LIBRARY
University of Bahrain

UNIVERSITY OF BAHRAIN

College of Engineering

Department of Electrical and Electronics Engineering

Optimal Relays Settings by Using Biogeography Based Optimization "BBO" Technique

A Thesis Submitted in Partial Fulfillment of the Requirements for the Master of Science Degree in Electrical Engineering

Submitted by

Ali Ridha A.Ali Mohammed Alroomi

20012006

Supervised by

Dr. Jawad Hassan Talaq
Associate Professor

Dr. Fadhel Abbas Albasri
Assistant Professor

University of Bahrain

Kingdom of Bahrain

December, 2014

DECLARATION

I declare that this work is the result of my own investigation and that it has not already been accepted in substance for any degree, nor is it being currently submitted for any degree.

Signed : ..... Date : December 25,2014

Ali R. Alroomi

University of Bahrain LIBRARY

The Thesis defense committee considers the Thesis titled :

"OPTIMAL RELAYS SETTINGS BY USING BIOGEOGRAPHY BASED OPTIMIZATION "BBO" TECHNIQUE"

Submitted by Ali Ridha A.Ali Mohammed Alroomi to the Department of electrical and electronic engineering satisfactory and acceptable for the Master of Science degree in Electrical Engineering.

Thesis Defense Committee:

.....

Dr. Jawad Hassan Tallaq — Department of Electrical and Electronic Engineering
Supervisor

.....

Dr. Fadhel Abbas Albasri — Department of Electrical and Electronic Engineering
Co-Supervisor

.....

Prof. Mohamed A. El-Sayed — Electrical Engineering Department
External Examiner — Kuwait University

.....

Prof. Khaled Zehar — Department of Electrical and Electronic Engineering
Internal Examiner — University of Bahrain

Abstract

Optimal coordination of directional overcurrent relays (DOCRs) is a critical issue in electrical power engineering. This practical problem is highly constrained and non-linear optimization problem. The operating time of each relay depends on two independent variables called plug setting (*PS*) and time multiplier setting (*TMS*). The sum of all the operating times of these DOCRs, when work as primary relays, is the objective function of such a problem. As the network becomes larger and more complex, the number of relays will increase, and thus finding the optimal solution becomes very hard. Since 1988, many numerical techniques are employed. However, still there are many open areas that are not covered in the literature.

This thesis has multi-contributions; it uses a new population-based evolutionary algorithm called biogeography-based optimization (BBO), evaluates the performance of ten types of constraint-handling techniques, proposes a new sub-algorithm to avoid the infeasibility condition due to selectivity constraint, and presents a new hybridization between the BBO algorithm and the standard linear programming algorithm (BBO-LP). The performance of the proposed technique is evaluated using different test systems, and the results are compared with the performance of the other optimization methods presented in the literature .The results show that the hybrid model with the proposed modifications outperforms many other optimizations presented in the literature in terms of feasibility, quality and processing speed.

Moreover, this thesis presents an extensive study on the performance of the original forms of the BBO algorithm by making a comparison between them through 23 benchmark functions with different dimensions and complexities. Furthermore, this study also tries to solve the root problem of the original BBO forms by tackling their exploration weakness instead of solving its effects by complicated ways. Two modifications are proposed; modifying the probabilistic selection process and minimizing the range of the clear duplication process. The proposed essential modifications on BBO (BBO-EM) are extensively tested on 120 test functions with different dimensions and complexities, and then used to create four advanced BBO models based on updated search space, and respectively called bounds single updater (BSU-BBO), bounds multi updater (BMU-BBO), iterative search space (ISS-BBO), and adaptive variable bounds (AVB-BBO). Finally, the optimal coordination of the 3-bus system is solved again by ISS-BBO.

Also, different time-current characteristic curves (TCCCs) are extensively studied on the 15-bus system.

Acknowledgements

I begin in the name of ALLAH, the Gracious, the Merciful .. The most Great who create the Universe and sustaining it .. For him is all the gratitude and praise. Thank God for the wisdom and perseverance that he has been bestowed upon me during this research project, and indeed, throughout my life.

Dr. Fadhel A. Al-Basri and Dr. Jawad H. Talaq, for making this research possible. Their continuous support, advice, patience, motivation, enthusiasm, and immense knowledge, as well as their pain-staking effort in proof reading the drafts, are greatly appreciated. Their valuable guidance helped me in all the time of research and writing of this thesis. Their great hints and ideas open lots of doors for my future researches. Thanks Drs.

Besides my supervisors, I would like to thank Prof. Khalid Al-Zehar and Dr. Raida Al-Alawi who became as a problem solver for me if I face anything at anytime.

My sincere thanks also go to all other doctors who taught me during my master program. Thank you Prof. Shoukry Hassan, Dr. Maamar Taleb, Dr. Hassan Elragal, Dr. Saied Ali Al-Mousawi and Dr. Ebrahim A. Rahman.

I thank my close friends for their long discussions and conversations about many of close and far related engineering topics.

Last but not least; I would like to thank my family, especially my parents: My father who dedicated his life to make me happy, my mother who became as a candle that lights my full life. Many thanks goes to my wife and the rest of my family.

Dedication

I dedicate this thesis to:

My best father and mother in this world ...

My honey son "Ridha" ...

My patient wife ...

My best friends "Zakariya" and "S. Ahmed" ...

This is for you

Table of Contents

Subject	Page
Declaration	**i**
Abstract	**ii**
Acknowledgements	**iii**
Dedication	**iv**
Table of Contents	**v**
List of Tables	**ix**
List of Figures	**xii**
List of Appendices	**xiv**
List of Abbreviations	**xv**
List of Symbols	**xviii**
Chapter 1 Introduction	
1.1 Literature Review	**3**
1.2 Motivation and Objectives	**6**
1.3 Contributions	**7**
1.4 Thesis Organization	**9**
1.5 Publications	**9**
Chapter 2 Mathematical Optimization	
2.1 Optimal Problem Formulation	**12**
2.1.1 The Standard Format	13
2.2 Types of Optimization Techniques	**16**
2.2.1 Traditional Optimization Algorithms	16
2.2.2 Non-Traditional Optimization Algorithms	18
2.2.3 Hybrid Optimization Algorithms	21
2.3 Constraint-Handling Techniques	**22**
2.3.1 Random Search Method	26
2.3.2 Constant Penalty Function	27
2.3.3 Binary Static Penalty Function	28
2.3.4 The Superiority of Feasible Points (SFP) - Type I	29
2.3.5 The Superiority of Feasible Points (SFP) - Type II	29
2.3.6 The Eclectic Evolutionary Algorithm	30
2.3.7 Typical Dynamic Penalty Function	30

2.3.8 Exponential Dynamic Penalty Function	31
2.3.9 Adaptive Multiplication Penalty Function	32
2.3.10 Self-Adaptive Penalty Function (SAPF)	33
2.4 Performance Criteria for Evaluating Algorithms	**35**
Chapter 3 Fundamentals of Power System Protection	
3.1 Faults Classifications	**40**
3.2 Protection System	**46**
3.3 Zones of Protection	**52**
3.4 Primary and Backup Protection	**53**
3.5 Performance and Design Criteria	**54**
3.5.1 Reliability	54
3.5.2 Sensitivity	54
3.5.3 Speed	55
3.5.4 Selectivity	55
3.5.5 Performance versus Economics	55
3.5.6 Adequateness	55
3.5.7 Simplicity	56
3.6 Overcurrent Protective Relays	**56**
3.7 Directional Overcurrent Protective Relays	**61**
3.8 Coordination of Directional Overcurrent Relays	**64**
3.9 Case Study: Is the Coordination of DOCRs an Iterative Problem?	**65**
3.9.1 Coordination Problem of a Radial System	65
3.9.2 Coordination Problem of a Ring System	66
3.9.3 Conclusion of Case Study	75
3.10 General Optimal Coordination of DOCRs	**76**
3.10.1 Constructing the Power System Network	77
3.10.2 Power Flow Analysis	78
3.10.3 Primary/Backup Pairs Identification	79
3.10.4 Short-Circuit Analysis	86
3.10.5 Applying Optimization Techniques	88
3.10.6 Adaptive Optimal Coordination	94
Chapter 4 Biogeography-Based Optimization	
4.1 Literature Review	**96**

4.2 The Theory of Island Biogeography	**101**
4.3 Biogeography-Based Optimization (BBO)	**112**
4.3.1 Migration Stage	113
4.3.2 Mutation Stage	117
4.3.3 Clear Duplication Stage	119
4.3.4 The Overall BBO Algorithm	120
4.3.5 Performance Comparison of the four Original BBO Forms	121
4.4 Biogeography Based Optimization with Essential Modifications (BBO-EM)	**130**
4.4.1 The Proposed Essential Modifications on BBO	131
4.4.2 Performance Comparison Between BBO and BBO-EM	134
4.4.3 Conclusions and Suggestions	141
4.5 Iterative Variable Bounds – Biogeography Based Optimization (IVB-BBO)	**142**
4.5.1 The Proposed Modifications on BBO	143
4.5.2 Comparison Between ISS-BBO and Other Models	156
4.5.3 conclusions and Suggestions	164
4.6 Adaptive Variable Bounds - Biogeography Based Optimization (AVB-BBO)	**165**
4.6.1 Comparison Between AVB and Iterative Models	166
4.6.2 Conclusions and Suggestions	168
Chapter 5 Optimal Coordination of DOCRs Using BBO	
5.1 The Proposed Modifications	**172**
5.1.1 Constraint-Handling Techniques	173
5.1.2 Avoiding Infeasibility Condition	173
5.1.3 Hybridization with Linear Programming	175
5.2 Performance Comparison and Discussions	176
5.2.1 Test Case 1: 3-Bus System	178
5.2.2 Test Case 2: 4-Bus System with Normal Configuration	185
5.2.3 Test Case 3: 8-Bus System with Normal Configuration	189
5.2.4 Test Case 4: 9-Bus System with Normal Configuration	191
5.2.5 Test Case 5: 15-Bus System	195
Chapter 6 Conclusion and Future Works	
6.1 Suggestions	**216**

6.2 Scope of Future Works	**217**
6.3 Applications and Soft Tools	**218**
References	**219**
Appendix A	
Appendix B	
Appendix C	
Appendix D	
Appendix E	
Arabic Abstract	

List of Tables

No.	Table	Page
2.1	Types of Problem	12
3.1	The Probability of Faults Occurrence Corresponding to Their Types	44
3.2	The Probability of Faults Occurrence on Different Elements of the Power Systems	45
3.3	Standard Values of Inverse Time Overcurrent Relays' Constants	60
3.4	Data of Case Study "3-Bus System"	68
3.5	Bus Types and Their Known/Unknown Variables for Power Flow Analysis	79
3.6	Primary/Backup Relay Pairs of a Simple 5-Bus System	81
4.1	Performance Comparison of the Four Original BBO Models	124
4.2	Normalized CPU Time of the Four Original BBO Models	125
4.3	The Overall Performance of PMB and SPMB with/without Mutation Stage	126
4.4	Performance Test I – Different Dimensions	126
4.5	Performance Test II – Different Population Size	127
4.6	Performance Test III – Different Side Constraints	127
4.7	The Simulation Parameters for BBO-EM	135
4.8	The Required Generation Number for Various *n*-Dimensional Problems	135
4.9	Performance Comparison Between BBO and BBO-EM Models (n = 2)	137
4.10	Performance Comparison Between BBO and BBO-EM Models (n ≠ 2)	138
4.11	The Overall Performance of BBO and BBO-EM with/without Mutation Stage	140
4.12	Normalized CPU Time of the BBO and BBO-EM Models	140
4.13	Performance Comparison Between the Models of PMBBBO, BSU-BBO, BMU-BBO and ISS-BBO	158
4.14	Normalized CPU Time of the BBO, BSU-BBO, BMUBBO and ISS-BBO Models	159
4.15	Performance Comparison of ISS Model with Different Values of *Elit* and *SF*	160
4.16	Comparison of Results over 30 Trails of ISS-BBO with Linear Migration Rates and BBO with Sinusoidal and Generalized Sinusoidal Migration Rates	162
4.17	Performance Comparison Between the Models of AVBBBO,	

	BSU-BBO, BMU-BBO and ISS-BBO	169
5.1	The BBO Parameters for Each Test Case	178
5.2	Cost Function Comparison of the 10 Constraint-Handling Techniques for the Test Case 1 - Scenario 1	182
5.3	Cost Function Comparison of the 10 Constraint-Handling Techniques for the Test Case 1 - Scenario 2	183
5.4	Performance Criteria of the 10 Constraint-Handling Techniques for the Test Case 1 - Scenario 2	184
5.5	The Value of the Binary Static Penalty Multiplier for Each Network System	184
5.6	Comparison Between BBO and BBO-LP for the Test Case 1 - Scenario 2	184
5.7	Comparison with Seeker Algorithm for the Test Case 1 - Scenario 2	184
5.8	Comparison Between BBO and ISS-BBO for the Test Case 1 - Scenario 3	185
5.9	Comparison Between BBO and BBO-LP for the Test Case 2	188
5.10	Feasibility Comparison Between BBO and BBO-LP for the Test Case 2	188
5.11	Literature Results for the Test Case 2	188
5.12	Comparison Between BBO and BBO-LP for the Test Case 3	190
5.13	Feasibility Comparison with Other Paper Result for the Test Case 3	190
5.14	Comparison Between BBO and BBO-LP for the Test Case 4	193
5.15	Feasibility Comparison with Other Paper Result for the Test Case 4	194
5.16	BBO Performance within 500 Generations for the Test Case 4	194
5.17	Overall Comparison with Other Paper Result for the Test Case 5 - Scenario 1	197
5.18	Comparison Between BBO and BBO-LP for the Test Case 5 - Scenario 1	197
5.19	Feasibility Comparison of MINLP and Seeker Algorithms for the Test case 5 - Scenario 1	198
5.2	Feasibility Comparison of BBO and BBO-LP for the Test case 5 - Scenario 1	199
5.21	Performance Quality of BBO-LP under Different Parameters for the Test Case 5 - Scenario 1	200
5.22	Optimal Coordination of the Test Case 5 with All Standards - Scenario 2	205
5.23	Optimal Coordination of the Test Case 5 with North American Standards - Scenario 2	206
5.24	Optimal Coordination of the Test Case 5 with European	

	Standards - Scenario 2	207
5.25	Optimal Coordination of the Test Case 5 with IEEE and IEC Standards - Scenario 2	208
5.26	Optimal Coordination of the Test Case 5 with IEEE Standards - Scenario 2	209
5.27	Optimal Coordination of the Test Case 5 with IEC Standards - Scenario 2	210
5.28	Performance Comparison Between DOCRs with Different TCCCs and DOCRs with only IDMT Model – Test Case 5	211

List of Figures

No.	Figure	Page
2.1	Optimal Design Flowchart	13
2.2	Constraint-Handling Techniques	23
3.1	General Representation of Complete Power System Cycle	39
3.2	Cause and Effect Diagram of Different Faults Scenarios	41
3.3	The Possibilities of Active Short Circuit Fault	44
3.4	Illustrated 1Ø Protection System	46
3.5	The General 3Ø Protection System	49
3.6	The General Hardware of a Numerical Relay	51
3.7	Zones of Protection	53
3.8	Time-Current Characteristic Curve of Instantaneous Overcurrent Relay	57
3.9	Illustration of Two Bus Radial Network	58
3.10	Time-Current Characteristic Curve of Definite Time Overcurrent Relay	59
3.11	Time-Current Characteristic Curves of Inverse Time Overcurrent Relays	60
3.12	Adjustable TMS of Overcurrent Relay with IDMT Model	61
3.13	Single-End Fed Power System of Parallel Feeders with only OCRs	61
3.14	Single-End Fed Power System of Parallel Feeders with OCRs and DOCRs	62
3.15	Radial Network with Directional Overcurrent Relays	65
3.16	Single-line diagram of the 3-bus system	66
3.17	TCCCs of R4/R6 Relay Pair with Respect to I_{relay}	74
3.18	TCCCs of R4/R6 Relay Pair with Respect to PSM	75
3.19	Flowchart of the General Optimal Coordination of DOCRs	77
3.20	Determining Primary/Backup Relay Pairs of a Simple 5-Bus System	81
3.21	Fault Analysis for the 3-Bus System Using MATLAB Simulink	86
4.1	Migration Process Between Islands with Different Isolation, Shape, and Size	103
4.2	Equilibrium Model of a Biota of a Single Island	104
4.3	Equilibrium Model of faunas of Different Islands	105
4.4	Simplified Equilibrium Model of a Biota of a Single Island	106
4.5	The Colonization Curve	108
4.6	Comparison Between $Ps(t)$ and $m(t)$ at Different m_{max}	118
4.7	General Flowchart of the BBO Algorithm	122

4.8	Curves of Fitness Functions of PMB, SMB, SPMB, and SSMB for Some Selected Functions	128
4.9	The Original and Modified *SIV*-Selection Process for Migration and Mutation Stages	132
4.10	Curves of Fitness Functions of BBO and BBO-EM for some selected functions	139
4.11	Illustrations in the Iterated Search Space and the Location of the Optimal Solution	151
4.12	Distribution of Generation Levels in a Decaying Form	153
4.13	Flowchart of the ISS-BBO Algorithm	155
4.14	Log-Log Curves of Fitness Functions of BBO, BSU, BMU and ISS for some selected functions	161
4.15	Illustration in the Adaptive Variable Bounds and the Location of the Optimal Solution	166
4.16	The Actual Updating Manner of Iterative and Adaptive Models on $5+2$ of Perm No.1 Function	170
5.1	Four Categories of Selectivity Constraint Lines	175
5.2	Flowchart of the Proposed BBO-LP Algorithm	177
5.3	Single-Line Diagram of the 3-Bus System for Test Case 1	178
5.4	Single-Line Diagram of the 4-Bus System for Test Case 2	185
5.5	Single-Line Diagram of the 8-Bus System for Test Case 3	189
5.6	Single-Line Diagram of the 9-Bus System for Test Case 4	191
5.7	The Best Obtained Solutions Through 100 Trails for the Test Case 4	192
5.8	Fitness Function of the Conventional BBO Algorithm within 500 Generations for the Test Case 4	192
5.9	Single-Line Diagram of the 15-Bus System Test Case 5	195
5.10	Proposed Modified Flowchart of the BBO Algorithm	202
5.11	Log-Log Curves of Fitness Function with Different TCCCs Test Case 5	212

List of Appendices

No.	Appendix
A	The derivation of the species colonization curve on an island
B	The 1st proof of the iterative variable bounds algorithms
C	The 2nd proof of the iterative variable bounds algorithms
D	The required data of the electrical case tests
E	Some information about the involved benchmark functions

List of Abbreviations

Abbreviation	Description
ACO	Ant Colony Optimization
ANSI	American National Standards Institute
AVB	Adaptive Variable Bounds
BBBO	Blended Biogeography-Based Optimization
BBO	Biogeography-Based Optimization
BBO-EM	Biogeography-Based Optimization with Essential Modifications
BCO	Bee Colony Optimization
BFO	Bacterial Foraging Optimization
BMU	Bounds Multi Updater
BPS	Break-Point Set
BSU	Bounds Single Updater
CB	Circuit Breaker
CPU	Central Processing Unit
CT	Current Transformer
CTI	Coordination Time Interval
CTR	Current Transformer Ratio
DC	Direct Current
DE	Deferential Evolution
DG	Distributed Generation
DigSILENT	Commercial software
DOCR	Directional Overcurrent Relay
DSP	Digital Signal Processor
EA	Evolutionary Algorithm
EBBO	Enhanced Biogeography-Based Optimization
EG	External Grid
EMM	Ensemble of Migration Models
EMS	Energy Management System
EPF	Exterior Penalty Function
ES	Evolutionary Strategy

Abbreviation	Description
ETAP	Commercial software
FPAB	**F**lower **P**ollination by **A**rtificial **B**ees
GA	**G**enetic **A**lgorithm
GenSin	**Gen**eralized **Sin**usoidal
IDMT	**I**nverse **D**efinite **M**inimum **T**ime
IEC	**I**nternational **E**lectrotechnique **C**ommission
IEEE	**I**nstitute of **E**lectrical and **E**lectronics **E**ngineers
IPF	**I**nterior **P**enalty **F**unction
ISI	**I**sland **S**uitability **I**ndex
ISS	**I**terative **S**earch **S**pace
KKT	**K**arush-**K**uhn-**T**ucker
LP	**L**inear **P**rogramming
MATLAB	Commercial software
MGN	**M**inimum **G**eneration **N**umber
MINLP	**M**ixed-**I**nteger **N**on-**L**inear **P**rogramming
NFE	**N**umber of **F**unction **E**valuations
NFT	**N**ear **F**easibility **T**hreshold
NLP	**N**on-**L**inear **P**rogramming
OBBO	**O**ppositional **B**iogeography-**B**ased **O**ptimization
OBL	**O**pposition-**B**ased **L**earning
OCR	**O**vercurrent **R**elay
OLF	**O**verload **F**actor
P/B	**P**rimary/**B**ackup
PEB	**P**artial **E**migration-**B**ased
PMB	**P**artial **M**igration-**B**ased
PS	**P**lug **S**etting (or, **P**ick-up **S**etting)
PSA	**P**ossible **S**olution **A**rea
PSM	**P**lug **S**etting **M**ultiplier
PSO	**P**article **S**warm **O**ptimization
PT	**P**otential **T**ransformer
RAM	**R**andom **A**ccess **M**emory
RCBBO	**R**eal-**C**oded **B**iogeography-**B**ased **O**ptimization

Abbreviation	Description
ROM	Read Only Memory
SAPF	Self-Adaptive Penalty Function
SF	Safety Factor
SFP	Superiority of Feasible Points
Sin	Sinusoidal
SIV	Suitability Index Variable
SMB	Single Migration-Based
SMIB	Single Machine Infinite Bus
SPMB	Simplified Partial Migration-Based
SSMB	Simplified Single Migration-Based
StdDev	Standard Deviation
STI	Selective Time Interval
TCCC	Time-Current Characteristic Curve
TDS	Time Dial Setting
TEB	Total Emigration-Based
TIB	Total Immigration-Based
TMS	Time Multiplier Setting
WSO	Wasp Swarm Optimization

List of Symbols

Symbol	Description	Unit	Ch.
$A^{'}$	Biogeography matrix	$unitless$	4
A_d	Augmented incidence matrix	$unitless$	3
B	Number of buses	$unitless$	3
d	User-defined value	$unitless$	4
d_p	Penalty multiplier for the pth constraint	$unitless$	2
$d(X)$	Distance value	$unspecified$	2
e	Eigenvalues of the biogeography matrix $A^{'}$	$unitless$	4
E	The maximum possible emigration rate	%	4, 5
$e_{abs,i}$	Absolute error of the obtained solution $f_{best,i}$ for the ith trial	$unspecified$	2
$e_{biggest\_ever}$	The biggest absolute error among all T trials	$unspecified$	2
$e_{i,j}$	The matrix element located in the ith row and the jth column	$unitless$	3
e_{mean}	The mean of all $e_{abs,i}$ among T trials	$unspecified$	2
e_{std_dev}	The standard deviation of all $e_{abs,i}$ among T trials	$unspecified$	2
$e_{smallest\_ever}$	The smallest absolute error among all T trials	$unspecified$	2
$Elit$	Elitism; the saved best solutions to be used for the next generation	$unspecified$	4, 5
F	Fault on the transmission line	% or m	3
$f(x)$	Dependent variable or function of x	$unspecified$	2
$f_{best,i}$	The best obtained solution for ith trial	$unspecified$	2
f_{best_ever}	The best of the best obtained solutions among T trials	$unspecified$	2
f_{max}	The maximum element among vector of f	$unspecified$	2
f_{mean}	The mean of all f_{best} among T trials	$unspecified$	2
f_{min}	The minimum element among the vector of f	$unspecified$	2
f_{std_dev}	The standard deviation of all f_{best} among T trials	$unspecified$	2
$f(X)$	Dependent variable or function of X	$unspecified$	2
$f_{min}(X^*)$	Global optimum solution	$unspecified$	2
G_l	The generations of lth stage	$unitless$	4
$[G]_m$	The set of m inequality constraints	$unspecified$	2
g_q	The qth inequality constraint	$unspecified$	2
$\hat{g}_q(X)$	The qth equality or inequality constraint	$unspecified$	2

