** $|f(x_1^{k+1}) - f(x_1^k)| < \epsilon$

Algorithm B.2 Hooke's Jeeves algorithm

Input: $x^0, \Delta_i, \alpha > 1$ and ϵ $k \leftarrow 0$ $x^c = x^k$ **for** $i = 1$ to N_g **do** $f = f(x^c), f^+ = f(x_i^c + \Delta_i)$ and $f^- = f(x_i^c - \Delta_i)$ $f_{min} = \min(f, f^+, f^-)$ $x^c \leftarrow x_{f_{min}}$ **end for****if** $f(x^c) < f(x^k)$ **then** $f(x^{k+1}) = f(x^c)$ **repeat**

++k

 $x_p^{k+1} = x^k + (x^k - x^{k-1})$ $x^c = x_p^{k+1}$ **for** $i = 1$ to N_g **do** $f = f(x^c), f^+ = f(x_i^c + \Delta_i)$ and $f^- = f(x_i^c - \Delta_i)$ $f_{min} = \min(f, f^+, f^-)$ $x^c \leftarrow x_{f_{min}}$ **end for****if** $f(x^c) < f(x^k)$ **then** $f(x^{k+1}) = f(x^c)$ **end if****until** $f(x_1^{k+1}) < f(x^k)$ **else****if** $\|\Delta\| > \epsilon$ **then** $\Delta_i \leftarrow \Delta_i/2$ **else**

Terminate

end if**end if**

nation of exploratory and pattern moves is used, where exploratory search find the best point in the vicinity of the current point and Pattern search jumps in the direction of change, if better then proceed in that direction, else reduce size of exploratory move and continue. Pattern Search is a search at bigger magnitude and create a set of search directions iteratively which are linearly independent. The Hooke Jeeve's search process is shown in algorithm B.2.

APPENDIX C

GENERALIZED BENCHMARK FUNCTIONS

The benchmark functions undertaken to authenticate the applicability to solve optimization problem having discontinuous, non-differentiable, non-separable and multi-modal features are given below:

Minimize

1. **Griewank Function:** [380] Mathematically, it is defined as:

Minimize

$$F_2(x) = \sum_{i=1}^n \frac{x_i^2}{4000} - \prod \cos\left(\frac{x_i}{\sqrt{i}}\right) \quad (\text{C.1})$$

subject to $(-600 \leq x_i \leq 600)$

This function is continuous, differentiable, non-separable, scalable and multi-modal in nature. The global minimum is $f(x^*) = 0$, $x^* = f(0, \dots, 0)$.

2. **Rastrigins Function:** [380] Mathematically, it is defined as:

Minimize

$$F_3(x) = \sum_{i=1}^n [x_i^2 - 10\cos(2\pi x_i)] \quad (\text{C.2})$$

subject to $(-10 \leq x_i \leq 10)$

This function is highly multi-modal and non-differentiable in nature. The global minima is located at $x^* = f(0, \dots, 0)$, $f(x^*) = 0$.

3. **Rosenbrocks Function:** [380] Mathematically, it is defined as:

Minimize

$$F_4(x) = \sum_{i=1}^{n-1} [(x_i - 1)^2 + 100(x_{i+1} - x_i^2)^2] \quad (\text{C.3})$$

subject to $(-30 \leq x_i \leq 30)$

The global minimum is located at $x^* = f(1, \dots, 1)$, $f^* = 0$.

4. **Schwefel 2.22 Function:** [380] Mathematically, it is defined as:

Minimize

$$F_5(x) = \sum_{i=1}^n |x_i| + \prod_{i=1}^n |x_i| \quad (\text{C.4})$$

subject to $(-10 \leq x_i \leq 10)$ The global minima is located at $x^* = f(0, \dots, 0)$ and the minima is $f(x) = 0$.

5. **Generalized Schwefel Function:** [380] Mathematically, it is defined as:

Minimize

$$F_1(x) = - \sum_{i=1}^n x_i \sin(\sqrt{|x_i|}) \quad (\text{C.5})$$

subject to $-500 \leq x_i \leq 500$

This function is continuous, differentiable, separable, scalable and multi-modal in nature. The global minimum is $f(x^*) = 418.9829n$, $x^* = 420.9687$

6. **Sphere Function:** [380] It is the simplest of De Jongs functions. Mathematically, it is defined as:

Minimize

$$F_6(x) = \sum_{i=1}^n x_i^2 \quad (\text{C.6})$$

subject to $(-10 \leq x_i \leq 10)$ $(i = 1, 2, \dots, n)$

This function is continuous, differentiable, separable, scalable, unimodal in nature.

The global minima is located at $x^* = f(0, \dots, 0)$, $f(x^*) = 0$.

7. **Step Function:** [380] Mathematically, it is defined as:

Minimize

$$F_7(x) = \sum_{i=1}^n [|x_i|] \tag{C.7}$$

subject to $(-10 \leq x_i \leq 10)$

This function is discontinuous, non-differentiable, separable, scalable and unimodal in nature. The global minima is located $x^* = f(0, \dots, 0) = 0$, $f(x^*) = 0$.

8. **Step 2 Function:** [380] Mathematically, it is defined as:

Minimize

$$F(x) = \sum_{i=1}^n [|x_i + 0.5|] \tag{C.8}$$

subject to $(-10 \leq x_i \leq 10)$

This function is discontinuous, non-differentiable, separable, scalable and unimodal in nature. The global minima is located $x^* = f(0.5, \dots, 0.5) = 0$, $f(x^*) = 0$.