Symbol	Description	Unit	Ch.
$\hat{g}_q^{max}(X)$	The biggest element of $\hat{g}_q(X)$	$unspecified$	2
G_T	The total number of generations	$unitless$	4
h_p	The pth equality constraint	$unspecified$	2
$[H]_l$	The set of l equality constraints	$unspecified$	2
I	Current	A	3
I	The maximum possible immigration rate	%	4
$I_{f,i}^{min}$	The minimum fault current that should be detected by the ith relay	A	3, 5
$I_{n,i}$	The nominal current rating of the circuit protected by the ith relay	A	3, 5
I_{relay}	Current seen by the relay after be stepped down through CT	A	3
$I_{R_i,k}$	Short circuit current at the kth fault location that is seen by the ith primary relay	A	3, 5
$I_{R_j,k}$	Short circuit current at the kth fault location that is seen by the jth backup relay	A	3, 5
ISI^*	Global optimum solution	$unspecified$	4
ISI_{best}	The best ISI among k-islands	$unspecified$	4
$\bar{k}$	Population size	$unitless$	2, 4
k	Fault location	% or m	3, 5
K	Constant value	$unspecified$	2
l	The total number of fault locations	$unitless$	3
L	The total number of lines	$unitless$	3
m	Mutation rate	%	4
m_{max}	User-defined maximum mutation rate	%	4, 5
$M(X)$	Normalized magnitude of the constraint violations	$unitless$	2
n	The total number of independent variables "dimension"	$unitless$	2, 4, 5
n_r	The total number of relays	$unitless$	3
$N(X)$	Normalized $f(X)$	$unitless$	2
P_{max}	The maximum migration probability	%	4
P_s	The migration probability of the species	%	4
PS_i^{max}	Upper bound of PS of the ith relay	A	3, 5
PS_i^{min}	Lower bound of PS of the ith relay	A	3, 5
$P(X)$	Penalty function	$unspecified$	2
r_q	Penalty multiplier for the qth constraint	$unitless$	2, 5

Symbol	Description	Unit	Ch.
R	The ratio between the total generations and and the generations at the first stage	%	4
$R_{i,pr}$	The ith Primary relay	$unitless$	3
$R_{j,bc}$	The jth Backup relay	$unitless$	3
R_L	Line resistance	Ω	3
R_m	User-defined mutation range	%	4
R_s	Source resistance	Ω	3
S	The colonization on an island "species count"	$unitless$	4
$\hat{S}$	The equilibrial number of species	$unitless$	4
S_{max}	The maximum number of species on an island	$unitless$	4
SIV_{max}	Upper bound of SIV	$unspecified$	4
SIV_{min}	Lower bound of SIV	$unspecified$	4
t	Optimization generation as a time	$unitless$	2
T	Relay operating time	s	3
T_r	The total number of trials	$unitless$	2
$\hat{T}$	Species turnover rate at eqilibrium	%	4
T_{CB}	Circuit breaker operating time	s	3
$T_{pr,q}^{far}$	The operating time of the qth primary relay for the far-end fault	s	3, 5
$T_{i,k}$	The operating time of the ith primary relay for the kth fault location	s	3, 5
$T_{i,k}^{'}$	The transient operating time of the ith primary relay for the kth fault location	s	3, 5
$T_{i,primary}$	The operating time of the ith primary relay	s	3
$T_{j,backup}$	The operating time of the jth backup relay	s	3
$T_{j,k}$	The operating time of the jth backup relay for the kth fault location	s	3, 5
$T_{j,k}^{'}$	The transient operating time of the jth backup relay for the kth fault location	s	3, 5
T_{ik}^{max}	The maximum allowable operating time of the ith relay at the kth fault location	s	3, 5
T_{ik}^{min}	The minimum allowable operating time of the ith relay at the kth fault location	s	3, 5
$T_{pr,p}^{near}$	The operating time of the pth primary relay for the near-end fault	s	3, 5
$TMS^{initial}$	Initial guess of TMS	$unitless$	3
TMS_i^{max}	Upper bound of TMS of the ith relay	$unitless$	3, 5
TMS_i^{min}	Lower bound of TMS of the ith relay	$unitless$	3, 5

Symbol	Description	Unit	Ch.
T_{OS}	Relay overshoot time	s	3
T_{SM}	Safety margin time	s	3
u	The slope of a linear equation	%	4
v	Eigenvector of the biogeography matrix $A^{'}$	$unitless$	4
V_s	Source voltage	V	3
$v(X)$	Indication of constraints violation	$unspecified$	2
w_{ik}	The probability of a given fault at the kth fault location for the ith primary relay	$unitless$	3
x	Independent variable	$unspecified$	2
x^{max}	Upper bound or limit of x	$unspecified$	2
x^{min}	Lower bound or limit of x	$unspecified$	2
$[X]$	The set of design variables "the design vector"	$unspecified$	2
X^*	Global Optimum point	$unspecified$	2
X_L	Line reactance	Ω	3
X_s	Source reactance	Ω	3
Z	The objective function of the coordination problem	s	3, 5
Z_L	Load impedance	Ω	3
Z_s	Source impedance	Ω	3
α	Constant value	$unitless$	2, 3, 4, 5
β	Constant value	$unitless$	2, 3, 4, 5
γ	Constant value	$unitless$	2, 3, 5
ϵ	Minimum accepted error	$unspecified$	2
θ	Minimal emigration rate	%	4
$\phi(X)$	Penalized cost function	$unspecified$	2
λ	Immigration or speciation rate	%	4
μ	Emigration or extinction rate	%	4
3ϕ	Three Phase Quantity	$unspecified$	3, 5

Math Symbol	Description	Chapter
$\emptyset$	Empty set	2
$\mathcal{F}$	Feasible	2
$\forall$	For all	2
$\uparrow$	Increase	2, 5
$\downarrow$	Decrease	2, 5
$\propto$	Proportional to	2, 4, 5
$<$	Less than	2, 3, 4
$\leqslant$	Less than or equal to	2, 3, 5
$>$	Greater than	2, 4, 5
$\geqslant$	Greater than or equal to	2, 3, 4, 5
$\in$	Set membership	2, 3, 4
$\approx$	Approximately equal to	2, 3

Chapter 1

Introduction

1.1 Literature Review
1.2 Motivation and Objectives
1.3 Contributions
1.4 Thesis Organization
1.5 Publications

Chapter 1
Introduction

Directional overcurrent relays (DOCRs) are applied in multi-loop systems, ring feeder systems, double-end fed power systems, or even single-end fed power systems of parallel feeders. DOCRs can be used as primary relays for interconnected sub-transmission and distribution systems, and as local backup relays for transmission systems (Urdaneta, Perez, & Restrepo, 1997; Gers & Holmes, 2004; Paithankar & Bhide, 2003; Rao, 2008).

The coordination of DOCRs is a very important stage for any protection design. Correct relay coordination involves selecting the suitable relay setting that assures faults in the protected zone are cleared first by the corresponding primary relays and if they fail, the corresponding backup relays act after a coordination time delay. The relays have two settings, namely; **"plug setting"** (PS) and **"time multiplier setting"** (TMS). The operating time of each relay is a function of these two settings.

The coordination problem can be considered as a highly constrained mixed-integer non-linear programming (MINLP) problem, where TMS is continuous and PS is discrete. It requires an expert protection engineer to solve it analytically, where all faults, system contingencies and abnormalities are analyzed and predetermined. Alternatively, it can be solved easily by using optimization methods (Noghabi, Sadeh, & Mashhadi, 2009). The objective of the optimal coordination of DOCRs is to

find the values of PS and TMS of all relays that minimize the sum of their operating times when they act as primary relays and all the coordination constraints are satisfied (Urdaneta, Nadira, & Perez Jimenez, 1988).

In this Chapter, literature review, motivation and objectives, contributions, thesis organization, publications and used soft tools will be presented.

1.1 Literature Review

Solving the coordination problem of DOCRs is an old tedious task. In the past, the optimal coordination was impossible based on the availability of techniques and technologies at that time. Instead, an acceptable non-optimal coordination was implemented to meet the protection requirements for the simple electrical networks that were existing at that time.

There are different approaches to solve the coordination problem of DOCRs. They can be classified as: Topological methods (graph theory and functional dependencies) and optimization methods (including adaptive techniques), which are summarized in (Hui, 2006).

In 1960s, the computer was used to overcome the laborious calculation (Knable, 1961; Radke, 1963; Albrecht, Nisja, Feero, Rockefeller, & Wagner, 1964; Tsien, 1964). It was assumed that the relative sequence of computation of DOCRs settings is predetermined. A trial

and error approach is considered in (Albrecht et al., 1964). A. H. Knable (Knable, 1969) is the first one who considered both parts of the coordination problem; namely, the relay settings computation and the relative sequence of that computation by using heuristic scheme. Dwarakanath et al. (Dwarakanath & Nowitz, 1980) used the concept of linear graph theory to determine the relative sequence in a multi-loop network based on a minimal break point set (BPS). Jenkins et al. (Jenkins, Khincha, Shivakumar, & Dash, 1992) used the concept of functional dependency as a flexible and powerful alternative to identify the minimal BPS.

In 1988, Gauss-Seidel iterative technique was successfully used by Urdaneta et al. (Urdaneta et al., 1988). However, because the coordination of DOCRs is a non-linear problem, predetermined values of PS are considered in order to convert it to a linear problem. Although, this approach reduces the CPU time (due to using less complicated algorithms) and makes the problem easier to solve, it needs an expert protection engineer for setting the initial values of PS. In addition, there is no guarantee for converging to the global optima.

Linear Programming (LP) techniques, including simplex, two-phase simplex, and dual simplex, are commonly used if the coordination problem is constructed in a linear form (Moravej, Jazaeri, & Gholamzadeh, 2012). However, when both PS and TMS are considered as variables, the problem becomes non-linear and hence Non-Linear Programming (NLP) is used. Also, to increase the chance for finding feasible solutions, the values of PS are taken as a continuous and then rounded-off

at the end of the program for getting a discretized PS. As a result, the final solution may drift away from its previous optimum state. LP is fast and simple algorithm, but it is highly venerable to trapped in a local minima. NLP gives better performance, but it also converges to a local minima, especially with wrong selected initial values of PS and TMS, and it is more complex than LP and consumes more CPU time. In addition to the previously mentioned traditional algorithms, evolutionary algorithms (EAs), such as ant colony optimization (ACO), bee colony optimization (BCO), genetic algorithm (GA), differential evolution (DE), particle swarm optimization (PSO), etc, have been successfully applied to solve the optimal coordination problem (Ezzeddine & Kaczmarek, 2011b). EAs are multi-point search techniques and can be used to ensure reaching the area of global optimum, but their convergence speed is slow and requires high CPU time. Hybrid techniques have been considered to overcome the drawbacks of the previous optimization algorithms. However, these algorithms become more complex for implementation (Noghabi et al., 2009; Bedekar & Bhide, 2011).

Biogeography-based optimization (BBO) is a new population-based evolutionary algorithm that was presented by Dan. Simon in Dec. 2008 (Simon, 2008a). This technique is based on an old mathematical study that explains the geographical distribution of biological organisms. Its performance was evaluated based on 14 benchmark functions, and then applied to solve a real sensor selection problem for aircraft engine health estimation. BBO did well and proved that it is a very competitive and

fast method as compared to the other EAs. Although, there are many of models that deal with the modifications or/and hybridizations of the BBO algorithm with other optimization algorithms, there are still lots of venues for further research to improve the BBO algorithm performance and its application to solve the optimal coordination problem, which will be highlighted in the following Section.

1.2 Motivation and Objectives

The research area of the optimal coordination of DOCRs is very active since 1988 (Urdaneta et al., 1988). Lots of ideas were presented in the literature in order to solve the continuously growing problems that should be tackled in the field of power system protection. In the opposite direction, there are lots of remaining problems that need to be solved, such as finding new algorithms that can solve the optimal coordination problem within a short time. This is an important factor and one of the main properties of any protective system. Also, for on-line optimal coordination, the calculation has to be carried out quickly in order to prevent any damage to the equipments. Based on that, it gives enough motivation to select this area as the core of this thesis.

Most of the modern optimization techniques are already used for finding the optimal coordination. Thus, it led this study to select one of the latest invented evolutionary algorithms (*i.e.,* BBO), because of its performance on both solution quality and processing speed (Simon, 2008a). BBO was presented with different original forms, and there are

many unanswered questions, like which is the best original form? How to overcome the inherent weaknesses of the BBO algorithm without complicating its structure? Is it possible to create better algorithms than those presented in the literature? etc. All of these questions expand the motivation and give ambition to conduct more investigation on this new algorithm before applying it to solve the optimal coordination problem of DOCRs.

Based on that, the objective of this thesis is split into four phases: selecting the best BBO algorithm among the four original forms, applying the essential modifications to improve BBO performance, proposing advanced BBO models, and finally solving the optimal coordination problem of DOCRs for different case studies and testing scenarios.

1.3 Contributions

The researcher could not find any study in the literature that extensively analyse the performance of the four original BBO forms in order to present their strength and weakness, based on which the best form can be identified. Once the BBO inherent strengths and sources of exploration weakness are identified, a new superior algorithm can be proposed. The researcher could not also find any study in the literature focusing on the application of BBO in power system protection, particularly in the area of optimal coordination of protective relays, as well as there is no study focusing on the performance of different constraint-handling techniques. In addition, finding feasible and optimal solution is very difficult

for complicated systems, where the speed and simplicity of any proposed algorithm are very important issues that have to be considered. The following points summarize the new contributions of this thesis:

1. A comprehensive comparison of the four original BBO forms is done in order to identify the best one.
2. Enhancing the performance of the original BBO forms by correcting some sources of the weaknesses, and presenting new algorithm called BBO-EM.
3. Proposing four new BBO models, namely; BSU-BBO, BMU-BBO, ISS-BBO and AVB-BBO.
4. The proposed BBO-EM algorithm is applied to solve the optimal coordination problem of DOCRs and its performance is evaluated and compared for ten different constraint-handling techniques.
5. A new pre-processing sub-algorithm is proposed to avoid the infeasible solutions.
6. A hybridization of BBO with LP is proposed to improve the convergence speed by reducing the required number of iterations and hence reduce the CPU time.
7. The proposed ISS-BBO is applied to solve the optimal coordination problem of DOCRs.
8. The proposed BBO-EM is applied to solve the optimal coordination problem of DOCRs when their TCCCs are variable.

The performance of all the proposed BBO algorithms is extensively evaluated using large number of benchmark functions. Also, the proposed BBO algorithms are applied to solve the optimal coordination

problem of different IEEE test systems, and the results are compared with the performance of the other proposed methods in the literature.

1.4 Thesis Organization

This thesis is organized as follows: Chapter 2 covers the mathematical representation of any optimization problem. Chapter 3 presents the fundamentals of power system protection, illustrates with examples the coordination problem of DOCRs, and formulates the general mathematical model of the optimal coordination problem of DOCRs. The selection, correction and modification on the original BBO form are proposed and presented in Chapter 4. In Chapter 5, the optimal coordination of the DOCRs is solved using several proposed BBO based algorithms, including hybridization of BBO with LP. Finally, the conclusions, suggestions and future works are presented in Chapter 6.

1.5 Publications

- Conferences:

  1. Ali R. Alroomi, Fadhel A. Albasri and Jawad H. Talaq, "Performance Comparison between the Original Forms of Biogeography-Based Optimization," *Second International Conference on Advanced Information Technologies and Applications (ICAITA-2013)*, pp. 121-140, Nov. 2013.
  2. Ali R. Alroomi, Fadhel A. Albasri and Jawad H. Talaq, "Essential Modifications on Biogeography-Based Optimization Algorithm," *Second In-*

ternational Conference on Advanced Information Technologies and Applications (ICAITA-2013), pp. 141-160, Nov. 2013.

- Journals:

1. Ali R. Alroomi, Fadhel A. Albasri and Jawad H. Talaq, "A Comprehensive Comparison of the Original Forms of Biogeography-Based Optimization Algorithms," *Inter. Journal on Soft Computing, Artificial Intel. and Applications*, vol. 2, no. 5/6, pp. 11-30, Dec. 2013.
2. Ali R. Alroomi, Fadhel A. Albasri and Jawad H. Talaq, "Solving the Associated Weakness of Biogeography-Based Optimization Algorithm," *Inter. Journal on Soft Computing*, vol. 4, no. 4, pp. 1-20, Nov. 2013.
3. Fadhel A. Albasri, Ali R. Alroomi and Jawad H. Talaq, "Optimal Coordination of Directional Overcurrent Relays Using Biogeography-Based Optimization Algorithms," *To be published in IEEE Transactions on Power Delivery.*
4. Ali R. Alroomi, Fadhel A. Albasri and Jawad H. Talaq, "Iterative Search Space - Biogeography Based Optimization," *Submitted for publication.*
5. Ali R. Alroomi, Fadhel A. Albasri and Jawad H. Talaq, "Adaptive Variable Bounds - Biogeography Based Optimization," *Submitted for publication.*
6. Fadhel A. Albasri, Ali R. Alroomi and Jawad H. Talaq, "Optimal Coordination of Directional Overcurrent Relays Using Iterative Search Space Biogeography-Based Optimization Algorithm," *Submitted for publication.*

Chapter 2
Mathematical Optimization

2.1 Optimal Problem Formulation
2.2 Types of Optimization Techniques
2.3 Constraint-Handling Techniques
2.4 Performance Criteria for Evaluating Algorithms

Chapter 2
Mathematical Optimization

The design problem could be one of the four possible types, as summarized in Table 2.1 (Eiben & Smith, 2003; Leondes, 1998).

Table (2.1)
Problem's types

Constraints	Objective Function	
	Yes	No
Yes	Constrained Optimization Problem	Constraint Satisfaction Problem
No	Free Optimization Problem	No problem

This Chapter will give a brief summary about the general standard formulation of any optimization problem, and will state the main advantages and disadvantages of the three main optimization techniques. In addition, a detailed description about ten selected constraint handling techniques will be covered later, and followed by the performance criteria that will be used in this research to evaluate the obtained results.

2.1 Optimal Problem Formulation

The optimal design formulation of any problem can be described through the general flowchart of Fig.(2.1) (Deb, 2010). The main elements of that formulation can be listed as (Venkataraman, 2009; Deb, 2010; Marriott & Stuckey, 1998):

- Design Parameters

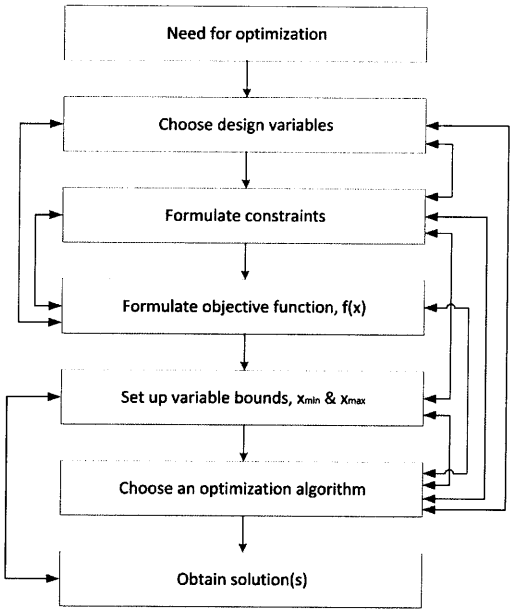


Figure (2.1)
Flowchart of the optimal design procedure

- Design Variables
- Design Function
- Objective Function(s)
- Design Constraints
  - Functional Constraints
    - Equality Constraints
    - Inequality Constraints
  - Side Constraints

2.1.1 The Standard Format

First, assume the followings:

- The symbol h is used to represent one equality constraint $\Rightarrow$ Total equality constraints: $[h_1, ..., h_l]$, or $h_p : p = 1, ..., l$; or simply

using vector notation $[H]$

- The symbol g is used to represent one inequality constraint $\Rightarrow$ Total inequality constraints: $[g_1, ..., g_m]$, or $g_q : \ q = 1, ..., m$; or simply using vector notation $[G]$
- The side constraints are defined as $x_i \in \left[x_i^{min}, x_i^{max}\right] : \ i = 1, ..., n$, or in a vector notation as $X \in \left[X^{min}, X^{max}\right]$ or $(X^{min} \leqslant X \leqslant X^{max})$
- If the global optimum solution is known, then it is mentioned as $f_{min}(X^*)$ at X^*

The optimization problem of any design function can be formulated in a general standard mathematical model by taking into account all the previous issues as follows:

$$\min_X f(x_1, x_2, ..., x_n)$$

$$\begin{aligned}
\text{Subjected to: } h_1(x_1, x_2, ..., x_n) &= 0 \\
h_2(x_1, x_2, ..., x_n) &= 0 \\
\vdots &= \vdots \\
h_l(x_1, x_2, ..., x_n) &= 0 \\
g_1(x_1, x_2, ..., x_n) &\leqslant 0 \\
g_2(x_1, x_2, ..., x_n) &\leqslant 0 \\
\vdots &\leqslant \vdots \\
g_m(x_1, x_2, ..., x_n) &\leqslant 0
\end{aligned}$$

$$x_1^{min} \leqslant x_1 \leqslant x_1^{max}$$

$$x_2^{min} \leqslant x_2 \leqslant x_2^{max}$$

$$\vdots \quad \leqslant \quad \vdots \quad \leqslant \quad \vdots$$

$$x_n^{min} \leqslant x_n \leqslant x_n^{max}$$

$$\text{where: } f_{min}(X^*) = a \ , \quad x_i^* = (s_1, s_2, \cdots, s_n) \tag{2.1}$$

Equation (2.1) can also be expressed as:

$$\min_X f(x_1, x_2, ..., x_n)$$

$$\text{Subjected to: } h_p(x_1, x_2, ..., x_n) \ = \ 0 \ , \quad p = 1, 2, \cdots, l$$

$$g_q(x_1, x_2, ..., x_n) \ \leqslant \ 0 \ , \quad q = 1, 2, \cdots, m$$

$$x_i^{min} \leqslant x_i \leqslant x_i^{max} \quad i = 1, 2, \cdots, n$$

$$\text{where: } f_{min}(X^*) = a \ , \quad x_i^* = s_i \tag{2.2}$$

In addition, Eq.(2.1) can also be formulated in vector notation as:

$$\min_X f(X), \ [X]_n$$

$$\text{Subjected to: } [h(X)]_p \ = \ 0 \ , \quad p = 1, 2, \cdots, l$$

$$[g(X)]_q \ \leqslant \ 0 \ , \quad q = 1, 2, \cdots, m$$

$$X^{min} \leqslant X \leqslant X^{max}$$

$$\text{where: } f_{min}(X^*) = a \ , \quad X^* = S \tag{2.3}$$

2.2 Types of Optimization Techniques

In general, optimization techniques can be classified under two main categories; namely traditional and modern (or non-traditional) techniques.