APPENDIX D

TEST POWER SYSTEMS

In order to validate the capability of proposed algorithms to solve real life engineering problems, the power system dispatch problems are considered. These problems are economic load dispatch, minimum emission dispatch and multiobjective load dispatch. The power generators of considered power systems have smooth/ non-smooth cost function. The power generators of these systems have additional complexity of prohibited operating zone and ramp rate limit constraints. In order to, include the power transmission losses in the study, the *B-coefficients* are used to obtain transmission losses. The systems used in study have small, medium and large power generation capacity. The small generation capacity, electric power system consists of ten multi-fuel operated generators and additional complexities of generation and transmission system. The medium capacity system has forty power generators with non-smooth behavior, whereas the large capacity electric power system has 140 power generators with additional complexities. The power generator characteristics of considered systems along with their reference are as follows:

D.1 SYSTEM DATA FOR TEST POWER SYSTEM-1

A power generation unit with multiple cost curves needs to operate on the lower contour of the intersecting cost curves. The resulting cost function is a combined cost function representing impact of multiple fuels. Each segment of the combined cost function implies some information about the fuel being used for the operation of power generator. A test power system consisting of ten-thermal power generating units considering multiple fuel

Table D.1: Power generator's cost coefficients

Unit j	Fuel type FT	Cost coefficients				
		$a_{j,FT}$ (\$/MW ² h)	$b_{j,FT}$ (\$/MW ² h)	$c_{j,FT}$ (\$/h)	$d_{j,FT}$ (\$/h)	$e_{j,FT}$ (rad/MW)
1	1	0.002176	-0.3975	26.97	0.02697	-3.975
	2	0.001861	-0.3059	21.13	0.02113	-3.059
2	1	0.004194	-1.269	118.4	0.1184	-12.69
	2	0.001138	-0.03988	1.865	0.001865	-0.3988
	3	0.00162	-0.198	13.65	0.01365	-1.98
3	1	0.001457	-0.3116	39.79	0.03979	-3.116
	2	1.18E-05	0.4864	-59.14	-0.05914	4.864
	3	0.000804	0.03389	-2.875	-0.00288	0.3389
4	1	0.001049	-0.03114	1.983	0.001983	-0.3114
	2	0.002758	-0.6348	52.85	0.05285	-6.348
	3	0.005935	-2.338	266.8	0.2668	-23.38
5	1	0.001066	-0.08733	13.92	0.01392	-0.8733
	2	0.001597	-0.5206	99.76	0.09976	-5.206
	3	0.00015	0.4462	-53.99	-0.05399	4.462
6	1	0.002758	-0.6348	52.85	0.05285	-6.348
	2	0.001049	-0.03114	1.983	0.001983	-0.3114
	3	0.005935	-2.338	266.8	0.2668	-23.38
7	1	0.001107	-0.1325	18.93	0.01893	-1.325
	2	0.001165	-0.2267	43.77	0.04377	-2.267
	3	0.000245	0.3559	-43.35	-0.04335	3.559
8	1	0.001049	-0.03114	1.983	0.001983	-0.3114
	2	0.002758	-0.6348	52.85	0.05285	-6.348
	3	0.005935	-2.338	266.8	0.2668	-23.38
9	1	0.001554	-0.5675	88.53	0.08853	-5.675
	2	0.007033	-0.04514	15.3	0.01423	-0.1817
	3	0.000612	-0.01817	14.23	0.01423	-0.1817
10	1	0.001102	-0.09938	13.97	0.01397	-0.9938
	2	4.16E-05	0.5084	-61.13	-0.06113	5.084
	3	0.001137	-0.2024	46.71	0.04671	-2.024

options without and with valve point loading effect is studied for 2700 MW power demand. The multiple fuel cost characteristics are referred from [163] and are tabulated in Table D.1. Table D.2 provides the power generation limits of generators and the type of fuels available for generator and corresponding fuel limits.

The multi fuel power generators have additional complexity of prohibited operating zone constraints. The corresponding information has been referred from [376] and tabulated in D.3. The table provides information about number of prohibited operating zones applied on a power generator and their corresponding limits. The multi-fuel system has been considered in multiobjective framework and the gaseous pollutant's emission coeffi-

Table D.2: Test power system-1, Fuel option and corresponding power generator limits

Generator		Fuel and power generator limits				
j		P_j^{min}				P_j^{max}
1	FT		[1]		[2]	
	Limit	[100.0]		[196.0]		[250.0]
2	FT		[2]		[3]	[1]
	Limit	[50.0]	[114.0]		[157.0]	[230.0]
3	FT		[1]		[3]	[2]
	Limit	[200.0]	[332.0]		[388.0]	[500.0]
4	FT		[1]		[2]	[3]
	Limit	[99.0]	[138.0]		[200.0]	[265.0]
5	FT		[1]		[2]	[3]
	Limit	[190.0]	[338.0]		[407.0]	[490.0]
6	FT		[2]		[1]	[3]
	Limit	[85.0]	[138.0]		[200.0]	[265.0]
7	FT		[1]		[2]	[3]
	Limit	[200.0]	[331.0]		[391.0]	[500.0]
8	FT		[1]		[2]	[3]
	Limit	[99.0]	[138.0]		[200.0]	[265.0]
9	FT		[3]		[1]	[3]
	Limit	[130.0]	[213.0]		[370.0]	[440.0]
10	FT		[1]		[3]	[2]
	Limit	[200.0]	[362.0]		[407.0]	[490.0]

Table D.3: Test power system-1, Prohibited operating zone (POZ)

Generator	Number of POZ	POZ limit (MW)		
3	3	[215.0, 225.0]	[305.0, 335.0]	[420.0, 450.0]
5	3	[200.0, 220.0]	[260.0, 335.0]	[390.0, 420.0]
7	3	[230.0, 255.0]	[365.0, 395.0]	[430.0, 455.0]
10	2	[270.0, 295.0]	[380.0, 400.0]	

coefficients tabulated in Table D.4 have been referred for implementation of minimum emission dispatch of undertaken system.

In order to, consider the network transmission losses along with multi-fuel system, the B-coefficients available in [330] for 10 generator system are considered. The *B-coefficients* $B_{ij}(MW^{-1})$, B_{j0} and $B_{00}(MW)$ are tabulated in Table D.5 for reference.

Table D.4: Gaseous pollutant's emission coefficient

unit j	Fuel type FT	$a_{2j,FT}$ (lb/MW^2h)	$b_{2j,FT}$ (lb/MWh)	$c_{2j,FT}$ (lb/h)	$d_{2j,FT}$ (lb/h)	$e_{2j,FT}$ (MW^{-1})
1	1	0.048	-2.22	60	1.31	0.0569
	2	0.048	-2.22	60	1.31	0.0569
2	1	0.0762	-2.36	100	1.31	0.0569
	2	0.054	-3.14	120	0.91	0.0454
	3	0.085	-1.89	50	0.99	0.0406
3	1	0.0854	-3.08	80	1.31	0.0569
	2	0.0242	-3.06	100	0.65	0.02846
	3	0.031	-2.32	130	0.65	0.02846
4	1	0.0335	-2.11	150	0.65	0.02846
	2	0.425	-4.34	280	0.65	0.02846
	3	0.0322	-4.34	220	0.65	0.02846
5	1	0.0338	-4.28	225	0.65	0.02846
	2	0.0296	-4.18	300	0.5	0.02075
	3	0.0512	-3.34	520	0.5	0.02075
6	1	0.0496	-3.55	510	0.5	0.02075
	2	0.0496	-3.55	510	0.5	0.02075
	3	0.0151	-2.68	220	0.5	0.02075
7	1	0.0151	-2.66	222	0.5	0.02075
	2	0.0151	-2.68	220	0.5	0.02075
	3	0.0151	-2.68	220	0.5	0.02075
8	1	0.0145	-2.22	290	0.5	0.02075
	2	0.0145	-2.22	285	0.5	0.02075
	3	0.0138	-2.26	295	0.5	0.02075
9	1	0.0138	-2.26	295	0.5	0.02075
	2	0.0132	-2.42	310	0.5	0.02075
	3	0.0132	-2.42	310	0.5	0.02075
10	1	1.842	-1.11	360	0.99	0.0406
	2	1.842	-1.11	360	0.99	0.0406
	3	1.842	-1.11	360	0.99	0.0406

Table D.5: B-matrix for multi-fuel system ($\times 10^{-4}$) [330]

$$B_{ij} = \begin{bmatrix} 0.49 & 0.14 & 0.15 & 0.15 & 0.16 & 0.17 & 0.17 & 0.18 & 0.19 & 0.20 \\ 0.14 & 0.16 & 0.16 & 0.16 & 0.17 & 0.15 & 0.15 & 0.16 & 0.18 & 0.18 \\ 0.15 & 0.16 & 0.39 & 0.10 & 0.12 & 0.12 & 0.14 & 0.14 & 0.16 & 0.16 \\ 0.15 & 0.16 & 0.10 & 0.40 & 0.14 & 0.10 & 0.11 & 0.12 & 0.14 & 0.15 \\ 0.16 & 0.17 & 0.12 & 0.14 & 0.35 & 0.11 & 0.13 & 0.13 & 0.15 & 0.16 \\ 0.17 & 0.15 & 0.12 & 0.10 & 0.11 & 0.36 & 0.12 & 0.12 & 0.14 & 0.15 \\ 0.17 & 0.15 & 0.14 & 0.11 & 0.13 & 0.12 & 0.38 & 0.16 & 0.16 & 0.18 \\ 0.18 & 0.16 & 0.14 & 0.12 & 0.13 & 0.12 & 0.16 & 0.40 & 0.15 & 0.16 \\ 0.19 & 0.18 & 0.16 & 0.14 & 0.15 & 0.14 & 0.16 & 0.15 & 0.42 & 0.19 \\ 0.20 & 0.18 & 0.16 & 0.15 & 0.16 & 0.15 & 0.18 & 0.16 & 0.19 & 0.44 \end{bmatrix}$$