2.2.1 Traditional Optimization Algorithms

The traditional optimization methods are the most known methods that need no special acknowledgement from some fields of science, like biology and its branches. They are straight-forward algorithms that use regular mathematical steps, like finding the derivatives, constructing matrices, tracing the error deviation between two iterations, etc.

2.2.1.1 Advantages

Some of the advantages of these techniques can be summarized as follows (Momoh, 2001; Fletcher, 1987; Forst & Hoffmann, 2010; Chong & Żak, 2001; Nocedal & Wright, 2006; K. Y. Lee & El-Sharkawi, 2008):

- They are considered very fast algorithms. If the initial guess is good, then these techniques become very useful to be embedded in systems that need fast decision, such as the field of protection. The high speed performance comes from their operational principles, because they need only one point to be optimized. Thus, dealing with just one individual means fast calculation process, which is reflected directly on the overall consumed CPU time.

However, this term (*i.e., the speed issue*) may not be correct in some conditions. Suppose, there is a $high$-dimensional problem and needs to be solved by Newton-Raphson method; dealing with a large Jacobian matrix will consume high value of CPU time.

- These techniques are very old and they are available in thousands of books, which make them easy to be reached. There are many available ready made codes with different formats and languages (*like MATLAB, JAVA, C++, etc*).

- The gradient-based methods can be used as a second stage of hybrid global optimization techniques. One of these techniques will be used later in Chapter 5.

2.2.1.2 Disadvantages

Unfortunately, there are many limitations on these traditional optimization techniques. Some of these weaknesses can be summarized as follows:

- Some of them are very complicated algorithms and hard to be implemented.

- Some of them are restricted to 1-dimensional problems.

- Some of them need to find some derivatives and/or constructing matrices.

- Most of them can be easily trapped into a local optima if the initial guess is not good. In case of minimization problem, if this point is selected between two valleys with one local hill, then the algorithm can not escape from that local optima.

2.2.2 Non-Traditional Optimization Algorithms

They are also called **modern optimization algorithms**. Based on the previous disadvantages, the researchers are motivated to search for other new techniques that have the ability to converge accurately and quickly to the area of the global optima. They utilized many of the available mathematical models from the branches of physics science (*like simulated annealing from matter physics, and gravitational search algorithm from the law of gravity*) and the branches of biology science (*like genetic algorithm from genetic science, and biogeography based-optimization from biogeography science*).

With these inspirations, the researchers have successfully applied these models in order to solve the most complex problems by just plugging-in the design function without any derivative, matrices and/or good guess for the initial point. As a result, the new methods are called heuristic-based methods (Floreano & Mattiussi, 2008; Luke, 2009).

The modern optimization techniques can also be classified according to their number of generated individuals per one generation. If the algorithm depends only on one individual (*like traditional techniques*), it is called **single-solution algorithm**; such these algorithms are tabu

search and simulated annealing. Whereas, the term **population-based algorithm** is used to identify the algorithms that generates multiple individuals per one generation; such these algorithms are particle-swarm optimization and biogeography-based optimization. Of course, the algorithm of non-traditional single-solution approaches has the ability to escape away from trapping into a local optima.

Before stating the advantages and disadvantages of these modern techniques, it seems essential to differentiate between **heuristic**, **meta-heuristic** and **hyper-heuristic** terms, because some references just mentioned them as a general word "heuristic" without any clear definition:

2.2.2.1 Heuristic vs. Meta-Heuristic vs. Hyper-Heuristic

The first information needed to be known is that the heuristic-based technique is a problem-dependent technique, which can learn from the given information about the design function and then adapt with that to provide good results. These techniques could fail due to the probability of trapping into a local optima.

Now, suppose there is no useful information about the given problem, like the known path for reaching the optimum solution or what it looks like, then the technique that can reach the space where the global optima is located in without knowing how and from where to start is called metaheuristic-based technique (**black-box optimization**). They benefit from the data of the obtained solutions to determine the loca-

tion of the best answer within the search space. Therefore, it may not guarantee settle exactly on the global optimum solution, and instead may provide an approximation to it.

Hyper-heuristic is the most advanced technique comparing to the preceding two techniques. It depends on the heuristic or metaheuristic technique to create its own search space. Therefore, the word "hyper-heuristic" can be translated as "heuristic search for heuristic" (Luke, 2009; Talbi, 2009; Gendreau & Potvin, 2010).

2.2.2.2 Advantages

Some of the advantages of the heuristic-based methods are (Deb, 2010; Eiben & Smith, 2003; Luke, 2009; Rao, 2009; Simon, 2013):

- No need anymore to find the derivative or constructing any matrix.
- Very robust and can converge to the space of the global optima.
- They are relatively easier to be understood.
- Designer will need less time to create the program.
- The designed program can be used as a general purpose optimizer for any plugged-in design function.

2.2.2.3 Disadvantages

Some of the disadvantages of the heuristic-based methods can be summarized as follows:

- The population-based techniques are time consuming. The processing time will increase significantly as the population size increases.
- The final solution is tuned after high number of generations, because the convergence is based on the randomly generated individuals rather than gradient-based as in traditional techniques.
- Still the number of available references and codes is smaller than that of the traditional techniques, especially for the most recent invented techniques.
- To understand their principles and how they work, the investigator needs to study some special topics in physics or/and biology sciences, which may become hard for those people who have pure background only in the field of engineering, computer science or economics.

2.2.3 Hybrid Optimization Algorithms

In general, the main objective of the new non-traditional techniques is to maximize the advantages and minimize the disadvantages of the previous two branches of the optimization algorithms. In addition, the

investigator has large area for maneuver and can apply his hybridizations as well as modifications in many locations of the algorithm. Therefore, the overall algorithm will integrate all of these improvements.

Minimizing the disadvantages that are associated with the individual optimization algorithms does not mean that the overall hybrid algorithm will have zero disadvantages. These additional sub-algorithms will lead to slow processing time (*unless modifying/bypassing some stages of the original algorithms or/and reducing the simulation parameters*), very complicated algorithm, hard to be modified by other programmers and there is no standard format since the hybridization can be done anywhere.

In Chapter 5, a new proposed hybridization between the biogeography-based optimization algorithm and the standard linear programming algorithm (**Hybrid BBO-LP**) will be introduced with high accuracy and processing time performance.

2.3 Constraint-Handling Techniques

The side constraints can be satisfied easily by driving the independent variables to be within the decided bounds. Unfortunately, this direct solving approach does not work with the functional constraints (*i.e., equality and inequality*) because they require more complicated constraint-handling techniques. Selecting the correct type is very important, because the algorithm's speed and accuracy can be affected markedly by the inefficient techniques (Eiben & Smith, 2003).

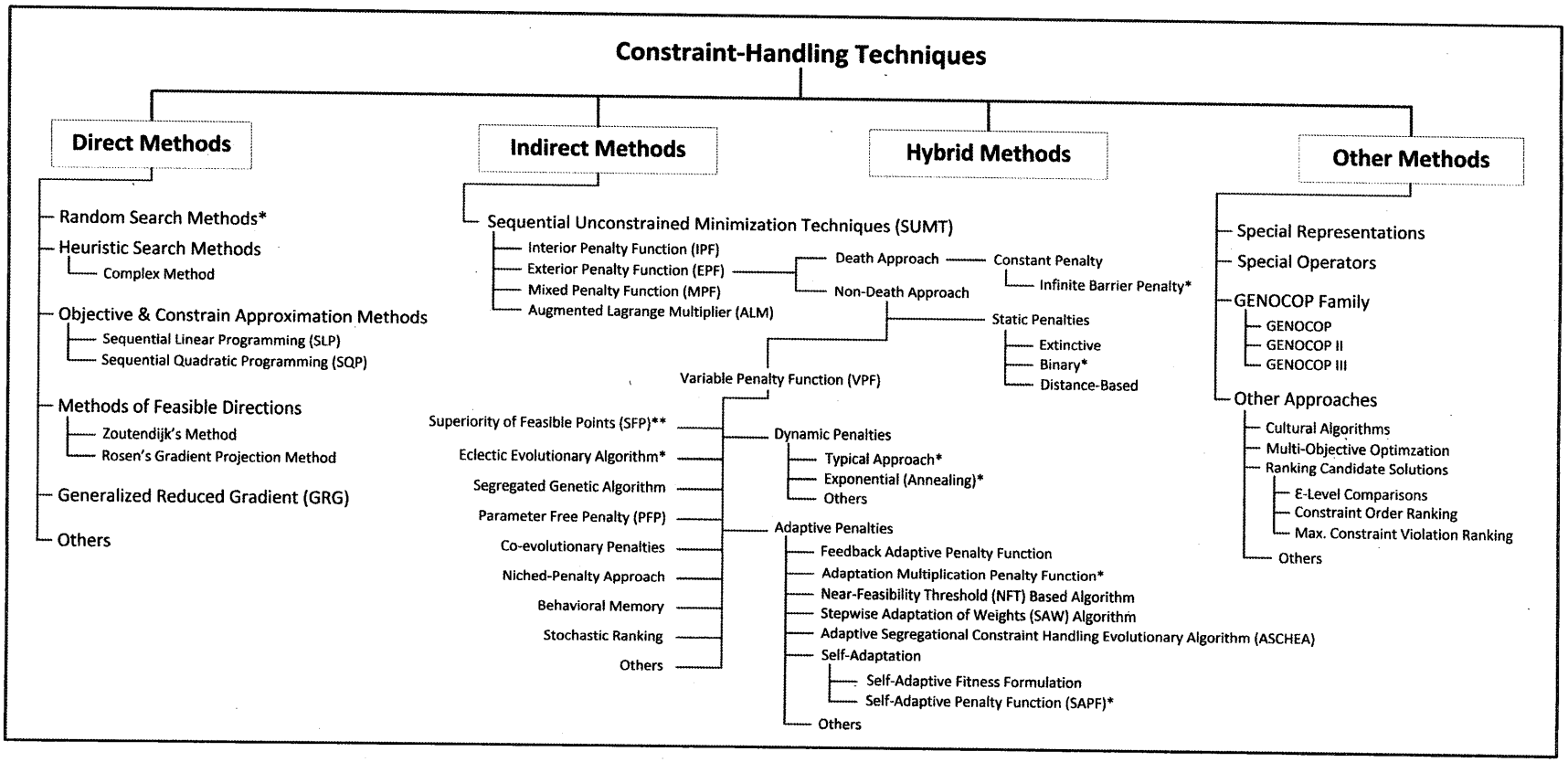


Figure (2.2)
Summary of the most popular constraint-handling techniques

Figure (2.2) summarizes the most popular constraint-handling techniques (Deb, 2010; Venkataraman, 2009; Simon, 2013; Rao, 2009; Eiben & Smith, 2003; Yeniay, 2005; Kajee-Bagdadi, 2007; Yu & Gen, 2010). From this list, only ten different techniques are selected and analysed in this work, which will be described in details in this Section.

The **penalty functions** are often used, because the other approaches are hard to be modeled or/and need derivatives (Eiben & Smith, 2003). In addition, the **exterior penalty function (EPF)** is preferred in constrained EAs, because the **interior penalty function (IPF)** requires feasible individuals which in turns complicate the solution of the proposed constrained problem in EAs. As a result, this Section will focus mainly on both the exterior penalty approaches and the **classical random search**, which is classified as one of the **direct search methods**.

By referring to Eqs.(2.1)-(2.3), EPFs can easily transform them into unconstrained optimization problems by using **additive** and **multiplicative** forms, and can be respectively described as (Yeniay, 2005):

$$\min_X \phi(X), \ \ where \ \ \phi(X) = \begin{cases} f(X), & \text{if } X \in \mathcal{F} \\ f(X) + P(X), & \text{if } X \notin \mathcal{F} \end{cases} \tag{2.4}$$

$$\min_X \phi(X), \ \ where \ \ \phi(X) = \begin{cases} f(X), & \text{if } X \in \mathcal{F} \\ f(X) \times P(X), & \text{if } X \notin \mathcal{F} \end{cases} \tag{2.5}$$

where $P(X)$ is called a **penalty term**, which is equal to zero for feasible individuals, and be a positive value in case any one of the constraints is

violated. Thus, the **penalized cost function** $\phi(X)$ becomes higher than its actual value in $f(X)$. This $P(X)$ can be provided in different forms based on the type of the penalty functions. The most common form is:

$$\begin{aligned} P(X) &= \sum_{q=1}^{m} r_q \, \hat{g}_q(X) + \sum_{p=1}^{l} d_p \, \hat{h}_p(X) \\ \text{where: } \hat{g}_q(X) &= [max\,(0, g_q(X))]^{\beta} \\ \hat{h}_p(X) &= |h_p(X)|^{\gamma} \end{aligned} \tag{2.6}$$

where r_q and d_p are called the **penalty multipliers**, and β and γ are user-defined positive constants which are commonly set equal to either 1 or 2 (Simon, 2013; Rao, 2009; Eiben & Smith, 2003).

Because $[H]_l = 0$ can not be easily satisfied, so an acceptance amount of tolerance "$\pm\ \epsilon$" is used instead of zero. Thus, the equality constrains are satisfied if:

$$-\epsilon \leqslant h_p(X) \leqslant \epsilon \ , \quad p = 1, 2, \cdots, l \tag{2.7}$$

This equation can be split into two parts of inequality constraints. The standard format of them is:

$$\begin{aligned} h_p(X) - \epsilon &\leqslant 0 \\ -h_p(X) - \epsilon &\leqslant 0 \end{aligned} \tag{2.8}$$

Therefore, Using Eq.(2.8), Eq.(2.6) can be reduced down to:

$$P(X) = \sum_{q=1}^{m+l} r_q \, \hat{g}_q(X) \; ; \quad \text{where}$$

$$\hat{g}_q(X) = \begin{cases} [max\,(0, g_q(X))]^{\beta}, & \text{for } q \in [1, m] \\ [max\,(0, |h_q(X)| - \epsilon)]^{\beta}, & \text{for } q \in [m+1, m+l] \end{cases} \tag{2.9}$$

Before describing the selected exterior penalty functions, it is important to mention that the penalized cost function "ϕ" is not applicable with the selected random search method. This classical direct method depends only on an internal loop that generates random elements of the design vector $[X]$. These ten selected constraint-handling techniques are described in the following lines:

2.3.1 Random Search Method

It is very simple, and it can be easily programmed through the pseudocode in Algorithm 1.

Unfortunately, this technique is not efficient and consumes high

Algorithm 1 rebuild unfeasible individuals directly by using random search technique

Require: all the constraints' values of each individual as a vector C

1: **for** $j \leftarrow 1$ *to* $\bar{k}$ **do** {where $\bar{k}$ = population size}
2: **while** any element of C_j is violated **do** {there are $\bar{k}$ vectors of C}
3: Randomly generate new design vector X
4: Determine new C_j
5: **end while**
6: **end for**

CPU time as the problem's complexities increase (Rao, 2009). These complexities could be the problem's dimension, its type (convex or non-convex, explicit or implicit, etc), type of design variables (continuous, discrete, binary, etc), number of design constraints, type of design constraints, range of the search space, etc.

2.3.2 Constant Penalty Function

In this type, the penalty term $P(X)$ is set to a very high value when any constraint is violated. During function evaluation process, this intensive penalty will reject all the infeasible individuals. This is why it is called "**death penalty**" (Rao, 2009; Simon, 2013).

The drawback of this simple type of the exterior penalty functions is that when the infeasible individuals are near to a feasible space or if they have some good information that can guide the algorithm to see the feasible space, this death exterior penalty approach will erase all of these useful data. Thus, both the individuals with little or with many violations will be treated by the same rejection action, and all the infeasible individuals will completely disappear within the next generation (Eiben & Smith, 2003).

The infinite barrier penalty in (Deb, 2010; Yeniay, 2005) is considered here. It comes in an additive form, where $P(X)$ in Eq.(2.9) is calculated as:

$$P(X) = R \sum_{q=1}^{m+l} \hat{g}_q(X) \tag{2.10}$$

where the penalty multiplier "r_q" is replaced with a very large constant number "R", which is usually set equal to 10^{20}. From this equation, the penalty term will go to almost infinity (i.e., $P(X) \approx +\infty$) if any violation is detected. Some of death penalty functions are defined with only R term, so the high value of $P(X)$ becomes a fixed value and independent of the violations magnitude.

2.3.3 Binary Static Penalty Function

It is an additive penalty, in which the distance function $\hat{g}_q(X)$ in Eq.(2.9), which represents the constraint dependent function, can be a simple binary value equal to 1 if the constraint is violated, otherwise it becomes zero (Eiben & Smith, 2003). If r_q is taken as a constant value for all $(l + m)$ constraints, then the final expression of this penalty function can be expressed as:

$$P(X) = r \sum_{q=1}^{l+m} \hat{g}_q(X), \quad \hat{g}_q(X) = \begin{cases} 1, & \text{if } \hat{g}_q(X) \text{ is violated} \\ 0, & \text{if } \hat{g}_q(X) \text{ is satisfied} \end{cases} \tag{2.11}$$

Now, the dimension of the vector r becomes one, which is easy to be tuned. In contrast, the other static penalty functions are hard to be tuned, see Fig.(2.2). In addition, the binary static can compromise between the speed and accuracy, which are very important factors in the engineering applications. Based on that, this binary type will be used to solve the optimal coordination problems of the protective relays. Its performance

will be shown in Chapter 5.

2.3.4 The Superiority of Feasible Points (SFP) - Type I

In this method, the penalized cost function of Eq.(2.4) is expressed as:

$$\begin{aligned} \min_X \phi'(X) &= \phi(X) + \theta(X) \\ \text{where: } \theta(X) &= \begin{cases} 0, & \text{if } \mathcal{F} = \emptyset \text{ or } X \in \mathcal{F} \\ \alpha, & \text{if } \mathcal{F} \neq \emptyset \text{ and } X \notin \mathcal{F} \end{cases} \end{aligned} \tag{2.12}$$

The value of α is the biggest feasible individual:

$$\alpha = max\,[f(Y)] \;:\; Y \in \mathcal{F} \tag{2.13}$$

This is done to make sure that the bad feasible individual is better, or at least, equal to the best infeasible individual (Yeniay, 2005), where the penalty multiplier r_q, in Eq.(2.9), can be set as a fixed number (Michalewicz, 1995). In this thesis, r_q is set equal to 10.

2.3.5 The Superiority of Feasible Points (SFP) - Type II

SFP-I does not cover the range when $f(X) < 0$, while SFP-II does. The second type has similar expression, except the value of α. Here, it is modified to be equal to the difference between the biggest feasible

and the smallest infeasible individuals as (Simon, 2013; Kajee-Bagdadi, 2007):

$$\alpha = max\left[0, max_{Y\in\mathcal{F}} f(Y) - min_{Y\notin\mathcal{F}}\phi(Y)\right] \tag{2.14}$$

2.3.6 The Eclectic Evolutionary Algorithm

The bad thing of the extinctive static penalty function, in Fig.(2.2), is that the solution quality is very sensitive to the values of the penalty multiplier vector "r_q" (Yeniay, 2005). The eclectic EA works in a similar principle of SFP types, where the bad feasible individuals are considered to be better than the best infeasible individuals, but in different approach (Simon, 2013). Instead of using the amount of constraint violation, it uses the number of violated constraints as an information to penalize the cost function as:

$$\phi(X) = \begin{cases} f(X), & \text{if } X \in \mathcal{F} \\ K\left[1 - \dfrac{v(X)}{(m+l)}\right], & \text{if } X \notin \mathcal{F} \end{cases} \tag{2.15}$$

where K is a very large positive constant (*it is taken as* $K = 1 \times 10^9$ (Yeniay, 2005)), $v(X)$ is the number of satisfied or non-violated constraints and $(m + l)$ is the total number of constraints.

2.3.7 Typical Dynamic Penalty Function

If the soft counter that counts the number of utilized generations inside the optimization algorithm is taken as a time "t", then the amount of

the penalized value is increased as the number of generations increases. This action will provide two things:

- At the beginning of the optimization, the penalization process will forgive the infeasible individuals with small penalized values. Thus, the algorithm will collect some useful information about the search space of the optimized system.
- As the number of generations increases, that penalized values will also increase to force each individual to go inside the feasible search space.

The penalized cost function is defined as (Rao, 2009; Simon, 2013; Eiben & Smith, 2003):

$$\phi(X) = f(X) + (c.t)^{\alpha} P(X) \tag{2.16}$$

where c and α are constants. In this study, they are taken as 0.5 and 2, respectively (Simon, 2013). $P(X)$ is similar to that of Eq.(2.9), but with $r = 1$ and $\beta = 2$.

2.3.8 Exponential Dynamic Penalty Function

Instead of an additive form as in Eq.(2.16), this type comes in a multiplicative form. It considers the same assumptions, except that the penalty value will grow in an exponential rate rather than a linear rate

as in the typical dynamic approach. The new penalization is defined as (Rao, 2009; Simon, 2013; Eiben & Smith, 2003):

$$\phi(X) = f(X) \times e^{\frac{P(X)}{T}} \tag{2.17}$$

where $T = \frac{1}{\sqrt{t}}$, which approaches zero as the number of generations approaches infinity (i.e., $t \to \infty \Rightarrow T \to 0 \Rightarrow \phi(X) \to \infty$).

This type of dynamic penalties is valid just for minimization, where $f(X) \geqslant 0 \; \forall \; X$. Otherwise, the normalized version, as described in (Simon, 2013), has to be used instead. For relays' coordination problem, Eq.(2.17) can be used.

2.3.9 Adaptive Multiplication Penalty Function

From its name, it comes in multiplicative form. The cost function is penalized as (Yu & Gen, 2010):

$$\phi(X) = f(X) \times \left\{ 1 + \frac{1}{(m+l)} \sum_{q=1}^{m+l} \left[\frac{\hat{g}_q(X)}{\hat{g}_q^{max}(X)} \right]^{\alpha} \right\} \tag{2.18}$$

There are many adaptive approaches available in the literature. The first one was proposed by (Hadj-Alouane & Bean, 1997). Among the various types of adaptive penalties, the adaptive multiplication approach is selected because it solves the associated problems that come with the typical feedback and near feasibility threshold (NFT-based) approaches (Yeniay, 2005). The original equation contains subtraction operator, while

Eq.(2.18) has addition operator. This adjustment is essential to let this penalty function to be embedded for minimization problems instead of maximization. $\hat{g}_q(X)$ can be calculated by using Eq.(2.9) with $\beta = 1$, and $\hat{g}_q^{max}(X)$ is the biggest element of $\hat{g}_q(X)$ vector, which can be obtained as:

$$\hat{g}_q^{max}(X) = max\left[\epsilon, max\left(\hat{g}_q(X)\right)\right] \tag{2.19}$$

Epsilon in Eq.(2.19) is very important to avoid dividing by zero when all the individuals are feasible. For example, ϵ can be set equal to 10^{-20}. Thus, if there is no infeasible individual, then the summation term becomes zero instead of infinity.