$$B_{j0} = [0.0 \ 0.0 \ 0.0 \ 0.0 \ 0.0 \ 0.0 \ 0.0 \ 0.0 \ 0.0 \ 0.0]$$

$$B_{00} = 0.0$$

D.2 SYSTEM DATA FOR TEST POWER SYSTEM-2

A power system having 40-thermal generator power system has been undertaken for

Table D.6: Power generation cost coefficient of forty generators

Generator j	$a1_j$ (\$/MW ² h)	$b1_j$ (\$/MW ^h)	$c1_j$ (\$/h)	$d1_j$ (\$/h)	$e1_j$ (rad/MW)	P_j^{min} (MW)	P_j^{max} (MW)
1	0.0069	6.73	94.705	100	0.084	36	114
2	0.0069	6.73	94.705	100	0.084	36	114
3	0.02028	7.07	309.54	100	0.084	60	120
4	0.00942	8.18	369.03	150	0.063	80	190
5	0.0114	5.35	148.89	120	0.077	47	97
6	0.01142	8.05	222.33	100	0.084	68	140
7	0.00357	8.03	287.71	200	0.042	110	300
8	0.00492	6.99	391.98	200	0.042	135	300
9	0.00573	6.6	455.76	200	0.042	135	300
10	0.00605	12.9	722.82	200	0.042	130	300
11	0.00515	12.9	635.2	200	0.042	94	375
12	0.00569	12.8	654.69	200	0.042	94	375
13	0.00421	12.5	913.4	300	0.035	125	500
14	0.00752	8.84	1760.4	300	0.035	125	500
15	0.00752	8.84	1760.4	300	0.035	125	500
16	0.00752	8.84	1760.4	300	0.035	125	500
17	0.00313	7.97	647.85	300	0.035	220	500
18	0.00313	7.95	649.69	300	0.035	220	500
19	0.00313	7.97	647.83	300	0.035	242	550
20	0.00313	7.97	647.81	300	0.035	242	550
21	0.00298	6.63	785.96	300	0.035	254	550
22	0.00298	6.63	785.96	300	0.035	254	550
23	0.00284	6.66	794.53	300	0.035	254	550
24	0.00284	6.66	794.53	300	0.035	254	550
25	0.00277	7.1	801.32	300	0.035	254	550
26	0.00277	7.1	801.32	300	0.035	254	550
27	0.52124	3.33	1055.1	120	0.077	10	150
28	0.52124	3.33	1055.1	120	0.077	10	150
29	0.52124	3.33	1055.1	120	0.077	10	150
30	0.0114	5.35	148.89	120	0.077	47	97
31	0.0016	6.43	222.92	150	0.063	60	190
32	0.0016	6.43	222.92	150	0.063	60	190
33	0.0016	6.43	222.92	150	0.063	60	190
34	0.0001	8.95	107.87	200	0.042	90	200
35	0.0001	8.62	116.58	200	0.042	90	200
36	0.0001	8.62	116.58	200	0.042	90	200
37	0.0161	5.88	307.45	80	0.098	25	110
38	0.0161	5.88	307.45	80	0.098	25	110
39	0.0161	5.88	307.45	80	0.098	25	110
40	0.00313	7.97	647.83	300	0.035	242	550

simultaneously minimizing fuel cost and gaseous pollutants emission. The data re-

garding operating cost and emission characteristics are given in [330]. The system

Table D.7: Gaseous pollutant's emission coefficient of forty generators

Generator j	a_{2j} (lb/MW^2h)	b_{2j} (lb/MWh)	c_{2j} (lb/h)	d_{2j} (lb/h)	e_{2j} (MW^{-1})
1	0.048	-2.22	60	1.31	0.0569
2	0.048	-2.22	60	1.31	0.0569
3	0.0762	-2.36	100	1.31	0.0569
4	0.054	-3.14	120	0.9142	0.0454
5	0.085	-1.89	50	0.9936	0.0406
6	0.0854	-3.08	80	1.31	0.0569
7	0.0242	-3.06	100	0.655	0.02846
8	0.031	-2.32	130	0.655	0.02846
9	0.0335	-2.11	150	0.655	0.02846
10	0.425	-4.34	280	0.655	0.02846
11	0.0322	-4.34	220	0.655	0.02846
12	0.0338	-4.28	225	0.655	0.02846
13	0.0296	-4.18	300	0.5035	0.02075
14	0.0512	-3.34	520	0.5035	0.02075
15	0.0496	-3.55	510	0.5035	0.02075
16	0.0496	-3.55	510	0.5035	0.02075
17	0.0151	-2.68	220	0.5035	0.02075
18	0.0151	-2.66	222	0.5035	0.02075
19	0.0151	-2.68	220	0.5035	0.02075
20	0.0151	-2.68	220	0.5035	0.02075
21	0.0145	-2.22	290	0.5035	0.02075
22	0.0145	-2.22	285	0.5035	0.02075
23	0.0138	-2.26	295	0.5035	0.02075
24	0.0138	-2.26	295	0.5035	0.02075
25	0.0132	-2.42	310	0.5035	0.02075
26	0.0132	-2.42	310	0.5035	0.02075
27	1.842	-1.11	360	0.9936	0.0406
28	1.842	-1.11	360	0.9936	0.0406
29	1.842	-1.11	360	0.9936	0.0406
30	0.085	-1.89	50	0.9936	0.0406
31	0.0121	-2.08	80	0.9142	0.0454
32	0.0121	-2.08	80	0.9142	0.0454
33	0.0121	-2.08	80	0.9142	0.0454
34	0.0012	-3.48	65	0.655	0.02846
35	0.0012	-3.24	70	0.655	0.02846
36	0.0012	-3.24	70	0.655	0.02846
37	0.095	-1.98	100	1.42	0.0677
38	0.095	-1.98	100	1.42	0.0677
39	0.095	-1.98	100	1.42	0.0677
40	0.0151	-2.68	220	0.5035	0.02075

has power demand of 10500 MW. The same data is reproduced in Tables D.6 and D.7.

D.3 SYSTEM DATA FOR TEST POWER SYSTEM-3

Test power system-3 considered for study is Korean power system [321] having 140 thermal power generating units with ramp-rate limits and power demand of 49342 MW. This system has been analyzed with two options, *i.e.*, (i) neglecting valve point effects and POZ; and (ii) considering valve-point loading effects and POZ. Fuel cost characteristics coefficients along with minimum, maximum and ramp rate limits are given in Table D.8. The gaseous pollutant's emission coefficients are obtained by extending the emission coefficients of Test system-2 (Appendix D.7) and are tabulated in Table D.9. Data regarding prohibited operating zones is tabulated in Table D.10.

Table D.8: Power generation cost coefficients of Korean power system

Unit j	$a1_j$ (\$/MW ² h)	$b1_j$ (\$/MW/h)	$c1_j$ (\$/h)	$d1_j$ (\$/h)	$e1_j$ (rad/MW)	P_j^{min} (MW)	P_j^{max} (MW)	UR_j (MW/h)	DR_j (MW/h)	P_j^0 (MW)
1	0.032888	61.242	1220.645	0	0	71	119	30	120	98.4
2	0.00828	41.095	1315.118	0	0	120	189	30	120	134
3	0.003849	46.31	874.288	0	0	125	190	60	60	141.5
4	0.003849	46.31	874.288	0	0	125	190	60	60	183.3
5	0.042468	54.242	1976.469	700	0.08	90	190	150	150	125
6	0.014992	61.215	1338.087	0	0	90	190	150	150	91.3
7	0.007039	11.791	1818.299	0	0	280	490	180	300	401.1
8	0.003079	15.055	1133.978	0	0	280	490	180	300	329.5
9	0.005063	13.226	1320.636	0	0	260	496	300	510	386.1
10	0.005063	13.226	1320.636	600	0.055	260	496	300	510	427.3
11	0.005063	13.226	1320.636	0	0	260	496	300	510	412.2
12	0.003552	14.498	1106.539	0	0	260	496	300	510	370.1
13	0.003901	14.651	1176.504	0	0	260	506	600	600	301.8
14	0.003901	14.651	1176.504	0	0	260	509	600	600	368
15	0.003901	14.651	1176.504	800	0.06	260	506	600	600	301.9
16	0.003901	14.651	1176.504	0	0	260	505	600	600	476.4
17	0.002393	15.669	1017.406	0	0	260	506	600	600	283.1
18	0.002393	15.669	1017.406	0	0	260	506	600	600	414.1
19	0.003684	14.656	1229.131	0	0	260	505	600	600	328
20	0.003684	14.656	1229.131	0	0	260	505	600	600	389.4
21	0.003684	14.656	1229.131	0	0	260	505	600	600	354.7
22	0.003684	14.656	1229.131	600	0.05	260	505	600	600	262
23	0.004004	14.378	1267.894	0	0	260	505	600	600	461.5
24	0.003684	14.656	1229.131	0	0	260	505	600	600	371.6
25	0.001619	16.261	975.926	0	0	280	537	300	300	462.6
26	0.005093	13.362	1532.093	0	0	280	537	300	300	379.2
27	0.000993	17.203	641.989	0	0	280	549	360	360	530.8
28	0.000993	17.203	641.989	0	0	280	549	360	360	391.9
29	0.002473	15.274	911.533	0	0	260	501	180	180	480.1
30	0.002547	15.212	910.533	0	0	260	501	180	180	319
31	0.003542	15.033	1074.81	0	0	260	506	600	600	329.5
32	0.003542	15.033	1074.81	0	0	260	506	600	600	333.8
33	0.003542	15.033	1074.81	600	0.043	260	506	600	600	390
34	0.003542	15.033	1074.81	0	0	260	506	600	600	432
35	0.003132	13.992	1278.46	0	0	260	500	660	660	402
36	0.001323	15.679	861.742	0	0	260	500	900	900	428
37	0.00295	16.542	408.834	0	0	120	241	180	180	178.4
38	0.00295	16.542	408.834	0	0	120	241	180	180	194.1
39	0.000991	16.518	1288.815	0	0	423	774	600	600	474
40	0.001581	15.815	1436.251	600	0.043	423	769	600	600	609.8
41	0.90236	75.464	669.988	0	0	3	19	210	210	17.8
42	0.110295	129.544	134.544	0	0	3	28	366	366	6.9
43	0.024493	56.613	3427.912	0	0	160	250	702	702	224.3
44	0.029156	54.451	3751.772	0	0	160	250	702	702	210
45	0.024667	54.736	3918.78	0	0	160	250	702	702	212