2.3.10 Self-Adaptive Penalty Function (SAPF)

The algorithm of SAPF is based on two conditions (Simon, 2013):

- ◇ If the percentage of the feasible individuals out of the population size is low, then the penalized cost function "$\phi(X)$" has to be low for those infeasible individuals that have few violations.
- ◇ If that percentage of feasible individuals becomes high, then the low penalized cost function has to be assigned only to those infeasible individuals that have low cost "$f(X)$".

Thus, in order to identify the best individual in the current population, SAPF penalizes infeasible individuals by two terms, $d(X)$ and

$P(X)$ as (Sivananaithaperumal, Amali, Baskar, & Suganthan, 2011):

$$\phi(X) = d(X) + P(X) \tag{2.20}$$

where $d(X)$ is called the distance value. SAPF needs a special care for constructing Eq.(2.20). The steps can be summarized as (Simon, 2013; Sivananaithaperumal et al., 2011):

- Firstly, normalize $f(X)$ for each individual as:

$$N(X) = \frac{f(X) - f_{min}}{f_{max} - f_{min}} \tag{2.21}$$

  where $N(X) \in [0, 1] \, \forall \, X$; the best individual occurs when $N(X) = 0$, and vice versa when $N(X) = 1$.

- Secondly, compute the normalized magnitude of the constraint violations $M(X)$ as:

$$M(X) = \frac{1}{m + l} \sum_{q=1}^{m+l} \left(\frac{\hat{g}_q(X)}{\hat{g}_q^{max}(X)} \right) \tag{2.22}$$

  where $\hat{g}_q(X)$ and $\hat{g}_q^{max}(X)$ can be obtained in similar way as those of the preceding penalty function. Thus, $M(X) = 0 \, \forall \, X \in \mathcal{F}$ and $M(\bar{X}) > 0 \, \forall \, \bar{X} \notin \mathcal{F}$.

- Thirdly, for each individual, compute the distance $d(X)$ as:

$$d(X) = \begin{cases} M(X), & \text{if } \mathcal{F} = \emptyset \\ \sqrt{N^2(X) + M^2(X)}, & \text{if } \mathcal{F} \neq \emptyset \end{cases} \tag{2.23}$$

- Fourthly, the penalty term in Eq.(2.20) can be calculated as:

$$\begin{aligned} P(X) &= (1-r)Y_1(X) + rY_2(X) \\ \text{where: } Y_1(X) &= \begin{cases} 0, & \text{if } \mathcal{F} = \emptyset \\ M(X), & \text{if } \mathcal{F} \neq \emptyset \end{cases} \\ Y_2(X) &= \begin{cases} 0, & \text{if } X \in \mathcal{F} \\ N(X), & \text{if } X \notin \mathcal{F} \end{cases} \\ r &= \frac{number\ of\ feasible\ individuals}{population\ size} \end{aligned} \tag{2.24}$$

As can be clearly seen from all these equations, Eqs.(2.20)-(2.24), one of the main disadvantages of SAPF is that it consumes more CPU time than the other penalty functions.

The performance evaluation of these ten constraint-handling techniques will be covered later in Chapter 5.

2.4 Performance Criteria for Evaluating Algorithms

In order to evaluate the performance of any optimization algorithm, it is necessary first to define the performance criteria based on which the

algorithm's performance can be assessed and compared with other algorithms. The first stage is to run the algorithm to solve some standard **benchmark functions** (also known as **test functions**). These functions are classified into two main parts: **unimodal** (*having one optimum solution*) and **multimodal** (*having multiple optimum solutions*). Also, they can be classified as: **unconstrained**, **constrained**, **multi-objective**, **dynamic** and **noisy** benchmark functions.

To evaluate the performance, the algorithm should be run randomly many times in order to collect the obtained fitness for each run, so that the **best** and **worst** final solutions, and the **mean** and **standard deviation** of these solutions can be extracted. By these statistical tools, the obtained results can be compared with other results in the literature. It is important to use similar parameters (*i.e., population size, number of generations, etc*) in order to get a fair comparison. More performance evaluation tools are available in (Bartz-Beielstein, 2006) with extensive description and new ideas.

For the minimization problems, the best of the best and worst of the best obtained solutions per certain number of simulation runs can be respectively computed as follows:

$$f_{best\_ever} = min\left(f_{best,1}, f_{best,2}, \cdots, f_{best,i}, \cdots, f_{best,T_r}\right) \quad (2.25)$$

$$f_{worst\_ever} = max\left(f_{best,1}, f_{best,2}, \cdots, f_{best,i}, \cdots, f_{best,T_r}\right) \quad (2.26)$$

where T_r is the total number of trials and $f_{best,i}$ represents the best ob-

tained solution for the ith trial. Also, the mean and standard deviation can be calculated as follows (Johnson, 2000):

$$f_{mean} = \frac{\sum_{i=1}^{T_r} f_{best,i}}{T_r} \tag{2.27}$$

$$f_{std\_dev} = \sqrt{\frac{\sum_{i=1}^{T_r} (f_{best,i} - f_{mean})^2}{T_r}} \tag{2.28}$$

If the global optimal solution (f^*) is given in the data of the standard benchmark functions, then the absolute error (e_{abs}) between the estimated global optimal solution ($f_{best,i}$) and the exact solution (f^*) can be calculated as:

$$e_{abs,i} = |f_{best,i} - f^*| \tag{2.29}$$

Thus, Eqs.(2.25)-(2.28) will respectively replaced by:

Smallest error : $$e_{smallest\_ever} = min\,(e_{abs,1}, e_{abs,2}, \cdots, e_{abs,T_r}) \tag{2.30}$$

Largest error : $$e_{biggest\_ever} = max\,(e_{abs,1}, e_{abs,2}, \cdots, e_{abs,T_r}) \tag{2.31}$$

Mean error : $$e_{mean} = \frac{\sum_{i=1}^{T_r} e_{abs,i}}{T_r} \tag{2.32}$$

Standard deviation : $$e_{std\_dev} = \sqrt{\frac{\sum_{i=1}^{T_r} (e_{abs,i} - e_{mean})^2}{T_r}} \tag{2.33}$$

In addition to the above criteria, the computation speed of the algorithm can also be used as a performance criteria.

Chapter 3

Fundamentals of Power System Protection

3.1 Faults Classifications

3.2 Protection System

3.3 Zones of Protection

3.4 Primary and Backup Protection

3.5 Performance and Design Criteria

3.6 Overcurrent Protective Relays

3.7 Directional Overcurrent Protective Relays

3.8 Coordination of Directional Overcurrent Relays

3.9 Case Study: Is the Coordination of DOCRs an Iterative Problem?

3.10 General Optimal Coordination of DOCRs

Chapter 3
Fundamentals of Power System Protection

Any electric power system consists of three principal parts:

1. Power Generation
2. Power Transmission
3. Power Distribution

The connection between these parts can be graphically described in Fig.(3.1).

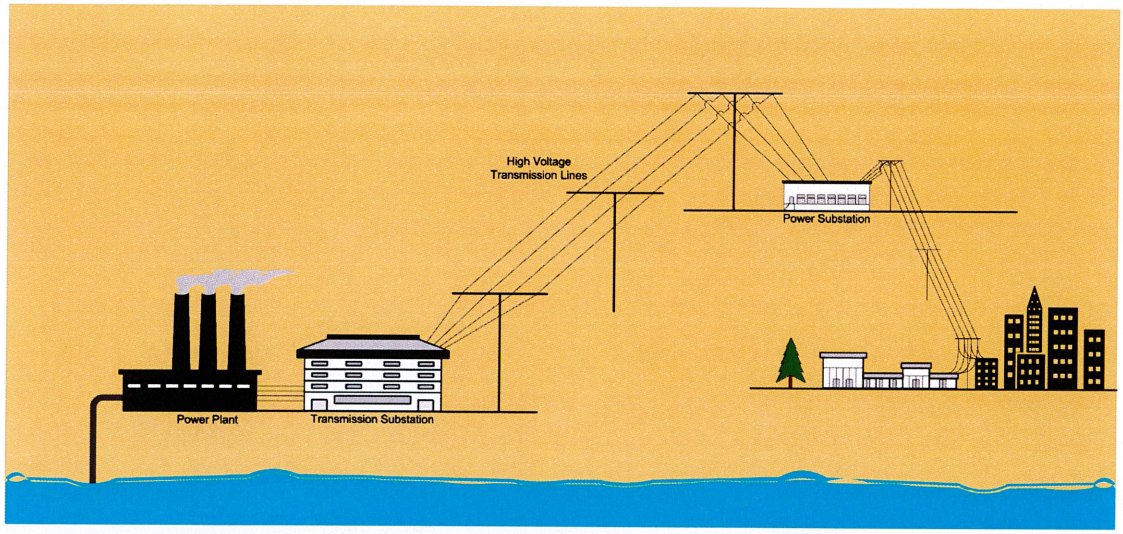


Figure (3.1)
General representation of complete power system cycle

In the past, the structure of these parts was very simple; and even if there is any power outage or load shedding, this will not be a big issue, since the dependability on the produced electricity at that time was very low.

With the modern civilization, a huge amount of electricity is consumed to power the houses' appliances, street lights, factories, transportation, communications, etc. Thus, the dependability becomes very high and no one can imagine how the human's life will be without the electrical energy. To meet these needs, the new systems are designed and managed in a way that the energy can be economically and reliably delivered to the utilization points (ALSTOM, 2002). Thus, the systems become very advanced and at the same time become very complicated to be operated and controlled without any failure. As a protection engineer, these failures are called "**faults**".

If any fault exists for a long time without clearance, then lots of bad scenarios could happen on the electric power system, like breaking of the cables' insulation due to the generated heat of the high current, damaging the devices and equipments (*or at least, reducing their prospective lifespan*), shortage in the delivered power to the customers, etc.

3.1 Faults Classifications

In the electric power systems, the fault means any abnormal flow of electric current. There are many causes to create faults, like: lightning strike, ice and snow storms, wind, flying objects, physical contact by animals, falling trees, human errors, insulation flashover, etc. The fault, as a major subject, could be categorized under two types:

1. **Open Circuit Fault** (or **Series Fault**)**:** it happens when the path of the current is interrupted.

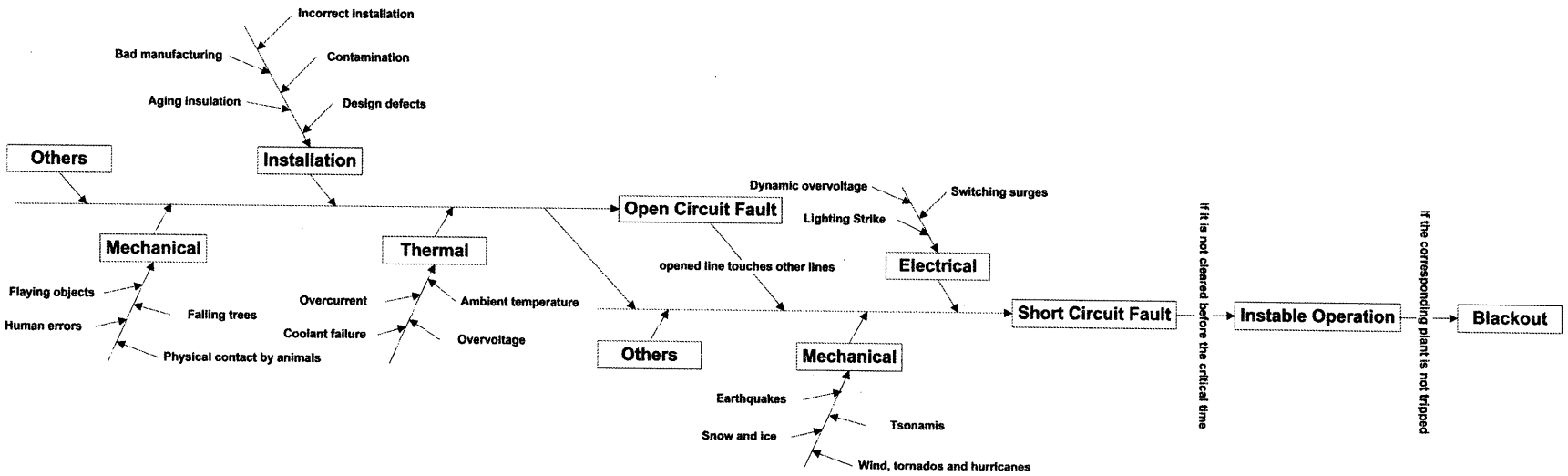


Figure (3.2)
Cause and effect diagram of different faults scenarios

2. **Short Circuit Fault** (or **Shunt Fault):** it happens when the flow of the current bypasses the load.

In most situations, the series fault does not lead into catastrophes except in some special cases. This type of faults could happen if one, or more of the lines, is broken without touching the other lines or grounded parts. Otherwise, it could lead to a shunt fault (Paithankar & Bhide, 2003).

The shunt fault is more dangerous, and it can create a big disaster into any part of the power system, or it can lead to cascading trips that would eventually leads to a system trip "blackout" if it is not cleared as fast and selective as possible, as shown in Fig.(3.2).

The faults can be classified as:

1. **Passive Fault:** it happens if the values of some variables are higher than the designed maximum limits. some examples are overloading, overvoltage, power swing, under frequency, etc.

2. **Active Fault:** it is a real fault, such as short circuits, and it needs to be cleared quickly.

Faults can also be classified into two types based on the fault duration as:

1. **Transient Fault** (or **Temporary Fault):** it is a fault that disappears by itself after short time. The main procedure to clear this fault is by switching-off the faulty part and then re-energizing the circuit again. This type of faults can decrease the lifespan of the conductors and/or equipments; but, it does not damage them usually. In many cases, the transient faults happens in the overhead transmission/subtransmission lines as a result of several causes such as momentary tree contact, bird or other animal contact, conductors contact due to wind or snowfall, insulator flashover due to lightning strike, etc. In practice, an automatic recloser is used to return the temporary faulty cable back into the service.

2. **Persistent Fault** (or **Permanent Fault):** the more dangerous scenario is that when it happens with a persistent fault, because that fault will remain even with opening and reclosing the circuit. If this type of faults happen, a quick trip action is urgently needed. Some examples of such faults on the overhead lines are big flying objects, human errors, insulation flashover, etc. The most common cause of such fault on the underground cables are due to mechanical damage.

For a three phase (3ϕ) system, there are five types of shunt faults as shown in Fig.(3.3) (Hewitson, Brown, & Balakrishnan, 2004).

In practice, phase-to-ground fault is the most probable type of faults, whereas three-phase fault is the most severe fault as shown in Table 3.1.

Table (3.1)
The probability of faults occurrence corresponding to their types (Blackburn & Domin, 2006; Bandyopadhyay, 2006)

Fault Type	Probability of Occurrence (%)	Severity Level
Single phase-to-ground	70 - 80	Lowest
Phase-to-phase	10 - 8	
Phase-to-phase-to-ground	17 - 10	
Three-phase	3 - 2	Highest

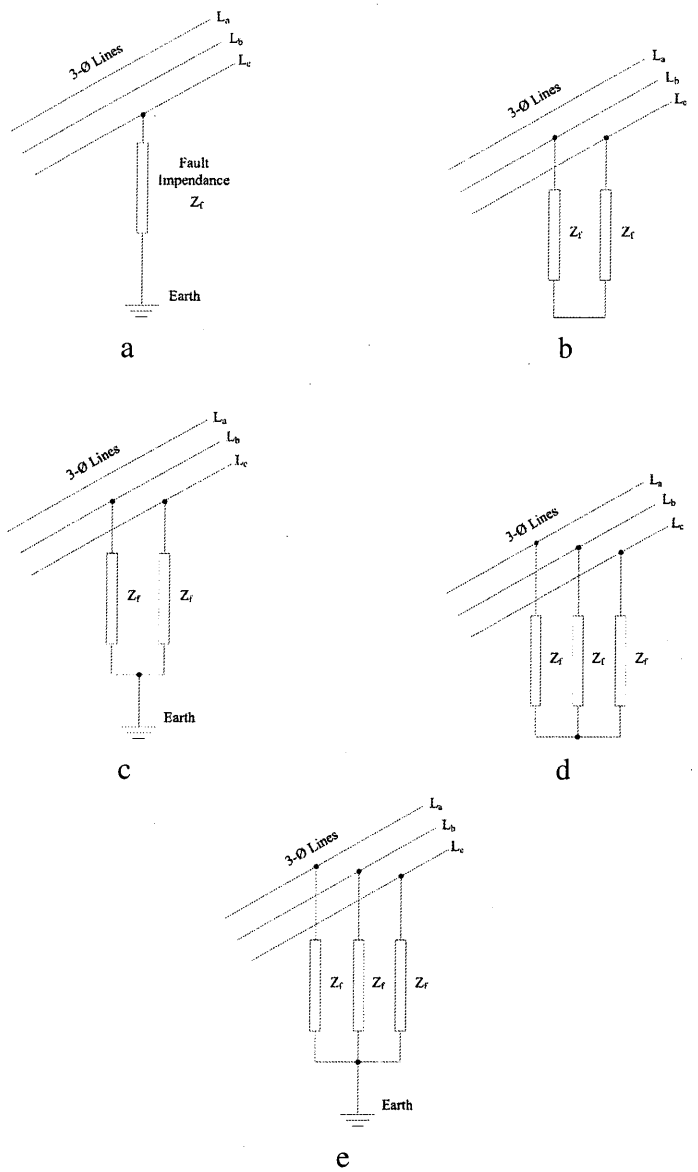


Figure (3.3)
The possibilities of active short circuit fault. (a) L-G Fault, (b) L-L Fault, (c) L-L-G Fault, (d) L-L-L Fault, and (e) L-L-L-G Fault

These classifications of the active faults are grouped into two groups:

Table (3.2)
The probability of faults occurrence on different elements of the power systems

Power System Element	Probability of Occurrence (%)
Overhead lines	50
Underground cables	9
Transformers	10
Generators	7
Switchgears	12
CT, PT, etc	12

1. **Symmetrical Faults** (or **Balanced Faults):** there are only two possibilities of balanced faults, which are the last two faults (i.e., *three-phase 3ϕ faults*) of Fig.(3.3). They are named symmetrical faults, because each phase has equal sinusoidal waves (i.e., *voltage* and *current*) on the axis and are shifted by 120° between each other. Thus, the self impedances of these three lines are equal; and at the same time, if there are any mutual impedances between these lines, then their values will be the same too.

2. **Asymmetrical Faults** (or **Unbalanced Faults):** the remaining of the five fault possibilities are classified as unbalanced faults. During these types of faults, the phases are not affected equally, where the magnitudes and angles of their voltages, currents and impedances will not be same anymore.

Table 3.2 shows the probability of faults occurrence on different elements of the power system (Paithankar & Bhide, 2003). It shows that faults in overhead lines represents 50% of all faults in other elements in the power system.

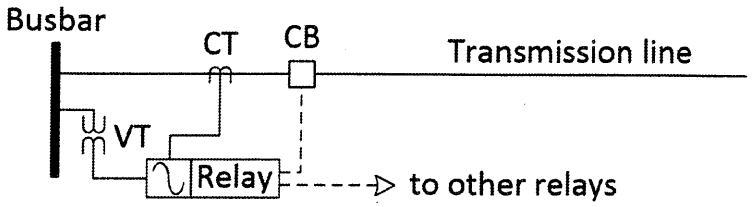


Figure (3.4)
Illustrated 1ϕ protection system

3.2 Protection System

As a sense of engineer, eliminating all the causes or sources of these faults is impossible mission due to many factors, like uncontrolled abnormal events, and the high cost of a highly secured and reliable system, especially if some of these faults are not pre-defined or rarely happen. Instead, the practical and relatively economic approach is to eliminate the effects of these causes (i.e. *the faults themselves*) by fast detection and clearing of the faults. By fast clearing action, any generated fault can be prevented from jeopardizing system integrity and its consequences can be mitigated effectively and the system can be preserved from any instable operating condition (Padiyar, 2008; Grainger & Stevenson, 1994; Glover, Sarma, & Overbye, 2010).

The faults can be detected easily if the system's voltages and/or currents are accessible and measurable, because from these fundamental signals all the required information can be extracted. These two fundamental signals need to be measured quickly and accurately. To accomplish this task, a **current transformer (CT)** and a **potential transformer (PT)** - also known as a **voltage transformer (VT)** - are used here, as shown in

Fig.(3.4). In addition, there are two stages of step down current and voltage quantities, the first one or the huge stepping is done through these main transformers, while the second stage is done through **auxiliary transformers** that can be located inside the protective relay itself. There are some issues that have to be taken into account regarding the received signals, like the effects of the decaying DC components, harmonic and CT saturation.

The protective relays can be classified according to their available technology:

- **Hardware-Based Protective Relays:**
  - Electromechanical (or Electromagnetic) Relays
  - Solid-State (or Static) Relays
  - Digital Relays
- **Numerical** (or **Processor-Based**) **Protective Relays:**
  - Microprocessor-Based Relays
  - DSP-Based Relays

The electromechanical relays are considered as the first generation of the protective relays that appeared since long time (1900s). The main idea behind their operations is based on a physical fact that when a current flows through a coil, it generates a magnetic field that can attract an iron rod, and the force of this attraction is proportional to the flowing current. This type of relays has many limitations, like its moving parts need periodic maintenance and calibration. Also, they give limited

setting values with some delays due to the condition of the **over-travel delay** (also known as **over-shot delay** or **coasting time**) that caused by the inertia. Moreover, they can not be connected to each other, and the designer needs three relays to protect the 3ϕ lines plus a ground relay, see Fig.(3.5). The main advantage of these relays is their stability and not sensitive to the network's condition. In addition, the protection engineers gained lots of experiences to deal with this kind of relays for long time; this is why they are still used as **local backup relays**. (McCleer, 1987; Paithankar & Bhide, 2003; Bandyopadhyay, 2006).

To overcome the problems of the mechanical moving parts as in the preceding technology and to increase the accuracy, the analog electronic devices were invented in order to do the same assigned protection tasks. The listed relays under this technology are called solid-state relays. The main disadvantages of these relays is that their operation can be affected by the surrounded temperature and they need very high precision components (like *resistors*, *capacitors*, *etc*) (Gers & Holmes, 2004).