Continued on next page

Table D.8 – Continued from previous page

Generator j	$a1_j$ (\$/MW ² h)	$b1_j$ (\$/MW/h)	$c1_j$ (\$/h)	$d1_j$ (\$/h)	$e1_j$ (rad/MW)	P_j^{min} (MW)	P_j^{max} (MW)	UR_j (MW/h)	DR_j (MW/h)	P_j^0 (MW)
46	0.016517	58.034	3379.58	0	0	160	250	702	702	200.8
47	0.026584	55.981	3345.296	0	0	160	250	702	702	220
48	0.00754	61.52	3138.754	0	0	160	250	702	702	232.9
49	0.01643	58.635	3453.05	0	0	160	250	702	702	168
50	0.045934	44.647	5119.3	0	0	160	250	702	702	208.4
51	0.000044	71.584	1898.415	0	0	165	504	1350	1350	443.9
52	0.000044	71.584	1898.415	1100	0.043	165	504	1350	1350	426
53	0.000044	71.584	1898.415	0	0	165	504	1350	1350	434.1
54	0.000044	71.584	1898.415	0	0	165	504	1350	1350	402.5
55	0.002528	85.12	2473.39	0	0	180	471	1350	1350	357.4
56	0.000131	87.682	2781.705	0	0	180	561	720	720	423
57	0.010372	69.532	5515.508	0	0	103	341	720	720	220
58	0.007627	78.339	3478.3	0	0	198	617	2700	2700	369.4
59	0.012464	58.172	6240.909	0	0	100	312	1500	1500	273.5
60	0.039441	46.636	9960.11	0	0	153	471	1656	1656	336
61	0.007278	76.947	3671.997	0	0	163	500	2160	2160	432
62	0.000044	80.761	1837.383	0	0	95	302	900	900	220
63	0.000044	70.136	3108.395	0	0	160	511	1200	1200	410.6
64	0.000044	70.136	3108.395	0	0	160	511	1200	1200	422.7
65	0.018827	49.84	7095.484	0	0	196	490	1014	1014	351
66	0.010852	65.404	3392.732	0	0	196	490	1014	1014	296
67	0.018827	49.84	7095.484	0	0	196	490	1014	1014	411.1
68	0.018827	49.84	7095.484	0	0	196	490	1014	1014	263.2
69	0.03456	66.465	4288.32	0	0	130	432	1350	1350	370.3
70	0.08154	22.941	13813	1200	0.03	130	432	1350	1350	418.7
71	0.023534	64.314	4435.493	0	0	137	455	1350	1350	409.6
72	0.035475	45.017	9750.75	1000	0.05	137	455	1350	1350	412
73	0.000915	70.644	1042.366	0	0	195	541	780	780	423.2
74	0.000044	70.959	1159.895	0	0	175	536	1650	1650	428
75	0.000044	70.959	1159.895	0	0	175	540	1650	1650	436
76	0.001307	70.302	1303.99	0	0	175	538	1650	1650	428
77	0.000392	70.662	1156.193	0	0	175	540	1650	1650	425
78	0.000087	71.101	2118.968	0	0	330	574	1620	1620	497.2
79	0.000521	37.854	779.519	0	0	160	531	1482	1482	510
80	0.000498	37.768	829.888	0	0	160	531	1482	1482	470
81	0.001046	67.983	2333.69	0	0	200	542	1668	1668	464.1
82	0.13205	77.838	2028.954	0	0	56	132	120	120	118.1
83	0.096968	63.671	4412.017	0	0	115	245	180	180	141.3
84	0.054868	79.458	2982.219	1000	0.05	115	245	120	180	132
85	0.054868	79.458	2982.219	0	0	115	245	120	180	135
86	0.014382	93.966	3174.939	0	0	207	307	120	180	252
87	0.013161	94.723	3218.359	0	0	207	307	120	180	221
88	0.016033	66.919	3723.822	0	0	175	345	318	318	245.9
89	0.013653	68.185	3551.405	0	0	175	345	318	318	247.9
90	0.028148	60.821	4322.615	0	0	175	345	318	318	183.6
91	0.01347	68.551	3493.739	0	0	175	345	318	318	288
92	0.000064	2.842	226.799	0	0	360	580	18	18	557.4
93	0.000252	2.946	382.932	0	0	415	645	18	18	529.5
94	0.000022	3.096	156.987	0	0	795	984	36	36	800.8
95	0.000022	3.04	154.484	0	0	795	978	36	36	801.5
96	0.000203	1.709	332.834	0	0	578	682	138	204	582.7
97	0.000198	1.668	326.599	0	0	615	720	144	216	680.7
98	0.000215	1.789	345.306	0	0	612	718	144	216	670.7
99	0.000218	1.815	350.372	0	0	612	720	144	216	651.7
100	0.000193	2.726	370.377	0	0	758	964	48	48	921
101	0.000197	2.732	367.067	0	0	755	958	48	48	916.8
102	0.000324	2.651	124.875	0	0	750	1007	36	54	911.9
103	0.000344	2.798	130.785	0	0	750	1006	36	54	898
104	0.00069	1.595	878.746	0	0	713	1013	30	30	905
105	0.00065	1.503	827.959	0	0	718	1020	30	30	846.5
106	0.000233	2.425	432.007	0	0	791	954	30	30	850.9
107	0.000239	2.499	445.606	0	0	786	952	30	30	843.7
108	0.000261	2.674	467.223	0	0	795	1006	36	36	841.4
109	0.000259	2.692	475.94	0	0	795	1013	36	36	835.7
110	0.000707	1.633	899.462	0	0	795	1021	36	36	828.8
111	0.000786	1.816	1000.367	0	0	795	1015	36	36	846
112	0.014355	89.83	1269.132	0	0	94	203	120	120	179
113	0.014355	89.83	1269.132	0	0	94	203	120	120	120.8
114	0.014355	89.83	1269.132	0	0	94	203	120	120	121
115	0.030266	64.125	4965.124	0	0	244	379	480	480	317.4
116	0.030266	64.125	4965.124	0	0	244	379	480	480	318.4

Continued on next page

Table D.8 – Continued from previous page

Generator j	$a1_j$ (\$/MW ² h)	$b1_j$ (\$/MW/h)	$c1_j$ (\$/h)	$d1_j$ (\$/h)	$e1_j$ (rad/MW)	P_j^{min} (MW)	P_j^{max} (MW)	UR_j (MW/h)	DR_j (MW/h)	P_j^0 (MW)
117	0.030266	64.125	4965.124	0	0	244	379	480	480	335.8
118	0.024027	76.129	2243.185	0	0	95	190	240	240	151
119	0.00158	81.805	2290.381	600	0.07	95	189	240	240	129.5
120	0.022095	81.14	1681.533	0	0	116	194	120	120	130
121	0.07681	46.665	6743.302	1200	0.043	175	321	180	180	218.9
122	0.953443	78.412	394.398	0	0	2	19	90	90	5.4
123	0.000044	112.088	1243.165	0	0	4	59	90	90	45
124	0.072468	90.871	1454.74	0	0	15	83	300	300	20
125	0.000448	97.116	1011.051	0	0	9	53	162	162	16.3
126	0.599112	83.244	909.269	0	0	12	37	114	114	20
127	0.244706	95.665	689.378	0	0	10	34	120	120	22.1
128	0.000042	91.202	1443.792	0	0	112	373	1080	1080	125
129	0.085145	104.501	535.553	0	0	4	20	60	60	10
130	0.524718	83.015	617.734	0	0	5	38	66	66	13
131	0.176515	127.795	90.966	0	0	5	19	12	6	7.5
132	0.063414	77.929	974.447	0	0	50	98	300	300	53.2
133	2.740485	92.779	263.81	0	0	5	10	6	6	6.4
134	0.112438	80.95	1335.594	0	0	42	74	60	60	69.1
135	0.041529	89.073	1033.871	0	0	42	74	60	60	49.9
136	0.000911	161.288	1391.325	0	0	41	105	528	528	91
137	0.005245	161.829	4477.11	0	0	17	51	300	300	41
138	0.234787	84.972	57.794	0	0	7	19	18	30	13.7
139	0.234787	84.972	57.794	0	0	7	19	18	30	7.4
140	1.111878	16.087	1258.437	0	0	26	40	72	120	28.6

Table D.9: Gaseous pollutant's emission coefficient of Korean system

Generator j	$a2_j$ (lb/MW ² h)	$b2_j$ (lb/MW/h)	$c2_j$ (lb/h)	$d2_j$ (lb/h)	$e2_j$ (MW ⁻¹)
1	0.048	-2.22	60	1.31	0.0569
2	0.048	-2.22	60	1.31	0.0569
3	0.0762	-2.36	100	1.31	0.0569
4	0.054	-3.14	120	0.91	0.0454
5	0.085	-1.89	50	0.99	0.0406
6	0.0854	-3.08	80	1.31	0.0569
7	0.0242	-3.06	100	0.65	0.02846
8	0.031	-2.32	130	0.65	0.02846
9	0.0335	-2.11	150	0.65	0.02846
10	0.425	-4.34	280	0.65	0.02846
11	0.0322	-4.34	220	0.65	0.02846
12	0.0338	-4.28	225	0.65	0.02846
13	0.0296	-4.18	300	0.5	0.02075
14	0.0512	-3.34	520	0.5	0.02075
15	0.0496	-3.55	510	0.5	0.02075
16	0.0496	-3.55	510	0.5	0.02075
17	0.0151	-2.68	220	0.5	0.02075
18	0.0151	-2.66	222	0.5	0.02075
19	0.0151	-2.68	220	0.5	0.02075
20	0.0151	-2.68	220	0.5	0.02075
21	0.0145	-2.22	290	0.5	0.02075
22	0.0145	-2.22	285	0.5	0.02075
23	0.0138	-2.26	295	0.5	0.02075
24	0.0138	-2.26	295	0.5	0.02075
25	0.0132	-2.42	310	0.5	0.02075
26	0.0132	-2.42	310	0.5	0.02075
27	1.842	-1.11	360	0.99	0.0406

Continued on next page

Table D.9 – *Continued from previous page*

Generator j	a_{2j} (lb/MW^2h)	b_{2j} (lb/MWh)	c_{2j} (lb/h)	d_{2j} (lb/h)	e_{2j} (MW^{-1})
28	1.842	-1.11	360	0.99	0.0406
29	1.842	-1.11	360	0.99	0.0406
30	0.085	-1.89	50	0.99	0.0406
31	0.0121	-2.08	80	0.91	0.0454
32	0.0121	-2.08	80	0.91	0.0454
33	0.0121	-2.08	80	0.91	0.0454
34	0.0012	-3.48	65	0.65	0.02846
35	0.0012	-3.24	70	0.65	0.02846
36	0.0012	-3.24	70	0.65	0.02846
37	0.095	-1.98	100	1.42	0.0677
38	0.095	-1.98	100	1.42	0.0677
39	0.095	-1.98	100	1.42	0.0677
40	0.0151	-2.68	220	0.5	0.02075
41	0.048	-2.22	60	1.31	0.0569
42	0.048	-2.22	60	1.31	0.0569
43	0.0762	-2.36	100	1.31	0.0569
44	0.054	-3.14	120	0.91	0.0454
45	0.085	-1.89	50	0.99	0.0406
46	0.0854	-3.08	80	1.31	0.0569
47	0.0242	-3.06	100	0.65	0.02846
48	0.031	-2.32	130	0.65	0.02846
49	0.0335	-2.11	150	0.65	0.02846
50	0.425	-4.34	280	0.65	0.02846
51	0.0322	-4.34	220	0.65	0.02846
52	0.0338	-4.28	225	0.65	0.02846
53	0.0296	-4.18	300	0.5	0.02075
54	0.0512	-3.34	520	0.5	0.02075
55	0.0496	-3.55	510	0.5	0.02075
56	0.0496	-3.55	510	0.5	0.02075
57	0.0151	-2.68	220	0.5	0.02075
58	0.0151	-2.66	222	0.5	0.02075
59	0.0151	-2.68	220	0.5	0.02075
60	0.0151	-2.68	220	0.5	0.02075
61	0.0145	-2.22	290	0.5	0.02075
62	0.0145	-2.22	285	0.5	0.02075
63	0.0138	-2.26	295	0.5	0.02075
64	0.0138	-2.26	295	0.5	0.02075
65	0.0132	-2.42	310	0.5	0.02075
66	0.0132	-2.42	310	0.5	0.02075
67	1.842	-1.11	360	0.99	0.0406
68	1.842	-1.11	360	0.99	0.0406
69	1.842	-1.11	360	0.99	0.0406
70	0.085	-1.89	50	0.99	0.0406
71	0.0121	-2.08	80	0.91	0.0454
72	0.0121	-2.08	80	0.91	0.0454
73	0.0121	-2.08	80	0.91	0.0454
74	0.0012	-3.48	65	0.65	0.02846
75	0.0012	-3.24	70	0.65	0.02846
76	0.0012	-3.24	70	0.65	0.02846