Because of all the previous problems and because the development in the field of electronics, the solid-state relays did not stand more than about one decade (in the period between 1960s and 1970s) after which the third generation appeared. The last one is called digital relay, because it is built based on the digital electronic components. Therefore, no more mechanical moving parts or any deviation due to the surrounded temperature. As a comparison, the digital relays provide high accuracy and require shorter time to give their final decision, which are very hard to be

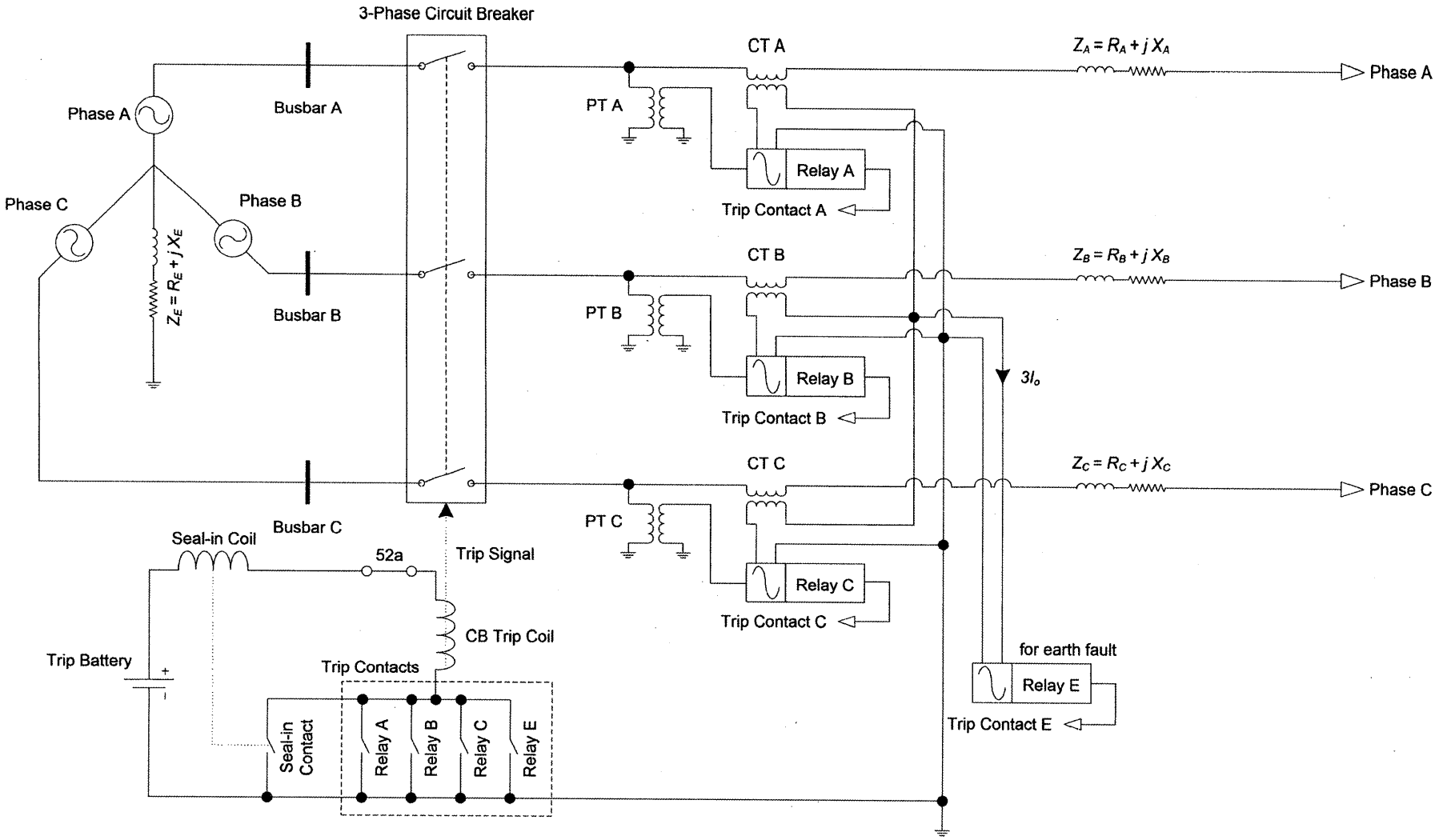


Figure (3.5)
The general 3ϕ protection system

integrated within the analog relays since they need high-precision passive and active electronic components and also need accurate tuning. In addition, by utilizing the latest improvements on the communication technology at that time, the manufacturers (like *SIEMENS*, *AREVA*, *ALSTOM*, *GE*, *ABB*, etc) reached to techniques that let these relays to communicate between each other through some special protocols and mediums (Mason, 1956; ALSTOM, 2002; Elmore, 1994).

Although, the digital relays solved all the inconvenient problems of the electromechanical and solid-state relays, they are still hardware-based relays. The fourth generation relays are the microprocessor-based relays, which are also known as numerical relays. The operation and settings of these relays can be updated easily through a software (Gers & Holmes, 2004).

Also, because the relays manufacturers found the numerical relays share the same hardware (*analog* and *digital inputs modules*, *auxiliary CTs and PTs*, *low pass filters*, *multiplexer*, *analog to digital converter*, *microprocessor*, *RAM and ROM*, *digital output module*, *communication card*, *power supply*, etc), a new proposed relay called **general purpose relay** (also known as **multifunctions relay**) is presented (Wood, Sidhu, Sachdev, & Nagpal, 1989). Thus, by uploading a new algorithm, the complete operation philosophy of the relay can be changed, see Fig.(3.6). For high processing computations, digital signal processor "DSP" was successfully used and the device is called DSP-based relay (Johns & Salman, 1995; Rebizant, Szafran, & Wiszniewski, 2011).

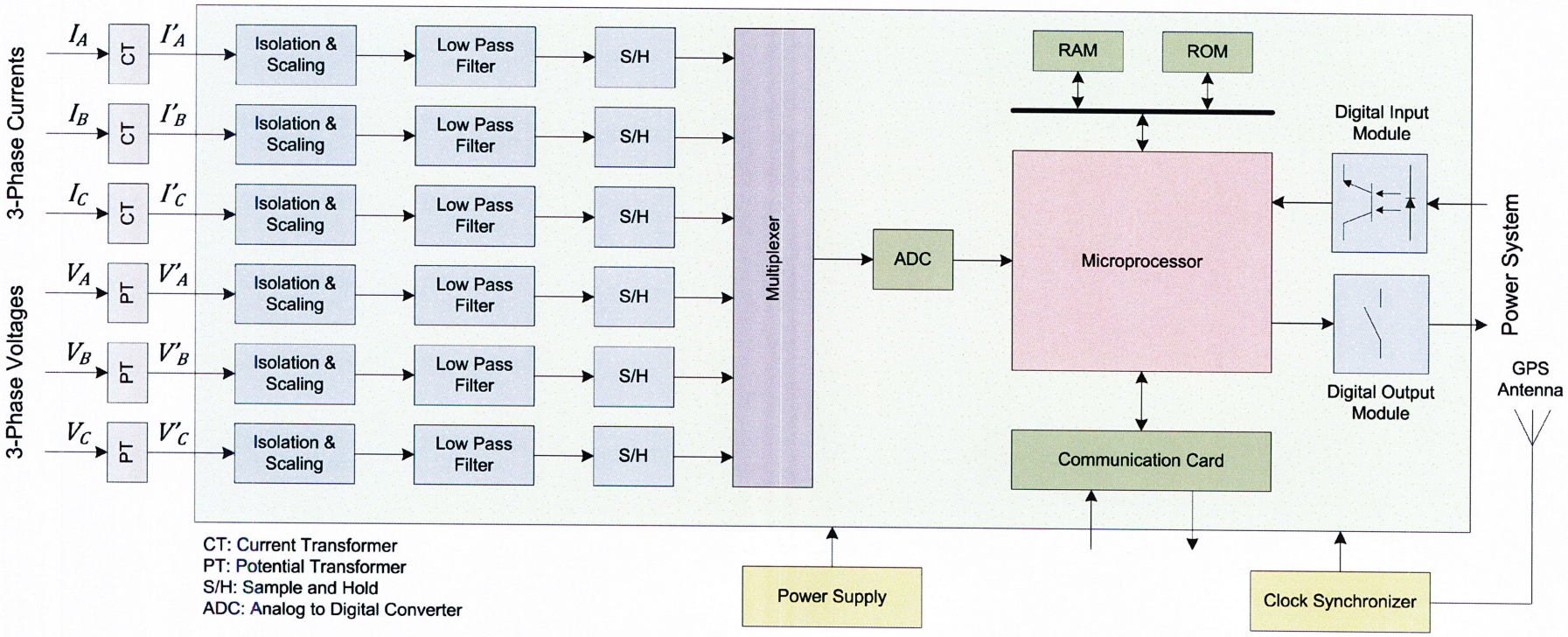


Figure (3.6)
The general hardware of a numerical relay

Except for the general purpose protective relay, there are many types of protective relays that can be classified according to their working principles:

- Distance Relays
- Overcurrent Relays
- Differential Relays
- Over/Under Voltage Relays
- Over/Under Frequency Relays
- Reverse Power Relays
- Overflux Relays
- etc

3.3 Zones of Protection

The power system is divided into zones in order to make the designed protection scheme adequate and to minimize the total number of isolated components. Each zone can be individually protected against their corresponding faults. If the faults are located inside the zone, then they are called **in-zone faults** (or **internal faults**), otherwise they are considered as **out-zone faults** (or **external faults**). These zones can be described through a simple system as shown in Fig.(3.7). Note that the overlaps between the zones are important in order to make sure that all spots in the power system are protected (Elmore, 1994; Gers & Holmes, 2004).

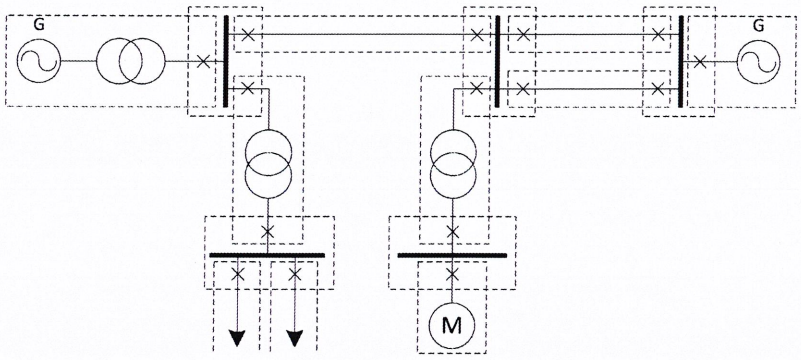


Figure (3.7)
Zones of protection

Practically, these zones are protected by using combinations of different protective relays, like these listed at the end of Section 3.2. For example, power transformers are protected by a group of relays, like differential, overcurrent, overflux, etc.

3.4 Primary and Backup Protection

The good protection system design can be created if the zones have number of primary relays that act as a first wall of protection for the corresponding in-zone faults, and backup relays that act as a second wall of protection for out-zone faults; whereas these backup relays could be installed locally on the same location of the primary relays and will see the same internal faults (they are known as **local backups**) or could be installed in other zones and will see that faults as external faults (they are known as **remote backups**). These backup relays are very important, because in the real life the probability of failure on any one of the primary protective relays always exists, which means that the second line of defense has to be ready all the time. Some examples will be taken later

to show how to select the corresponding backup relays for each primary relay.

3.5 Performance and Design Criteria

There are some general criteria that has to be fulfilled in order to accomplish the assigned protection task. They can be classified as (Gers & Holmes, 2004; Elmore, 1994; Blackburn & Domin, 2006):

3.5.1 Reliability

Reliability of a protection system is defined as its ability to operate correctly. Reliability has two aspects:

3.5.1.1 Dependability

The protective system can be designed to be very dependable if it operates correctly for in-zone faults.

3.5.1.2 Security

The protective system can be designed to be very secure if it does not operate incorrectly for out-zone faults.

3.5.2 Sensitivity

The protective system should detect any fault under any level of severity. The good design should take into account both the lowest and highest faults in order to set the protective relays correctly.

3.5.3 Speed

Isolating the fault as quick as possible is very important issue, and it is one of the reasons of the existing advanced technology in the filed of the protective relaying. But sometimes, an intentional delay is required to differentiate between the primary and backup protection, and to avoid misoperation due to load fluctuations.

3.5.4 Selectivity

If the correct set of relays are selected to clear the fault, then this action will enhance the system stability and increase the total amount of the delivered power to the end-users. The word "selectivity" can be shortly defined as: *maximum continuity of service with minimum system disconnection* (Blackburn & Domin, 2006).

3.5.5 Performance versus Economics

There is no doubt that the design of any protective system is better as the relays become faster with low inherent errors and as each zone is equipped by multiple types of relays, but the total cost will be very high. Therefore, the good design can be done if it balances between the performance and money.

3.5.6 Adequateness

One of the reasons of the high cost relays is that with multifunctions design. Instead of embedding all the features inside on relay, it is better to create variety of models for different assigned protection tasks.

3.5.7 Simplicity

If the protective system is designed in a simple way, then it will provide easy operation, maintenance, troubleshooting and future upgrading. The designed protection scheme can be accomplished by several ways with different complexities and options. Therefore, the simplest design does not always mean the cheapest design.

3.6 Overcurrent Protective Relays

Overcurrent protection can be achieved by the use of fuses or relays (Hodgkiss, 1995). OCRs are more suitable for large and complex networks, because the coordination between different devices becomes very hard. However, because fuses are cheap and available with different time responses, still they are used in combination with protective OCRs in some large networks.

In comparison with other expensive relays, overcurrent relays can compromise between the different design criteria, and this is why they are popular and widely used in power system protection (Elmore, 1994; Urdaneta et al., 1997).

OCRs have only currents as inputs from the power network. These input currents Is are provided by **current transformers (CTs)**. OCRs have two settings called **plug setting** (PS), also known as **pick-up setting** or **tap setting** in some references, and **time multiplier setting** (TMS), also known as **time dial setting** (TDS) or **time lever setting** in some

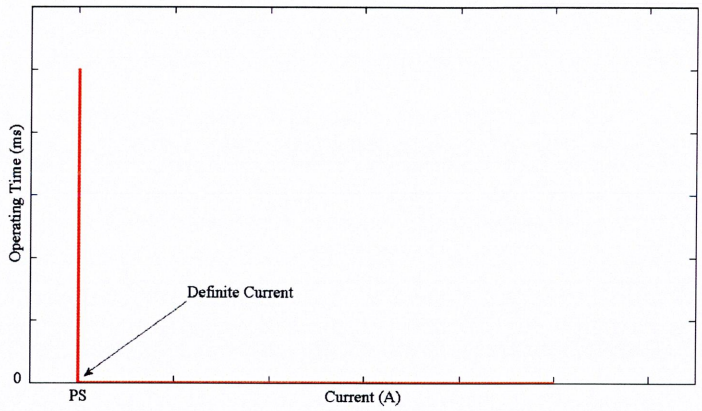


Figure (3.8)
Time-current characteristic curve of instantaneous overcurrent relay

references. The operating time of each relay is a function of these two inputs.

The severity of any fault that can be seen by OCR is defined as:

$$PSM = \frac{I_{relay}}{PS} \tag{3.1}$$

where PSM is a plug-setting multiplier, I_{relay} is a current seen by the relay referred to the secondary side of the CT. If $PSM < 1$, it means no fault exist in the system.

Based on the **time-current characteristic curves (TCCCs)**, OCRs can be categorized into three main types (Gers & Holmes, 2004):

1. **Instantaneous OCR** (or **Definite Current OCR**):

   This type of relays has only PS. The TCCC of the Instantaneous OCR is shown in Fig.(3.8), where the relay operates when the current exceeds its PS. This type of relays can not be applied when it has to differentiate between two faulty points if the source

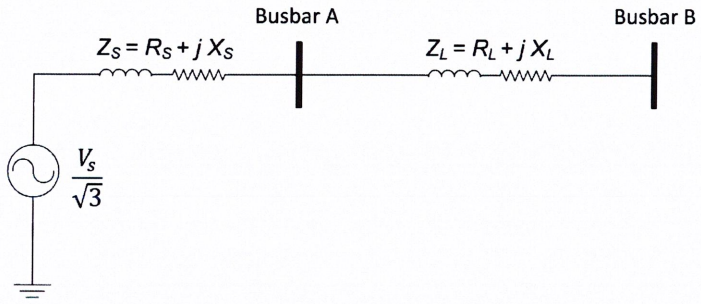


Figure (3.9)
Illustration of two bus radial network

impedance is much larger than the impedance between these two points. For example, if there are two faults, one at the busbar A and the other at busbar B of Fig.(3.9), then with $Z_S \gg Z_L$ the operating time for the nearest fault will be almost same that of the farthest fault.

2. **Definite Time OCR (DTOCR):**

   This type has plug setting (PS) and time delay setting (T). The relay operates only if it detects a fault current exceeding its current setting (PS) and the fault duration persisted for a time larger than its pre-specified time setting (T). Thus, the fault at different locations can be distinguished easily as shown in see Fig.(3.10). By referring to Fig.(3.9), the main disadvantage of this type is when the fault is located near the source, where higher fault current is expected, it will always wait for the predetermined time of delay setting (T) instead of taking instantaneous fast action to clear the fault. From that figure, Eq.(3.2) will decide if DTOCR should op-

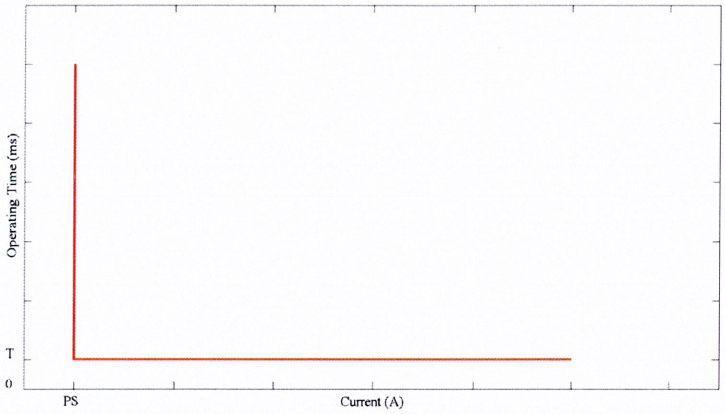


Figure (3.10)
Time-current characteristic curve of definite time overcurrent relay

erate or not as:

$$T = \begin{cases} D, & \text{if } PSM \geqslant 1 \\ \infty, & \text{otherwise} \end{cases} \tag{3.2}$$

where D is a predetermined time delay that DTOCR has to wait for.

3. **Inverse Time OCR:**

   This type of relays has both current setting (PS) and time multiplier setting (TMS). It can solve the preceding problems of DTOCR, because its operating time is inversely proportional to the value of the short-circuit current. The operating time can be expressed mathematically based on IEC and ANSI/IEEE standards as (IEC, 1989; IEEE, 1999):

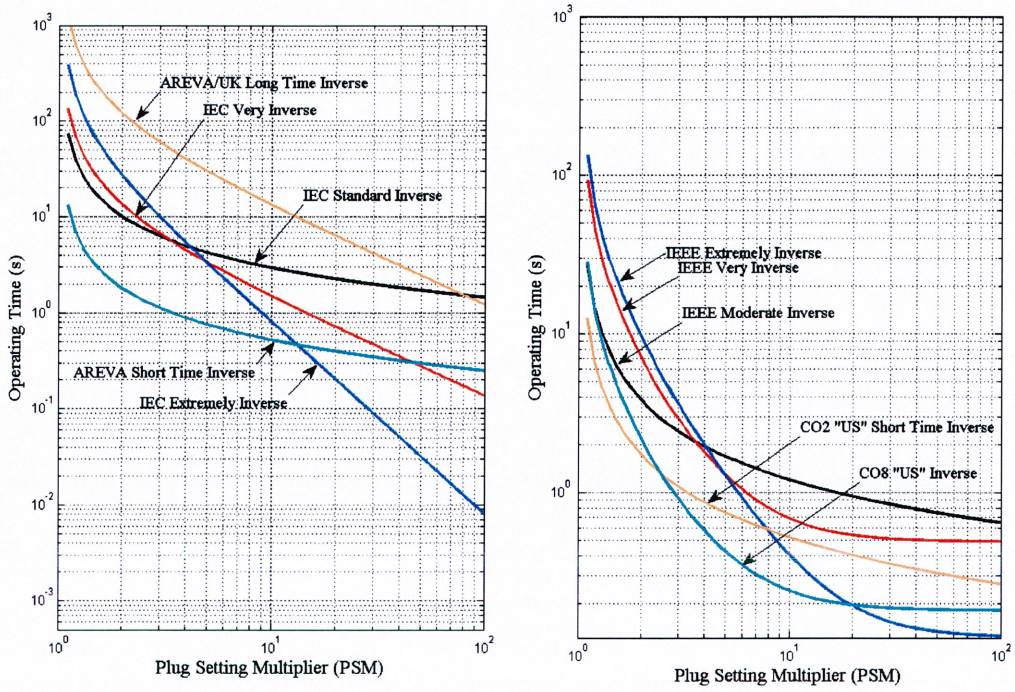
$$T = TMS \times \left[\frac{\beta}{PSM^{\alpha} - 1} + \gamma \right] \tag{3.3}$$

Table (3.3)
Standard values of inverse time overcurrent relays' constants

Type of Curve	Standard	α	β	γ
Moderately Inverse	IEEE	0.02	0.05150	0.11400
Very Inverse	IEEE	2.00	19.6100	0.49100
Extremely Inverse	IEEE	2.00	28.2000	0.12170
Inverse	CO8	2.00	5.95000	0.18000
Short-Time Inverse	CO2	0.02	0.02394	0.01694
Standard Inverse (IDMT)	IEC	0.02	0.14000	0.00000
Very Inverse	IEC	1.00	13.5000	0.00000
Extremely Inverse	IEC	2.00	80.0000	0.00000
Long-Time Inverse	AREVA / UK	1.00	120.000	0.00000
Short-Time Inverse	AREVA	0.04	0.05000	0.00000

where α, β and γ are constants. The values of these constants are tabulated in Table 3.3 for the most popular standards (AREVA, 2011; Gers & Holmes, 2004).

Figure (3.11) shows the log-log plots of these standard constant values that are listed in Table 3.3.



a: European Standards b: North American Standards

Figure (3.11)
Time-current characteristic curves of inverse time overcurrent relays (TMS=1.0)

Also for the same TCCC, the operating time can be varied by changing the value of TMS. Fig.(3.12) shows the log-log plot of the standard inverse characteristic curve (also called **Inverse Definite Minimum Time "IDMT"**) with different values of TMS.

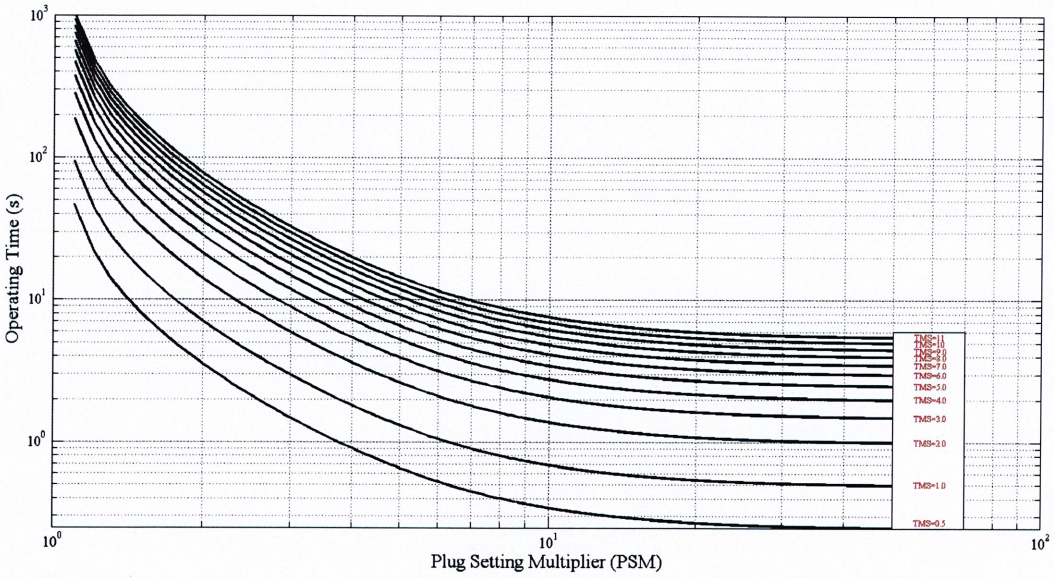


Figure (3.12)
Adjustable TMS of overcurrent relay with IDMT model

3.7 Directional Overcurrent Protective Relays

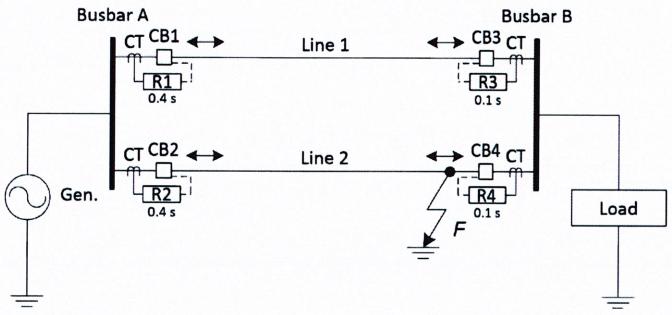


Figure (3.13)
Single-end fed power system of parallel feeders with only OCRs

The non-directional OCRs can detect the faults based only on the current magnitude of I and initiate the trip signal to their corresponding

circuit breakers to clear the faults. To know the weakness of these OCRs, consider the parallel line radial circuit shown in Fig.(3.13). In this example, suppose there is a fault F on the line 2 and near the busbar B. If only non-directional OCRs are used, then the overcurrent relays R_3 and R_4 will sense same fault current magnitude and send the trip signals at the same time to the circuit breakers CB_3 and CB_4, respectively. In addition, CB_2 will be tripped after a time delay by R_2 in order to clear the fault completely from the system. The problem with this protection scheme is that the connected load and line 1 will be unnecessarily disconnected and as a result the protection system is considered unreliable and unselective.

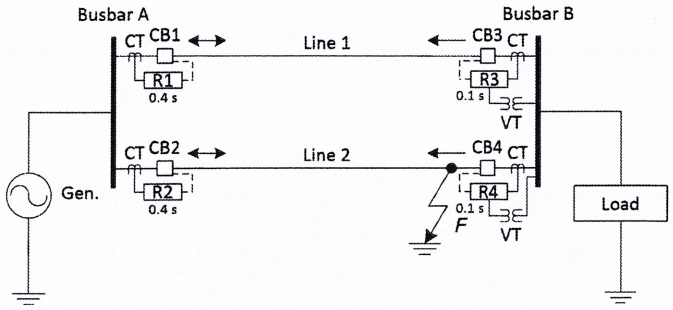


Figure (3.14)
Single-end fed power system of parallel feeders with OCRs and DOCRs

To solve this issue, an additional directional unit is combined with the general OCR to identify the direction of the fault current with respect to a reference signal. Based on that, both the current magnitude and direction are considered to trip the faulty element as fast and selectively as possible, where the remaining parts of the network can operate normally. This special protective device is called **directional overcurrent relay** (**DOCR**), which will be the focus of this thesis in studying the optimal coordination problem of DOCRs. In order to identify the direction of the

current, a **reference signal** (also known as a **polarization signal**) is provided as an input to the DOCR. usually, a voltage that can be provided through PT (*or, VT in some references*) is used as a polarizing signal. Current is also used as a polarizing signal in some applications in order to decrease the total cost. Except for instantaneous based DOCRs, these relays have both PS and TMS settings, and have similar TCCCs of the normal OCRs (Gers & Holmes, 2004).

Now, consider that R_3 and R_4 are directional OCRs which operates based on both magnitude and direction of the fault current as shown in Fig.(3.14). In this case, R_4 will sense both the magnitude and the correct direction of the short-circuit current and will trip CB_4. Similarly, R_2 will trip CB_2 after a time delay and hence the fault F is completely cleared from both ends of the line 2. On the other hand, R_3 will restrain (*i.e., not operate*), because the fault current direction is not same as its tripping direction despite that the fault current magnitude is same as that detected by R_4. Therefore, the continuity of the supply to the load through the healthy line 1 is preserved. Thus, in practical applications, DOCRs can be used as primary protection for interconnected sub-transmission and distribution systems, and as backup protection for transmission systems (Urdaneta et al., 1997).

Now, suppose that R_4 failed to operate for the fault F, then R_1 will act as a remote backup to clear the fault F by tripping CB_1 after a suitable time delay. Thus, the selectivity and reliability of a protection system are very important requirements in the field of power system protection.

For other complicated networks such as multi-loop systems, ring feeder systems, or even double-end fed power systems, depending only on non-directional OCRs is very complicated task and may not satisfy the reliability and selectivity requirements (Paithankar & Bhide, 2003).

3.8 Coordination of Directional Overcurrent Relays

The coordination of DOCRs can be shortly defined as "***the quality of selectivity among protective devices*** (Anthony, 1995)". It is very an important stage for any protection design. Correct relay coordination involves selecting the suitable relay setting that assures faults in the protected zone are cleared first by the corresponding primary relays and if they fail, the corresponding backup relays act after a **coordination time interval** (CTI), also known as **selective time interval (STI)** (Paithankar & Bhide, 2003) or **discrimination margin** (Gers & Holmes, 2004)), which can be calculated as:

$$CTI = T_{CB} + T_{OS} + T_{SM} \tag{3.4}$$

where T_{CB} is the operating time of the circuit breaker after receiving the trip signal from the primary relay. T_{OS} is the overshoot time (it is limited to the electromechanical relays, as mentioned in Section 3.2) of the primary relay. T_{SM} is the safety margin to take into account the error due to the relay timing error, CT ratio error, current magnitude measurement error, etc. The values of CTI range between 0.2 and 0.5 seconds

(Blackburn & Domin, 2006; Anthony, 1995).

Except for some special cases of radial networks, the coordination problem can be considered as a highly constrained **mixed-integer non-linear programming** (**MINLP**) problem, where TMS is continuous and PS is discrete (*in numerical relays they can be provided in almost continuous values*). Such problem requires an expert protection engineer to solve it analytically, where all faults, system contingencies and abnormalities are analyzed and predetermined. Alternatively, it can be solved easily using optimization methods (Noghabi et al., 2009).

3.9 Case Study: Is the Coordination of DOCRs an Iterative Problem?

This case study will solve the coordination problem of a very simple radial system and then standard IEEE 3-bus mesh system in order to show the iterative nature of non-radial systems and why it is important. In both examples, $CTI = 0.3s$ and only IDMT characteristic curve is used.

3.9.1 Coordination Problem of a Radial System

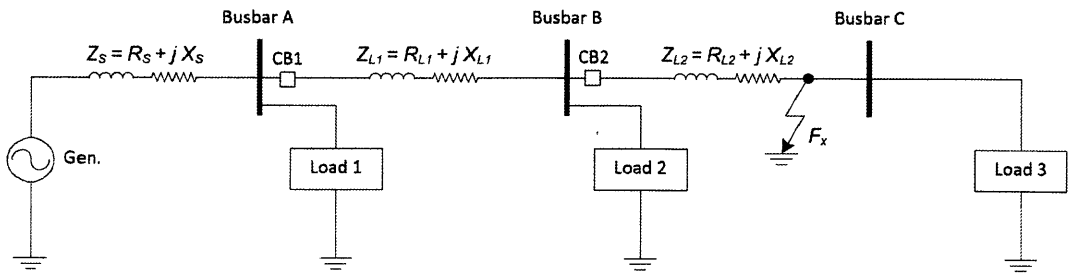


Figure (3.15)
Radial network with directional overcurrent relays

Consider the radial system shown in Fig.(3.15) with $TMS = 0.1s$. For fault at F_x, relay R_2 is the primary protection to clear that fault by tripping its circuit breaker CB_2 after a delay:

$$T_{R_2} = TMS_{R_2} \times \left(\frac{\beta}{PSM_{R_2}^{\alpha} - 1} + \gamma \right) = \left(\frac{0.14 \times 0.1}{PSM_{R_2}^{0.02} - 1} \right) second$$

The relay R_1 is a backup relay for R_2, so it should operate only if R_2 failed to clear F_x fault. If R_2 fails to operate, then R_1 has to isolate F_x with minimum operating time of:

$$T_{R_1} = T_{R_2} + CTI = \left(\frac{0.14 \times 0.1}{PSM_{R_2}^{0.02} - 1} + 0.3 \right) second$$

As seen from the radial system, R_2 will never be a backup relay for any fault located before busbar B. Thus, the coordination problem of this simple radial system can be solved easily in one pass.

3.9.2 Coordination Problem of a Ring System

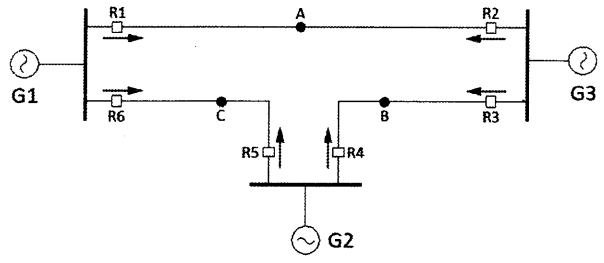


Figure (3.16)
Single-line diagram of the 3-bus system

Consider now the IEEE 3-bus ring system shown in Fig.(3.16) - *refer to Appendix D* - assuming fixed topology. For simplicity, numeri-

cal relays with very small value of $TMS^{min} = 0.025s$ and no limit on their minimum operating times are assumed available. From the figure of IEEE 3-bus system, DOCRs can detect short-circuit currents only in one direction, so there are actually two loops:

- **Clockwise Loop:** $R_1 \rightarrow R_3 \rightarrow R_5 \rightarrow R_1$
- **Anti-Clockwise Loop:** $R_6 \rightarrow R_4 \rightarrow R_2 \rightarrow R_6$

where arrows mean "***backup to***". The first loop starts with R_1 and ends with R_1 too. Similarly for the second loop, where it starts and ends with R_6. This means that the coordination problem is iterative, because the devices that are initially considered as primary relays will become backup relays for the other devices that are initially considered as backup relays. Thus, each DOCR could work as a primary relay as well as backup relay. Based on that, coordinating relays to operate accurately and selectively is a hard assigned job to the protection engineers.

The classical iterative technique is done by randomly selecting one relay from each loop. The first step is to guess its initial operating time by using a good starting point for its TMS, where its PS is predetermined by a good value that lies between the full load and minimum fault currents and using the engineering experiences and skills. As a result, Eq.(3.3) becomes a linear equation with one variable. All the predetermined values of PS are tabulated in Table 3.4.

Table (3.4)
Data of Case Study "3-bus system"

Relay No.	Plug Setting (A) "Primary Side of CT"	In-Zone Fault Current (A)	Out-Zone Fault Current (A)
R_1	180	1978.90	617.22
R_2	100	1525.70	145.34
R_3	100	1683.90	384.00
R_4	150	1815.40	545.00
R_5	120	1499.66	175.00
R_6	120	1766.30	466.17

Now, suppose R_1 and R_2 are selected from each loop and their $TMS^{initial} = 0.06$. Then, the procedure can be described as:

1. **Select the clockwise loop and start from R_1**

   - ***Iteration No.1:***

     For a fault at point A "F_A", R_1 should operate as a primary DOCR and its operating time T_1 can be calculated from Eq.(3.3) as:

$$T_{primary,1} = \frac{\beta \times TMS_1}{\left(\frac{I_{F_A}}{PS_1}\right)^{\alpha} - 1} = \frac{0.14\ (0.06)}{\left(\frac{1978.9}{180}\right)^{0.02} - 1} = 0.171s$$

     If R_1 failed to operate, R_5 will operate after CTI as:

$$T_{backup,5} = T_{primary,1} + CTI = 0.171 + 0.3 = 0.471s$$

     So, using this value in Eq.(3.3) yeilds:

$$0.471s = \frac{0.14 \times TMS_5}{\left(\frac{175}{120}\right)^{0.02} - 1} \Rightarrow TMS_5 = 0.0255$$

Now, for a fault at point C "F_C", R_5 should operate as a primary relay with operating time:

$$T_{primary,5} = \frac{0.14\ (0.0255)}{\left(\frac{1499.66}{120}\right)^{0.02} - 1} = 0.0689s$$

If R_5 failed to operate, R_3 will operate after CTI as:

$$T_{backup,3} = T_{primary,5} + CTI = 0.0689 + 0.3 = 0.369s$$

By using the value of $T_{backup,3}$, TMS_3 can be calculated as:

$$0.369s = \frac{0.14 \times TMS_3}{\left(\frac{384}{100}\right)^{0.02} - 1} \Rightarrow TMS_3 = 0.0719$$

Thus, for a fault at point B "F_B", R_3 should operate as a primary relay with operating time:

$$T_{primary,3} = \frac{0.14\ (0.0719)}{\left(\frac{1683.9}{100}\right)^{0.02} - 1} = 0.173s$$

Now, R_1 will act as a backup relay when R_3 fails to operate:

$$T_{backup,1} = T_{primary,3} + CTI = 0.173 + 0.3 = 0.473s$$

The new value of TMS_1 is calculated as:

$$0.473s = \frac{0.14 \times TMS_1}{\left(\frac{617.22}{180}\right)^{0.02} - 1} \Rightarrow TMS_1 = 0.0843$$

- ***Iteration No.2:***

At F_A:

$$T_{primary,1} = \frac{0.14\ (0.0843)}{\left(\frac{1978.90}{180}\right)^{0.02} - 1} = 0.240s$$

If R_1 failed to operate:

$$T_{backup,5} = T_{primary,1} + CTI = 0.240 + 0.3 = 0.540s$$

By using Eq.(3.3):

$$0.540s = \frac{0.14 \times TMS_5}{\left(\frac{175}{120}\right)^{0.02} - 1} \Rightarrow TMS_5 = 0.0292$$

At F_C:

$$T_{primary,5} = \frac{0.14\ (0.0292)}{\left(\frac{1499.66}{120}\right)^{0.02} - 1} = 0.0790s$$

If R_5 failed to operate:

$$T_{backup,3} = T_{primary,5} + CTI = 0.0790 + 0.3 = 0.379s$$

By using Eq.(3.3):

$$0.379s = \frac{0.14 \times TMS_3}{\left(\frac{384}{100}\right)^{0.02} - 1} \Rightarrow TMS_3 = 0.0738$$

At F_B:

$$T_{primary,3} = \frac{0.14\ (0.0738)}{\left(\frac{1683.9}{100}\right)^{0.02} - 1} = 0.178s$$

If R_3 failed to operate:

$$T_{backup,1} = T_{primary,3} + CTI = 0.178 + 0.3 = 0.478s$$

By using Eq.(3.3):

$$0.478s = \frac{0.14 \times TMS_1}{\left(\frac{384}{100}\right)^{0.02} - 1} \Rightarrow TMS_1 = 0.0852$$

- ***Iteration No.3:***

At F_A:

$$T_{primary,1} = \frac{0.14\ (0.0852)}{\left(\frac{1978.90}{180}\right)^{0.02} - 1} = 0.243s$$

If R_1 failed to operate:

$$T_{backup,5} = T_{primary,1} + CTI = 0.243 + 0.3 = 0.543s$$

By using Eq.(3.3):

$$0.543s = \frac{0.14 \times TMS_5}{\left(\frac{175}{120}\right)^{0.02} - 1} \Rightarrow TMS_5 = 0.0294$$

At F_C:

$$T_{primary,5} = \frac{0.14\ (0.0294)}{\left(\frac{1499.66}{120}\right)^{0.02} - 1} = 0.0794s$$

If R_5 failed to operate:

$$T_{backup,3} = T_{primary,5} + CTI = 0.0794 + 0.3 = 0.379s$$

By using Eq.(3.3):

$$0.379s = \frac{0.14 \times TMS_3}{\left(\frac{384}{100}\right)^{0.02} - 1} \Rightarrow TMS_3 = 0.0739$$

At F_B:

$$T_{primary,3} = \frac{0.14\ (0.0739)}{\left(\frac{1683.9}{100}\right)^{0.02} - 1} = 0.178s$$

If R_3 failed to operate:

$$T_{backup,1} = T_{primary,3} + CTI = 0.178 + 0.3 = 0.478s$$

By using Eq.(3.3):

$$0.478s = \frac{0.14 \times TMS_1}{\left(\frac{384}{100}\right)^{0.02} - 1} \Rightarrow TMS_1 = 0.0852$$

It can be clearly seen that the value of TMS_1 did not change from the last iteration, so it can be stopped here.

2. **Select the anti-clockwise loop and start from** R_2

By repeating the preceding steps, the anti-clockwise loop can be coordinated as well. With the starting point of $TMS_2 = 0.06$, the results are as follows:

- ***Iteration No.1:***

  $\Rightarrow T_{primary,2} = 0.150s \Rightarrow T_{backup,4} = 0.450s \Rightarrow TMS_4 = 0.0840$

  $\Rightarrow T_{primary,4} = 0.230s \Rightarrow T_{backup,6} = 0.530s \Rightarrow TMS_6 = 0.1040$

  $\Rightarrow T_{primary,6} = 0.264s \Rightarrow T_{backup,2} = 0.564s \Rightarrow TMS_2 = 0.0303$

- ***Iteration No.2:***

  $\Rightarrow T_{primary,2} = 0.0756s \Rightarrow T_{backup,4} = 0.376s \Rightarrow TMS_4 = 0.0701$

  $\Rightarrow T_{primary,4} = 0.192s \Rightarrow T_{backup,6} = 0.492s \Rightarrow TMS_6 = 0.0967$

  $\Rightarrow T_{primary,6} = 0.245s \Rightarrow T_{backup,2} = 0.545s \Rightarrow TMS_2 = 0.0292$

- ***Iteration No.3:***

  $\Rightarrow T_{primary,2} = 0.0730s \Rightarrow T_{backup,4} = 0.373s \Rightarrow TMS_4 = 0.0696$

  $\Rightarrow T_{primary,4} = 0.191s \Rightarrow T_{backup,6} = 0.491s \Rightarrow TMS_6 = 0.0964$

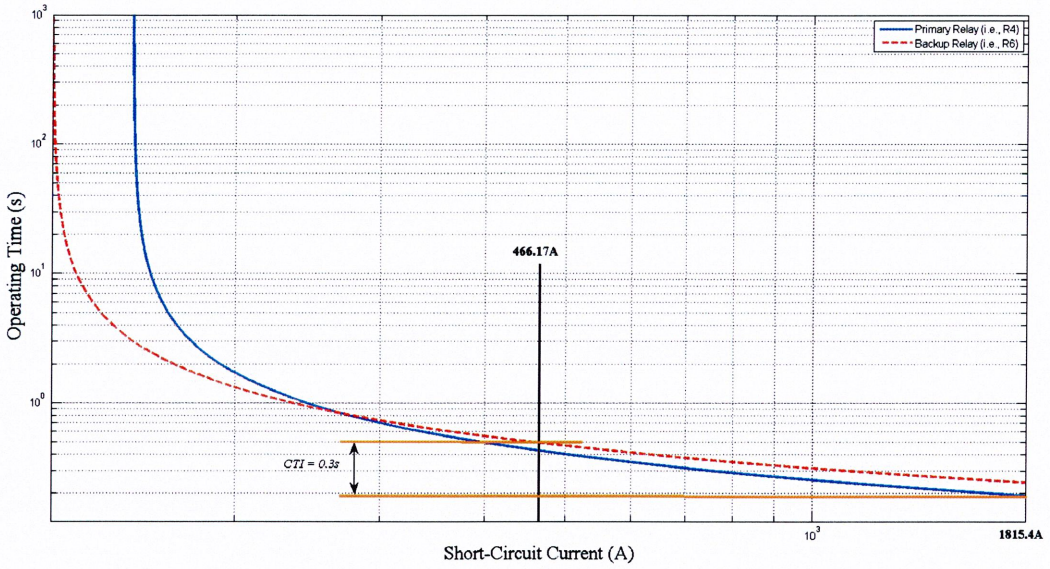


Figure (3.17)
TCCCs of R4/R6 relay pair with respect to I_{relay} on x-axis

$\Rightarrow T_{primary,6} = 0.244s \Rightarrow T_{backup,2} = 0.544s \Rightarrow TMS_2 = 0.0292$

Because TMS_2 did not change during the third iteration, so it can be stopped here.

Figure (3.17) shows the TCCCs of one selected Primary/Backup relay pair (R_4/R_6). The intersection of the two curves does not affect the selectivity among these two relays, because the current before this point is less than current seen by R_4. If the fault currents on the x-axis of the previous figure are divided by PS of each relay, then the characteristic curves can be represented with respect to PSM, as shown in Fig.(3.18).

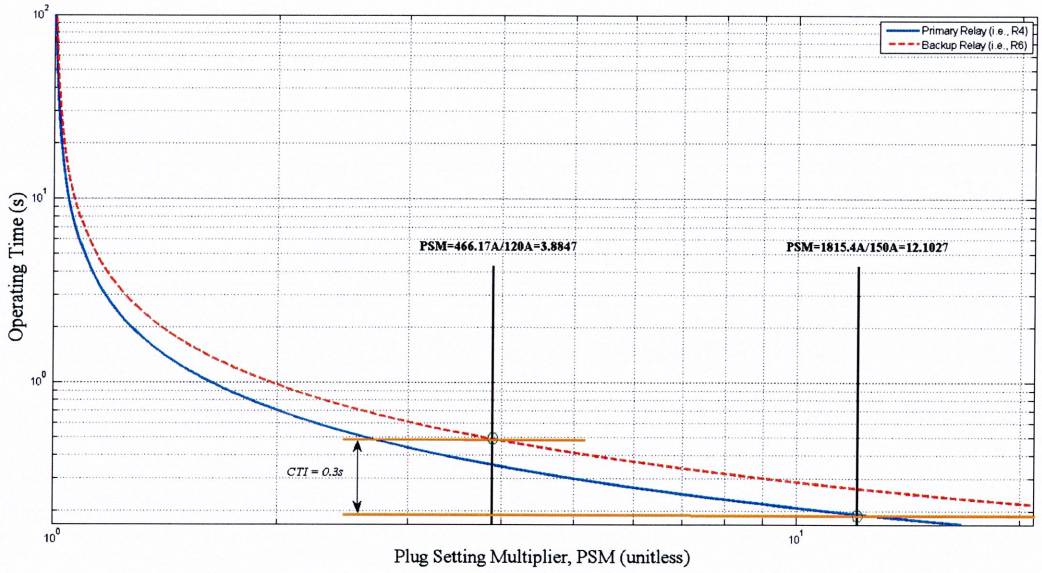


Figure (3.18)
TCCCs of R4/R6 relay pair with respect to PSM on x-axis

3.9.3 Conclusion of Case Study

From this case study, it has been shown that the coordination of DOCRs is an iterative problem; except for some special cases of radial systems, where all the settings of the relays are done within just one pass of calculations.

This classical iterative method is easy to solve, because the network is not complicated and many of assumptions have been considered to simplify the problem. The complexity of the coordination will increase if both TMS and PS are considered as variables in Eq.(3.3), where TMS is available in continuous form, PS is available within a set of discretized values, and the minimum and maximum allowable limits of TMS, PS and T are restricted. Moreover, this method cannot be applied for multi-loop networks in which some of the relays are shared between these loops. As a result, the concept of **minimum break point set (BPS)**

has been suggested to solve this stiff issue using the classical method which has been covered in the literature review in Chapter 1.