Continued on next page

Table D.9 – *Continued from previous page*

Generator j	a_{2j} (lb/MW^2h)	b_{2j} (lb/MWh)	c_{2j} (lb/h)	d_{2j} (lb/h)	e_{2j} (MW^{-1})
77	0.095	-1.98	100	1.42	0.0677
78	0.095	-1.98	100	1.42	0.0677
79	0.095	-1.98	100	1.42	0.0677
80	0.0151	-2.68	220	0.5	0.02075
81	0.048	-2.22	60	1.31	0.0569
82	0.048	-2.22	60	1.31	0.0569
83	0.0762	-2.36	100	1.31	0.0569
84	0.054	-3.14	120	0.91	0.0454
85	0.085	-1.89	50	0.99	0.0406
86	0.0854	-3.08	80	1.31	0.0569
87	0.0242	-3.06	100	0.65	0.02846
88	0.031	-2.32	130	0.65	0.02846
89	0.0335	-2.11	150	0.65	0.02846
90	0.425	-4.34	280	0.65	0.02846
91	0.0322	-4.34	220	0.65	0.02846
92	0.0338	-4.28	225	0.65	0.02846
93	0.0296	-4.18	300	0.5	0.02075
94	0.0512	-3.34	520	0.5	0.02075
95	0.0496	-3.55	510	0.5	0.02075
96	0.0496	-3.55	510	0.5	0.02075
97	0.0151	-2.68	220	0.5	0.02075
98	0.0151	-2.66	222	0.5	0.02075
99	0.0151	-2.68	220	0.5	0.02075
100	0.0151	-2.68	220	0.5	0.02075
101	0.0145	-2.22	290	0.5	0.02075
102	0.0145	-2.22	285	0.5	0.02075
103	0.0138	-2.26	295	0.5	0.02075
104	0.0138	-2.26	295	0.5	0.02075
105	0.0132	-2.42	310	0.5	0.02075
106	0.0132	-2.42	310	0.5	0.02075
107	1.842	-1.11	360	0.99	0.0406
108	1.842	-1.11	360	0.99	0.0406
109	1.842	-1.11	360	0.99	0.0406
110	0.085	-1.89	50	0.99	0.0406
111	0.0121	-2.08	80	0.91	0.0454
112	0.0121	-2.08	80	0.91	0.0454
113	0.0121	-2.08	80	0.91	0.0454
114	0.0012	-3.48	65	0.65	0.02846
115	0.0012	-3.24	70	0.65	0.02846
116	0.0012	-3.24	70	0.65	0.02846
117	0.095	-1.98	100	1.42	0.0677
118	0.095	-1.98	100	1.42	0.0677
119	0.095	-1.98	100	1.42	0.0677
120	0.0151	-2.68	220	0.5	0.02075
121	0.048	-2.22	60	1.31	0.0569
122	0.048	-2.22	60	1.31	0.0569
123	0.0762	-2.36	100	1.31	0.0569
124	0.054	-3.14	120	0.91	0.0454
125	0.085	-1.89	50	0.99	0.0406

Continued on next page

Table D.9 – *Continued from previous page*

Generator j	$a2_j$ (lb/MW^2h)	$b2_j$ (lb/MWh)	$c2_j$ (lb/h)	$d2_j$ (lb/h)	$e2_j$ (MW^{-1})
126	0.0854	-3.08	80	1.31	0.0569
127	0.0242	-3.06	100	0.65	0.02846
128	0.031	-2.32	130	0.65	0.02846
129	0.0335	-2.11	150	0.65	0.02846
130	0.425	-4.34	280	0.65	0.02846
131	0.0322	-4.34	220	0.65	0.02846
132	0.0338	-4.28	225	0.65	0.02846
133	0.0296	-4.18	300	0.5	0.02075
134	0.0512	-3.34	520	0.5	0.02075
135	0.0496	-3.55	510	0.5	0.02075
136	0.0496	-3.55	510	0.5	0.02075
137	0.0151	-2.68	220	0.5	0.02075
138	0.0151	-2.66	222	0.5	0.02075
139	0.0151	-2.68	220	0.5	0.02075
140	0.0151	-2.68	220	0.5	0.02075

Table D.10: Prohibited operating zones data for Korean power system's thermal generating units

Unit j	P_j^U (MW)	P_j^L (MW)	P_j^U (MW)	P_j^L (MW)	P_j^U (MW)	P_j^L (MW)
8	250	280	305	335	420	450
32	220	250	320	350	390	420
74	230	255	365	395	430	455
136	50	75	85	95	–	–