3.10 General Optimal Coordination of DOCRs

From the literature review in Chapter 1, the historical timeline of the coordination problem of directional overcurrent relays (DOCRs) has been presented. Also, some of the basic fundamentals have been covered in the previous Sections of this Chapter, which are very essential for those who want to understand the optimal coordination of DOCRs.

The optimal coordination of DOCRs problem can be solved numerically, and the optimal (*or at least near optimal*) solution could be obtained using optimization techniques.

The content of this Section will focus on how to formulate the coordination problem of DOCRs into a mathematical problem that can be solved later by any available n-dimensional optimization technique.

General Program Requirements:

To build a program as a numerical solver to optimally coordinate those relays between each other, then all the stages in the flowchart of Fig.(3.19) must be satisfied first. The following Sub-Sections will describe each stage of this flowchart.

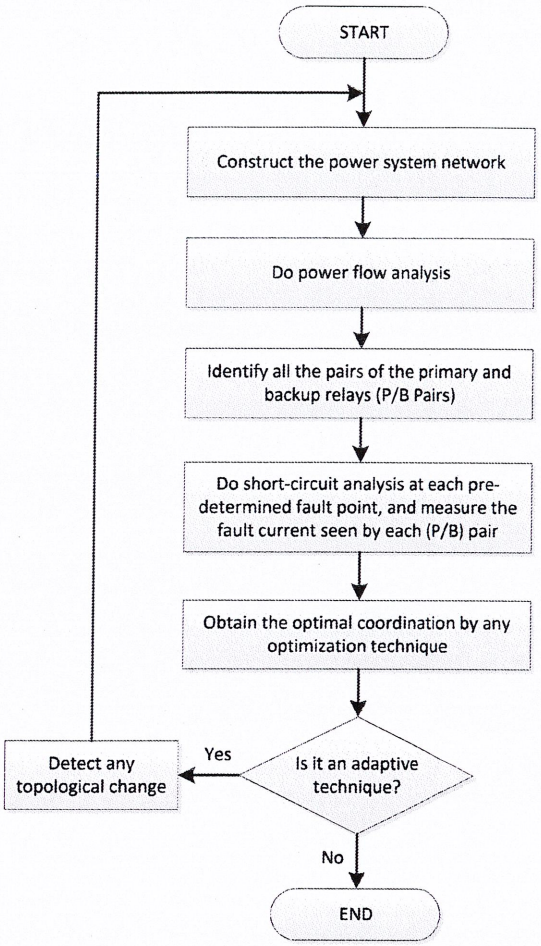


Figure (3.19)
Flowchart of the general optimal coordination of DOCRs

3.10.1 Constructing the Power System Network

This stage will provide all the required information about the given network; like its type (*radial network, double-line network, ring network, etc*), number of buses, number of lines with their impedances, number of generators and loads with their rated values, types of the protective devices and their TCCCs, types of faults and their locations, network's topology (*fixed or dynamic*), status of CBs, etc.

This can be considered as a first step of any proposed problem, be-

cause without satisfying this stage then there is no other choice to go ahead for the next stages. For example, if ETAP (*commercial software*) is used with some missed information of this stage, then it will show an error message when it is run.

3.10.2 Power Flow Analysis

The studies of power flow (load flow) are considered as a backbone for any analysis or design on the power system. If the given network is formulated in a nodal admittance form, then by the node currents the network can be expressed in linear algebraic equations. But practically, the systems are represented with power values instead of currents, which result in a set of non-linear algebraic equations (also known as **power flow equations**), and must be solved iteratively (Saadat, 1999; Husain, 2007).

For smooth calculations, the system's data is available in per-unit format and the system itself is assumed to be operated under normal balanced steady-state condition, so the three-phase system can be solved via a single-phase representation with a positive sequence network (Saadat, 1999; Glover et al., 2010). Power flow analysis is carried out in order to make sure that the following requirements are satisfied (Glover et al., 2010):

1. The magnitudes of the bus voltage are near to the rated values.
2. The power generation should meet the total consumed power (i.e., the summation of the demand power and losses).

Table (3.5)
Bus types and their known/unknwon variables for power flow analysis

Bus Type	Known	Unknown
Swing bus (slack or buck reference bus)	$\|V_i\|$ and δ_i	P_i and Q_i
Generator bus (voltage-controlled or PV bus)	P_i and $\|V_i\|$	Q_i and δ_i
Load bus (PQ bus)	P_i and Q_i	$\|V_i\|$ and δ_i,

3. Generators should not exceed the specified limits of the real and reactive power.
4. There is no overloading on the transformers or the transmission lines.

Thus, the designed power flow program should calculate the voltage magnitude "$|V_i|$", the angle "δ_i", the real power "P_i" and the reactive power "Q_i" for bus or node i (Glover et al., 2010; Husain, 2007). In power flow analysis, each bus has two known (specified) variables and two unknown variables. The types of buses depends upon the known variables, and can be summarized in Table 3.5.

Power flow analysis as a subject is very big and it is one of major courses for undergraduate engineers. Because of the space limitation, the reader can refer to (Saadat, 1999; Glover et al., 2010; Husain, 2007; Kothari & Dhillon, 2011) for detailed description.

3.10.3 Primary/Backup Pairs Identification

As has been covered in the previous sections of this Chapter, coordinating the relays means that the selectivity among groups of relays is done correctly and sequentially. So, if a fault happens in any zone, then the corresponding primary relay (R_{pr}) should operate first, and if it fails

or exceeds the assigned chance (i.e., CTI), one or group of backup relays (R_{bc}) should operate immediately. Also, each one of these backup relays will work as a primary relay for another group of relays. Thus, finding primary/backup pairs (P/B pairs) of all DOCRs is a very important step for solving the coordination problem.

There are many ways to determine these P/B relay pairs, and the main three methods are listed below:

3.10.3.1 Determining P/B Relay Pairs By Inspection

It is the most popular method, which is preferable for small and non-complicated networks. It is also recommended if the optimal coordination is done based on one fixed topology. The basic idea behind this method is to trace the current direction that DOCR can see, and then try to find the other DOCRs in other zones that can see the same current direction.

Figure (3.20) illustrates a simple 5-bus network. By applying the preceding method, if R_1 is considered as R_{pr}, then R_6 will work as R_{bc} after waiting CTI; this can be seen by tracing the arrows near the relays in Fig.(3.20). Also, if R_2 is considered as R_{pr}, then the group of $\{R_1, R_4, R_{11}\}$ will work as R_{bc}. After completing this process for all 12 DOCRs, then Table 3.6 will be constructed.

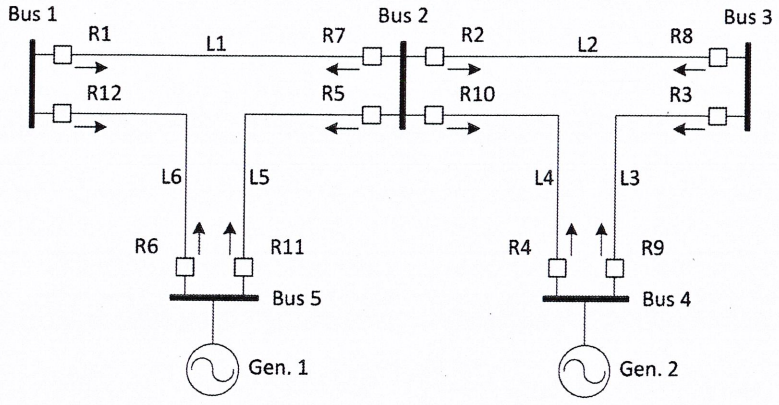


Figure (3.20)
Determining P/B relay pairs of a simple 5-bus system

Table (3.6)
P/B relay pairs of a simple 5-bus system

R_{pr}	1	2	3	4	5	6	7	8	9	10	11	12
R_{bc}	6	1,4,11	2	3	1,4,8	5	4,8,11	9	10	1,8,11	12	7

3.10.3.2 Determining P/B Relay Pairs Using Graph Theory

This approach can be briefly described as (Elrafie & Irving, 1993):

1. **Numbering all the relays:**

   - Because each line of the network has two DOCRs on its both ends, so the total DOCRs are equal to $2L$, where L is the total number of lines.
   - Make two groups of DOCRs with length equal to L. The first group will contain $\{R_1, R_2, \cdots, R_j, \cdots, R_L\}$, and the second group will contain $\{R_{L+1}, R_{L+2}, \cdots, R_{L+j}, \cdots, R_{2L}\}$; where $j = 1, 2, \cdots, L$.

- From each line, the relay of the first selected end will be numbered by j, and the relay of the other end will be numbered by $(L + j)$. Continue this process till reaching the last line, where R_j becomes R_L and R_{L+j} becomes R_{2L}.

2. **Constructing the nodal and augmented incidence matrices:**

- Construct the nodal incidence matrix A where the number of rows is equal to the number of buses B, and the number of columns is equal to the number of lines L:

$$A = [B \times L] = \begin{bmatrix} e_{1,1} & e_{1,2} & \cdots & e_{1,j} & \cdots & e_{1,L} \\ e_{2,1} & e_{2,2} & \cdots & e_{2,j} & \cdots & e_{2,L} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ e_{i,1} & e_{i,2} & \cdots & e_{i,j} & \cdots & e_{i,L} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ e_{B,1} & e_{B,2} & \cdots & e_{B,j} & \cdots & e_{B,L} \end{bmatrix} \quad (3.5)$$

- The elements $e_{i,j}$ of this matrix can be determined as:
  - For each ith bus, see the end of each line that is connected to the ith bus.
  - If that end contains R_j, then give it "+1".
  - Otherwise, if it contains R_{L+j}, then see the second end and give it "-1" on the same row of the ith bus.
- Once completing the nodal incidence matrix A, the augmented incidence matrix A_d can be constructed by duplicating A and

multiplying it by "-ve" sign:

$$A_d = [A, -A] \tag{3.6}$$

3. **Identifying P/B pairs:**

- From the augmented incidence matrix "A_d", the positive element of each column will represent $R_{bc,j'}$; where $j' = 1, 2, \cdots, j, j+1, \cdots, L, L+1, \cdots, L+j, \cdots, 2L$.
- If $R_{bc,j'}$ is selected from j'th column, then see the negative element from the same column and save the new ith row.
- From that ith row, select all the positive elements as primary relays of that backup relay $R_{bc,j'}$; except one positive element which needs to be excluded.
- If $R_{bc,j'}$ is located in the first group (i.e., $1 \leqslant j' \leqslant L$), then exclude the positive element in the location of $(j' + L)$, and in the location of $(j' - L)$ if $R_{bc,j'}$ is located in the second group (i.e., $L + 1 \leqslant j' \leqslant 2L$).

To clarify the preceding steps, let's take the previous example in Fig.(3.20). Note that these DOCRs are numbered in a way to fit with the method of graph theory itself. There are 6 lines, which give 12 DOCRs, and thus $j' = 1, 2, \cdots, 12$. If the selected line has $R_{j'}$ with $j' = j$ (i.e., is located between 1 to 6), then the other relay of the second end of that line will be numbered by R_{j+6}. From the network, the nodal and augmented

incidence matrices can be respectively determined as:

$$A = \begin{bmatrix} e_{1,1} & e_{1,2} & \cdots & e_{1,j} & \cdots & e_{1,L} \\ e_{2,1} & e_{2,2} & \cdots & e_{2,j} & \cdots & e_{2,L} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ e_{i,1} & e_{i,2} & \cdots & e_{i,j} & \cdots & e_{i,L} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ e_{B,1} & e_{B,2} & \cdots & e_{B,j} & \cdots & e_{B,L} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & -1 \\ -1 & 1 & 0 & -1 & 1 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}$$

$$A_d = \left[\begin{array}{cccccc|cccccc} 1 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 1 \\ -1 & 1 & 0 & -1 & 1 & 0 & 1 & -1 & 0 & 1 & -1 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 1 & -1 \end{array}\right]$$

From A_d, if column no.1 is selected, then $e_{1,1}$ indicates that the relay R_1 works as a backup relay for some primary relays. To find these relays, let's select the negative element of that column, which is located in the second row (i.e., $e_{2,1}$). From row no.2, the positive elements $\{e_{2,2}, e_{2,5}, e_{2,10}\}$ respectively indicates that the relays $\{R_2, R_5, R_{10}\}$ work as primary relays. Because the backup relay R_1 is listed in the first group, then the positive element $e_{2,1+L}$ (which is $e_{2,7}$) is excluded. Therefore, R_1 has to backup R_2, R_5 and R_{10}.

To see how to find the P/B relay pairs for those backup relays that are listed in the second group (i.e., $R_{bc,7}$ to $R_{bc,12}$), let's select column

no.8, the positive element $e_{3,8}$ means that R_8 works as a backup relay for some primary relays. $e_{2,8}$ is the negative element of that column, and it is located in the second row. Because R_8 is listed in the second group, so the positive element that has to be rejected is ($e_{2,8-L} = e_{2,2}$). The remaining positive elements $\{e_{2,5}, e_{2,7}, e_{2,10}\}$ will respectively represent the primary relays $\{R_5, R_7, R_{10}\}$. Thus, R_8 is considered as R_{bc} for R_5, R_7 and R_{10}.

By repeating these steps for all the columns of A_d, Table 3.6 will be constructed again.

This approach maybe relatively more difficult than the previous one, but it gives a systematic procedure to identify these P/B relay pairs automatically without any human mistake as if it is solved classically by hand. Based on that, graph theory method provides some capabilities to the researchers who want to study the optimal coordination of dynamic networks (i.e., networks with many configurations "topologies"), where all the pairs will be re-identified if any element (bus or line) is added into or removed out from the network.

3.10.3.3 Determining P/B Relay Pairs Using Softwares

In addition, by using some commercial softwares, that are based on LINKNET structure (Singh, Panigrahi, & Abhyankar, 2013), these P/B relay pairs can be determined easily. It is recommended to use this method for very complicated and large systems. However, the main prob-

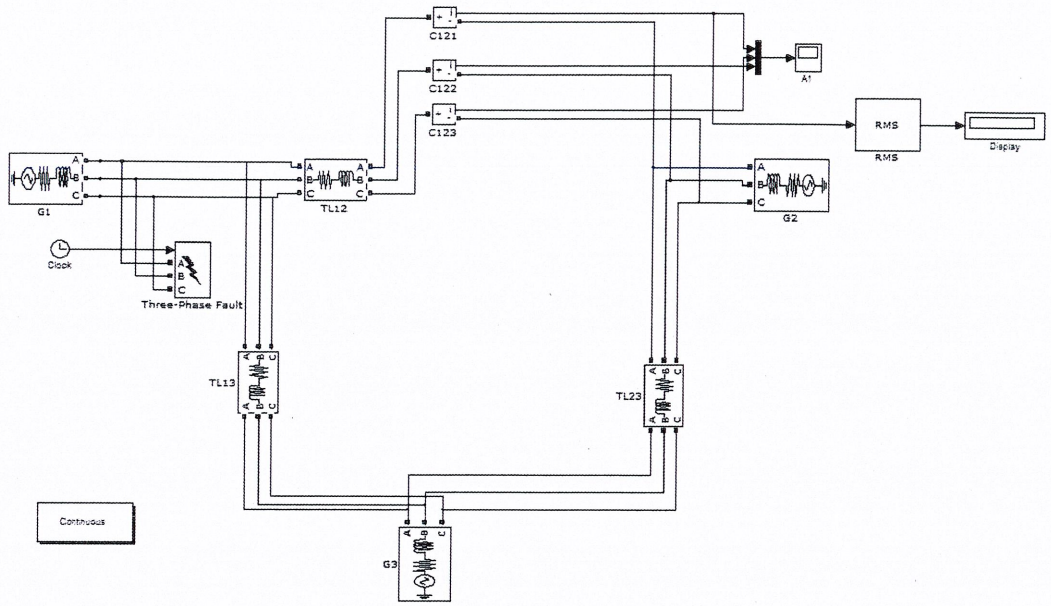


Figure (3.21)
Fault analysis for the 3-bus system using MATLAB Simulink

lem with these softwares is the hardness to make an interface with other related softwares (like MATLAB or Python), because the main objective is to obtain the optimal coordination, which can be done by MATLAB or other softwares that have the ability to solve numerical problems.

3.10.4 Short-Circuit Analysis

From Section 3.1, some basic principles have been covered. It has been shown that the severity of short-circuit current depends on the type of fault and its location. The good protective design should detect the highest overloads as well as the lowest severe faults, which are reflected on the allowable limits of PS. Thus, in order to carry out the process of optimal coordination, it is important to do first the fault analysis of the given network. 3ϕ fault analysis is used, because if the coordination is satisfied under this worst case, it will be satisfied for the other cases

(valid if only one TCCC is used for all DOCRs).

Two questions may arise:

- ***How to determine the minimum and maximum PS of DOCR based on the given maximum overload and minimum fault that can be seen by that relay?***
- ***How to select the location of faults on each line of the given network?***

It is better to answer these two questions in Sub-Section 3.10.5, because there are many aspects that need to be clarified.

The important thing of this Sub-Section is to say that when any one wants to solve the problem of optimal coordination of DOCRs, then the hand calculation of fault analysis is totally not recommended. Also, in the real life, doing short-circuit analysis will require many practical considerations.

Therefore, this weary calculation can be done quickly and smoothly within the previous commercial softwares as well as MATLAB Simulink, see Fig.(3.21). By this way, huge amount of effort and time can be saved, and can give the ability to easily re-analyze the system under different assumptions and scenarios. On the other hand, the classical calculation by hand may give some mistakes or errors in its tolerance and needs to be started again from the stage where the new assumptions or mistakes have not been considered.

3.10.5 Applying Optimization Techniques

From Chapter 2, it has been seen that for finding the optimal value of a given problem, it needs first to be translated into a mathematical expression. This Sub-Section will describe how to formulate the coordination problem, so it can be plugged in any proposed optimization technique.

Based on the TCCCs, OCRs can be categorized into three main types: instantaneous, definite time and inverse time. In this thesis only the inverse time DOCR relay is considered.

3.10.5.1 Objective Function

If an electrical network contains n_r relays, then the operating time of the ith relay for a fault at location k is computed as:

$$T_{ik} = f\left(TMS_i, PS_i, I_{R_i,k}\right) \quad , \quad i = 1, 2, \cdots, n_r \tag{3.7}$$

where $I_{R_i,k}$ is a fault current at location k that is seen by the primary relay R_i. T_{ik} can be modeled using the standard IEC or ANSI/IEEE characteristics (IEC, 1989; IEEE, 1999). By substituting Eq.(3.3) in Eq.(3.7), the following general expression is obtained (Gers & Holmes, 2004):

$$T_{ik} = TMS_i \times \left[\frac{\beta}{\left(\frac{I_{R_i,k}}{PS_i}\right)^{\alpha} - 1} + \gamma\right] \tag{3.8}$$

For the sake of presenting a fair comparison with the performance of the other methods presented in the literature, the IEC inverse definite minimum time (IDMT) characteristic is selected in this thesis, where the constants α, β and γ are set equal to 0.02, 0.14 and 0, respectively.

The objective is to find the values of PS_i and TMS_i of each DOCR (at predetermined I_R of all n_r relays and l fault locations from fault analysis), so the weighted sum of the operating times, when relays work as primary, is minimized as:

$$min\ Z = \sum_{i=1}^{n_r} \sum_{k=1}^{l} w_{ik} T_{pr,ik} \tag{3.9}$$

where w_{ik} represents the probability of a given fault that may occur in each location of the protective zones. All the weights in Eq.(3.9) are considered equal to one (Urdaneta et al., 1988).

Suppose there are l fault locations on each zone, Damborg et al. (Damborg, Ramaswami, Venkata, & Postforoosh, 1984) classified three levels of coordination criteria. The first one is called "**desired design criteria**" which considers two fault locations, "**close-in**" (also known as "**near-end**") and "**remote-bus**" (also known as "**far-end**" or "**tail-end**")[1]. That criteria could be relaxed to "**minimum design criteria**" where the relays are not allowing to operate for lesser fault currents. When larger classes of faults are studied, then one fault at the "**mid-**

[1]The near-end fault is the fault that happens at the nearest possible point of the line where the primary protective relay is installed, and vice versa for the far-end fault.

point" of each line could be considered and this case is known as "**enhanced design criteria**" (Birla, Maheshwari, & Gupta, 2006).

For the desired criteria, Eq.(3.9) is re-written as follows:

$$min\ Z = \sum_{p=1}^{m} T_{pr,p}^{near} + \sum_{q=1}^{n_r} T_{pr,q}^{far} \tag{3.10}$$

This expression will be applied later in Chapter 5 to solve 4-bus system. For other bus systems, where the minimum and enhanced criteria are considered, the expression will be reduced to:

$$min\ Z = \sum_{i=1}^{n_r} T_{pr,i} \tag{3.11}$$

Optimization of the objective function is subjected to the following constraints:

3.10.5.2 Coordination Criteria

If the primary relay fails to isolate the fault at location k, one or more backup relays in the upstream should initiate the trip signal after waiting, at least, the operating time of the primary relay plus the coordination time interval (i.e., CTI). This sequence will create discrimination margin between primary-backup (P/B) pairs which satisfies the selectivity criteria and prevents mal-operation by giving enough chance for the primary relay to isolate the in-zone fault before initiating the backup pro-

tection. Thus, the operating time T_{jk} of the backup relay R_j for a given fault at location k is subjected to the following constraint:

$$T_{jk} - T_{ik} \geqslant CTI \tag{3.12}$$

where T_{jk} is given as:

$$T_{jk} = TMS_j \times \left[\frac{\beta}{\left(\frac{I_{R_j,k}}{PS_j}\right)^{\alpha} - 1} + \gamma \right] \tag{3.13}$$

where $I_{R_j,k}$ is a fault current at location k that is seen by the backup relay R_j. The value of CTI varies between 0.2 and 0.5 s depending on the relay type, CB operating time, and security margin (Gers & Holmes, 2004; Blackburn & Domin, 2006; Anthony, 1995), see Eq.(3.4). The transient topologies occur when only one relay of a protected zone has operated. The word "transient" does not mean a transient current, it means that the network topology will be changed temporarily due to the operating delay of the other relay (Urdaneta et al., 1988). If these conditions are considered, then the optimal coordination problem should also satisfy the transient constraint, which can be formulated as:

$$T'_{jk} - T'_{ik} \geqslant CTI \tag{3.14}$$

where T'_{jk} and T'_{ik} are the operating time of the P/B pair under transient condition.

3.10.5.3 Limits on the Relay Operating Time

The operating time of relays is practically bounded by two limits, which can be expressed as:

$$T_{ik}^{min} \leqslant T_{ik} \leqslant T_{ik}^{max} \tag{3.15}$$

where T_{ik}^{min} and T_{ik}^{max} are the minimum and maximum operating times of R_i for a fault at location k, respectively. T_{ik}^{min} depends on the relay manufacture, whereas T_{ik}^{max} depends on the critical clearing time required to prevent equipment damage and preserve the system stability.

3.10.5.4 Limits on the Time Multiplier Setting

The TMS is bounded by the available range of setting provided by the relay manufacturer, which can be expressed as:

$$TMS_i^{min} \leqslant TMS_i \leqslant TMS_i^{max} \tag{3.16}$$

where TMS_i^{min} and TMS_i^{max} are the minimum and maximum values of TMS_i of the relay R_i.

3.10.5.5 Limits on the Plug Setting

Practically, the PS of the relay should be set higher than the circuit nominal current and lower than minimum fault current, which can be

expressed as:

$$PS_i^{min} \leqslant PS_i \leqslant PS_i^{max} \tag{3.17}$$

These two bounds can be calculated as:

$$PS_i^{min} = \frac{OLF \times I_{n,i}}{CTR} \tag{3.18}$$

$$PS_i^{max} = \frac{2}{3CTR} I_{f,i}^{min} \tag{3.19}$$

where $I_{n,i}$ is the nominal current rating of the circuit protected by the relay R_i. OLF is the overload factor, CTR is the CT ratio, and $I_{f,i}^{min}$ is the minimum fault current that should be detected by the relay R_i (Gers & Holmes, 2004).

To be more specific, Eqs.(3.18)-(3.19) are considered in solving the 9-bus system in Chapter 5 with $OLF = 1.25$, while the other bus systems take PS_i^{min} and PS_i^{max} as constants for all the relays.

3.10.5.6 Assumptions

It is important to note that this thesis takes into account the same assumptions that have been considered in the other studies reported in the literature in order to have a fair comparison between the performance of the proposed algorithm in Chapter 5 and the other reported algorithms. The assumptions are as follows:

- Networks' signals are not affected by decaying DC component,

harmonic and/or CT saturation.

- Thermal limits of transmission/sub-transmission lines are not considered.
- Networks' topologies are fixed, except for the 3-bus system.
- Time-Current Characteristic is limited to IDMT curve (except some scenarios).

3.10.6 Adaptive Optimal Coordination

If the study focuses only on one topology of the system, then the program can be terminated once the last stage of the flowchart in Fig.(3.19) is reached, and the coordination problem is considered to be optimized.

From practical side, there is no system that can operate on a steady-state condition, because the status of the available elements of the generation, transmission and utilization sides can be changed at any time. Therefore, if the researchers are interest to have an adaptive optimizer for solving the issue of dynamic configurations of the networks, then the unterminated loop will repeat all the previous stages based on the new status of the network.

These loops can be initiated continuously, or within a fixed interval to save the memory of the hardware for other applications embedded in the **energy management system** (**EMS**) software.

Chapter 4

Biogeography-Based Optimization

4.1 Literature Review

4.2 The Theory of Island Biogeography

4.3 Biogeography-Based Optimization (BBO)

4.4 Biogeography Based Optimization with Essential Modifications (BBO-EM)

4.5 Iterative Variable Bounds – Biogeography Based Optimization (IVB-BBO)

4.6 Adaptive Variable Bounds - Biogeography Based Optimization (AVB-BBO)

Chapter 4
Biogeography-Based Optimization

The science of biology becomes one of the main resources of inspiration to develop the modern optimization techniques, such as ant colony optimization (ACO), bee colony optimization (BCO), wasp swarm optimization (WSO), bacterial foraging optimization (BFO), genetic algorithm (GA), evolutionary strategy (ES), differential evolution (DE), particle swarm optimization (PSO), etc. Biogeography-based optimization (BBO) is a new population-based evolutionary algorithm (EA) and one of meta-heuristic algorithms. This technique is based on an old mathematical study (called **The Theory of Island Biogeography** (MacArthur & Wilson, 1967)) that explains the geographical distribution of biological organisms.

4.1 Literature Review

The first original form of BBO was introduced by Daniel Simon in Dec. 2008 and known as a **partial migration based BBO** (Simon, 2008a), where its performance was evaluated based on 14 benchmark functions, and then was tested to solve a real sensor selection problem for aircraft engine health estimation. BBO did well and proved that it is a very competitive method as compared to the other EAs. After three months, BBO was re-introduced again with additional three other forms and known as **single**, **simplified partial**, and **simplified single migra-**

tion based BBOs (Simon, 2011). Then, many of researches have been conducted to solve some practical problems such as economic emission load dispatch (Bhattacharya & Chattopadhyay, 2009), land cover feature extraction (Panchal, Goel, & Bhatnagar, 2009), unit commitment (Mohammed & Talaq, 2012), etc. The researchers found that the original versions of BBO showed some drawbacks; they have a good exploitation for global optimization, but they lack exploration (Pattnaik, Lohokare, & Devi, 2010; Gong, Cai, & Ling, 2010; Gong, Cai, Ling, & Li, 2010; Lohokare, Pattnaik, Panigrahi, & Das, 2013). Thus, additional efforts were spent in order to enhance their overall performance. Some researchers modified the BBO algorithm itself (Ma, Ni, & Sun, 2009; Ma, 2010; Ma & Simon, 2011a; Simon, Ergezer, & Du, 2009; Simon, 2011; Pattnaik et al., 2010), while the others proposed hybrid versions between BBO and other evolutionary algorithms (Gong, Cai, & Ling, 2010; Ergezer, Simon, & Du, 2009; Ma & Simon, 2011b). Although these approaches improved the BBO performance, they share a common problem that the overall BBO algorithm become more complicated and hard to be implemented practically.

In (Ergezer et al., 2009), the **Opposition-Based Learning (OBL)** is employed in BBO to improve its migration process, and thus the new created algorithm is presented as **Oppositional BBO (OBBO)**. The OBL compare the fitness of a solution estimate to its opposite, where the better one is used in the population. Thus, the convergence speed can be accelerated.

Other researches focused on the equilibrium species counts $\hat{S}$, because this location contains the highest probability P_{max} and the lowest mutation, as can be seen from Fig.(4.6). A variety of shapes are used for the immigration and emigration rates, as described by naturalists (MacArthur & Wilson, 1967; Lomolino, Riddle, & Brown, 2009; Cody, 2006; MacArthur & Connell, 1966; MacArthur, 1972). In (Ma et al., 2009), the constant and/or linear rates are used for λ_s and μ_s with a bias θ as a minimal emigration rate. This research was expanded more in (Ma, 2010), and employed other functions like trapezoidal, quadratic and sinusoidal, and found that the sinusoidal migration model achieved the best overall performance. Moreover, based on the results that are obtained from (Ma, 2010), and by employing the advantages gained from the past studies on the effects of successional changes on the immigration rate by James A. MacMahon in 1987 (McMahon, 1987) (*for more details, see ch.3 of* (Lomolino et al., 2009)), the performance of BBO can be improved significantly if its migration rates are designed with phase shifted sinusoidal function. The obtained model is named as **Generalized Sinusoidal (GenSin) Migration Model** (Ma & Simon, 2011a). This work was analyzed again in (Guo, Wang, & Wu, 2014) and proposed two other new models.

A **Real-Coded BBO (RCBBO)** is introduced in (Gong, Cai, Ling, & Li, 2010), and studied the Gaussian, Cauchy and Levy mutation operators. The algorithm is encoded by the floating point rather than integer. This modification improved the population diversity and enhanced the

algorithm exploration.

In (Pattnaik et al., 2010), a new modified algorithm called **Enhanced BBO (EBBO)** is presented. The idea behind it is that when the migration process is completed, some of the modified islands become similar. As a result, the diversity of the new generated solutions reduces. To prevent this situation, a new clear duplicated operator with random mutation is introduced.

The **Blended BBO (BBBO)** is another approach that has the ability to solve the preceding problem (Ma & Simon, 2011b). BBBO is equipped with blended crossover operator of the GA, where the emigrated features will not fully take place of the old features of the poor island. Instead, the new emigrating feature is a mixture of the poor island feature and the good island feature with a percentage ratio α, as described by Eq.(4.1):

$$ISI_i(SIV) \leftarrow \alpha ISI_i(SIV) + (1-\alpha) ISI_j(SIV) \qquad (4.1)$$

where the fraction α is a real number between 0 and 1. It could be constant, random or even proportional to the relative fitness of the solutions ISI_i and ISI_j. Based on that, the good island still can save some of its good features if it is probabilistically modified by less good island. When $\alpha = 0$, the blending part can be eliminated and the BBO algorithm will return back to its original form. This work was extended more in (Guo, Chen, Wang, Ge, & Wu, 2013), by using five different operators that are

motivated by the variation of crossover in GA.

In (Johal, Singh, & Kundra, 2010), a hybridization of **Flower Pollination by Artificial Bees (FPAB)** with BBO algorithm is constructed to classify a satellite image. The original BBO algorithm does not inherently have the clustering property. FPAB is used to find the clusters and these clusters are further classified using BBO, which acts as an optimal solution finder.

Other hybridization techniques have been reported in (Kundra & Sood, 2010; Sood & Kaur, 2012). However, the most interesting approach is in (Gong, Cai, & Ling, 2010), where the good exploitation of BBO is combined with the good exploration of **Differential Evolution (DE)** to create a very competitive algorithm. Similar concept is used in (Lohokare et al., 2013), in which a modified DE algorithm is used instead to solve the slow convergence of the BBO algorithm.

In (Ma, Fei, Ding, & Jin, 2012), it was observed that the performance of BBO is sensitive to the shapes of the emigration rate μ_s and the immigration rate λ_s, and found that it is impossible to always get good performance by just using one migration model. Thus, BBO with an **Ensemble of Migration Models (BBO-EMM)** is presented using three parallel populations process.

In addition to the original migration types that are described at the beginning of this section, three other variations of BBO migration were introduced with their corresponding Markov chain models in (Ma, Si-

mon, Fei, & Xie, 2013). These variations are called **Total Immigration-Based BBO (TIB-BBO)**, **Partial Emigration-Based BBO (PEB-BBO)**, and **Total Emigration-Based BBO (TEB-BBO)**. For 3-bit unimodal problems, TEB-BBO and PEB-BBO perform the best, while PMB-BBO performs the best for 3-bit deceptive problems, and all of these BBO variations have similar results for 3-bit multimodal problems.

The objective of this Chapter is to outline a clear path and strong basis for selecting the best algorithm among these four original forms, and then try to solve the associated problems that cause weaknesses on the exploration. Based on that, BBO with correct form and corrected algorithm will be presented in a comprehensive comparison through wide range of benchmark functions. Thus, any existing modification with wrong selected BBO form can be reviewed again with this guidance and essential corrections so that its performance can be enhanced more than what it already has. In addition, this new proposed BBO algorithm can be used as a foundation or as a first step of modifications for any future model. At the end of this Chapter, some advanced BBO models will be proposed based on the above guidance and essential corrections.

4.2 The Theory of Island Biogeography

Biogeography is a branch of biology, and it is a synthetic discipline, relying heavily on theory and data from ecology, population biology, systematics, evolutionary biology, and the earth sciences (Lomolino et al., 2009). Biogeography seeks to describe, analyze and explain the

geographic patterns and changing distributions of ecosystems and fossil species of plants (flora) and animals (fauna) through geological space and time (Jones, 1980; Cox & Moore, 1993).

Island, in biogeography, is any area of suitable habitat (local environment occupied by an organism (Lincoln, Boxshall, & Clark, 1982)) surrounded by an expense of unsuitable habitat and is endowed with exceptionally rich reservoirs of endemic, exclusive, strange and relict species (Maiti & Maiti, 2011). Islands as ecological systems have such salient features as simple biota, varying combinations of biotic and abiotic factors, and variability in isolation, shape, and size. Thus, island could be aquatic island, desert oasis, individual plants, caves, lakes or ponds, mountain-tops (sky-islands), microcosms or even patches of terrestrial ecosystems (Nierenberg, 1995; Myers & Giller, 1990). With these characteristics, islands represent themselves as natural experiments, see Fig.(4.1), and got highly attentions by the nineteenth century naturalists of the first rank, such as Alfred R. Wallace in East Indies (Shermer, 2011), Charles Darwin in Galapagos Islands (Sulloway, 1982) and Joseph D. Hooker in Southern Ocean (Desmond, 2007).

Island biogeography is a special field within biogeography science. This field was initially started by the ecologists Robert H. MacArthur and Edward O. Wilson in 1960 to 1963 with their published paper (MacArthur & Wilson, 1963), and continued their studies till 1967 when the final achievement was presented (MacArthur & Wilson, 1967); and recently, this theory has been revisited and expanded (Losos & Ricklefs, 2010).

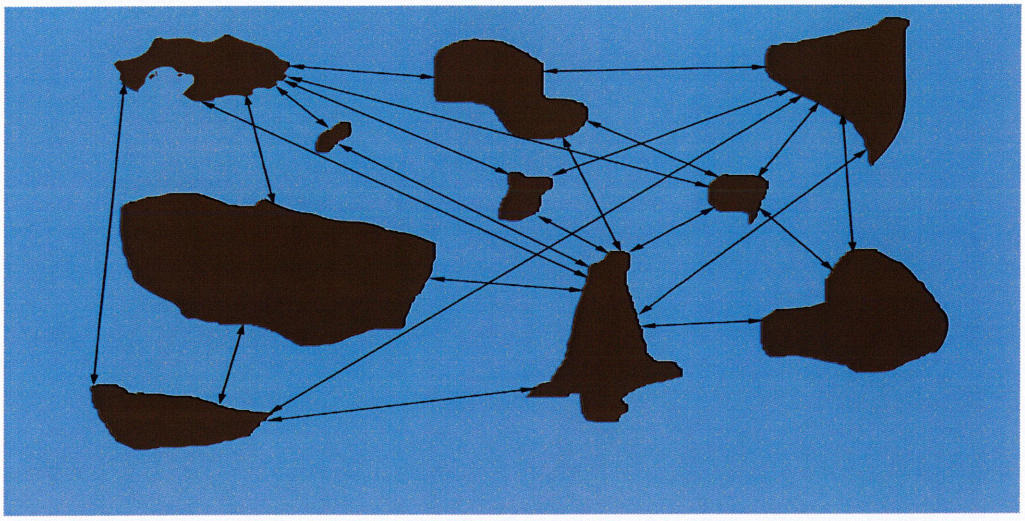


Figure (4.1)
Migration process between islands with different isolation, shape, and size

Island biogeography theory fully integrates much of ecology, population biology, evolution, and paleontology, with important implications for conservation of species (Losos & Ricklefs, 2010). It was developed with mathematical models for attempting to translate the ecology and biogeography from the traditional view to analytical view, and answering why some islands are rich of species while the others are poor, by establishing and explaining the biotic (*like predation, competition and interactions between species*) and abiotic (*like wind, water, sunlight, temperature, pressure and soil*) factors that affect the species richness of natural communities in an island (MacDonald, 2003). Thus, it gives the ability to predict the species counts that migrate between islands and then can find the optimum conservation areas (Lomolino et al., 2009; Jones, 1980; Cox & Moore, 1993; Maiti & Maiti, 2011). The equilibrium theory of island biogeography proposes that the number of inhabited species on an island is based on the dynamic equilibrium between new immigrated species onto an island and the extinct species out from that island (MacArthur &

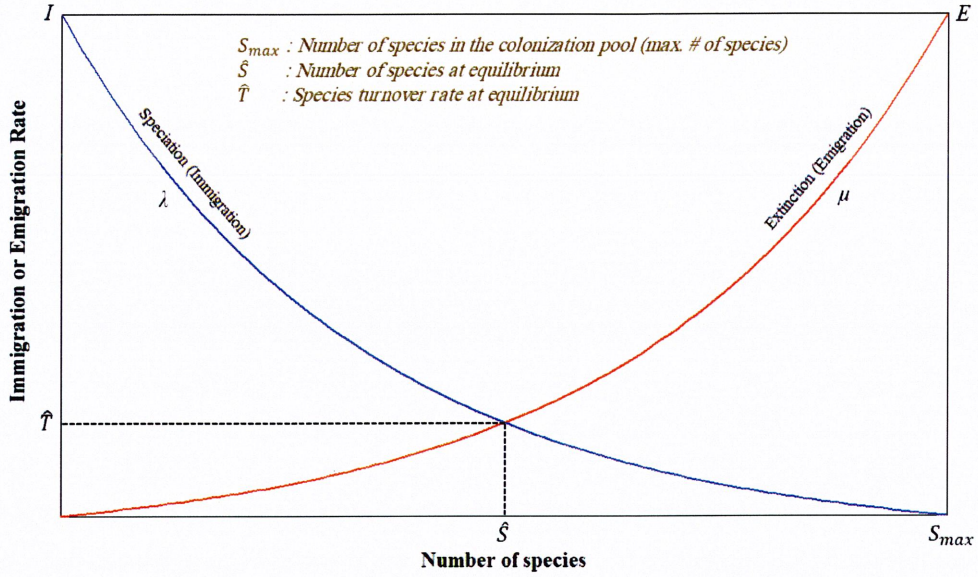


Figure (4.2)
Equilibrium model of a biota of a single island

Wilson, 1963, 1967; Losos & Ricklefs, 2010).

Figure (4.2) graphically represents the equilibrium model with exponential immigration (or speciation) rate λ and emigration (or extinction) rate μ, where they can also be plotted as a logistic, linear or any proper function (Lomolino et al., 2009; Cody, 2006; MacArthur & Connell, 1966). If the island's area and/or the distance (isolation) between the source and recipient islands are considered in the previous nonlinear model, then the equilibrium location $\hat{S}$ will be shifted to the right or left side, see Fig.(4.3) (MacArthur & Connell, 1966). Thus, the type of rate functions can shift $\hat{S}$ (Lomolino et al., 2009; MacArthur & Wilson, 1967; Losos & Ricklefs, 2010). The advantage of using different rate functions will be covered later.

I and E are the maximum possible immigration and emigration

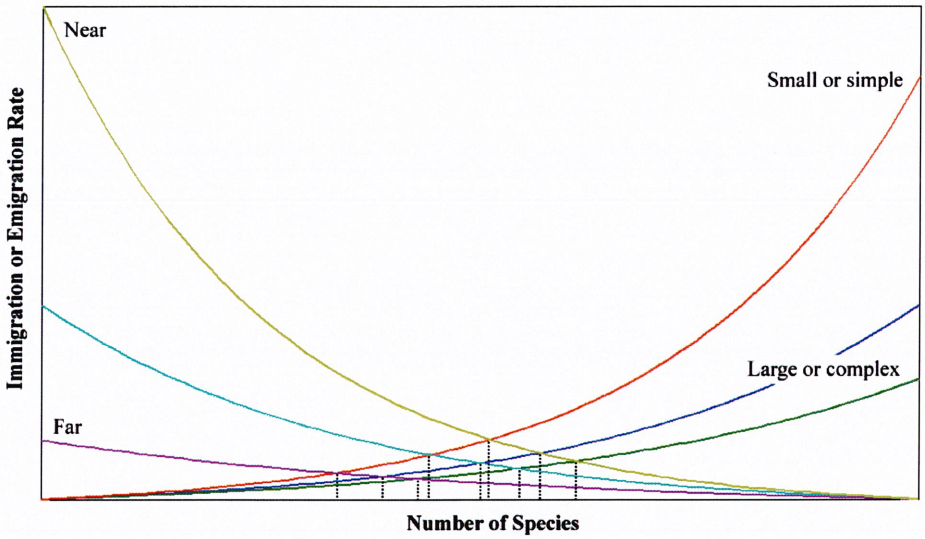


Figure (4.3)
Equilibrium model of faunas of different islands

rates, respectively. $\hat{S}$ is the number of species at equilibrium, $\hat{T}$ is the species turnover rate at equilibrium, and S_{max} is the maximum number of species on that island.

I occurs when there is no colonization process, or in other word, the island is empty and offers maximum opportunity to the species on the other islands for immigrating to settle on it; and as the number of arrived species on that island increases, the opportunity for settlement will decrease and thus the immigration rate λ will decrease too. Based on that, the species density increases, so the predation, competition and parasitism factors will increase; and as a result, the emigration rate μ will increase, and reaches its maximum value E at the minimum value of λ (MacArthur, 1972).

MacArthur and Wilson (MacArthur & Wilson, 1963, 1967) simplified the exponential model to be as a linear function, where $I = E$ as

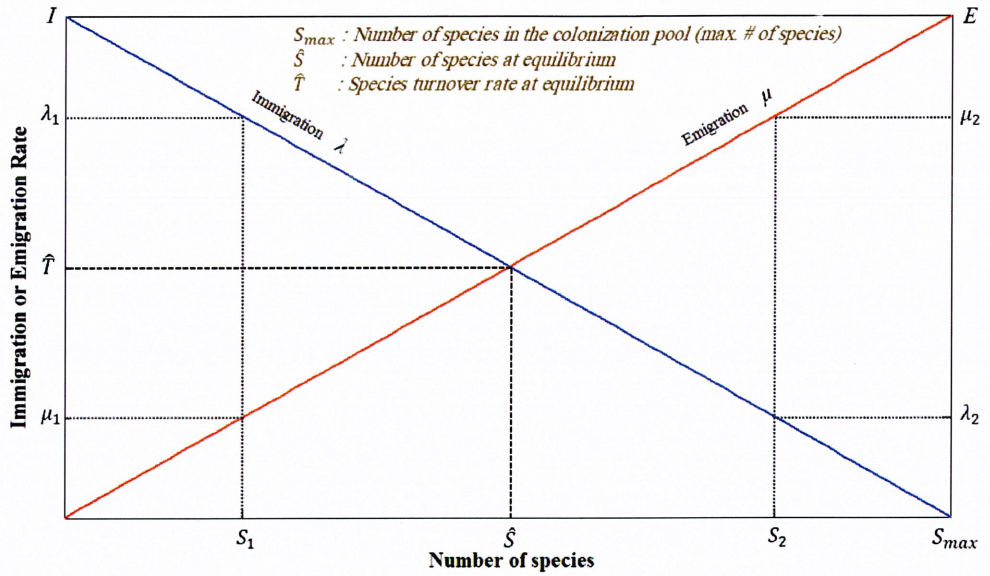


Figure (4.4)
Simplified equilibrium model of a biota of a single island

shown in Fig.(4.4) with mathematical expressions in order to theoretically explain how the migration process on a single island happens.

Now, let at time t, the recipient island has S species with probability $P_s(t)$, and λ_s and μ_s are respectively the immigration and emigration rates at the present of S species on that island. Then the variation from $P_s(t)$ to $P_s(t+\Delta t)$ can be described as:

$$P_s(t+\Delta t) = P_s(t)(1-\lambda_s\Delta t-\mu_s\Delta t)+P_{s-1}(t)\lambda_{s-1}\Delta t+P_{s+1}(t)\mu_{s+1}\Delta t \tag{4.2}$$

Also, $\hat{S}$ can be found by using different methods. From the basic of trigonometry:

$$\frac{\hat{T}}{\hat{S}} = \frac{E}{S_{max}} \Rightarrow \hat{T} = \left(\frac{E}{S_{max}}\right)\hat{S} \tag{4.3}$$

$$\frac{\hat{T}}{(S_{max}-\hat{S})} = \frac{I}{S_{max}} \tag{4.4}$$

Substituting Eq.(4.3) in Eq.(4.4) for $\hat{T}$:

$$\hat{S} = \left(\frac{I}{I+E}\right) S_{max} \tag{4.5}$$

Equation (4.5) can also be obtained by equalizing λ_s and μ_s rates at $\hat{S}$ as follows:

$$\mu_s = \left(\frac{E}{S_{max}}\right) S \tag{4.6}$$

$$\lambda_s = 1 - \mu_s = I\left(1 - \frac{S}{S_{max}}\right) \tag{4.7}$$

$$at\ S = \hat{S}:\ \lambda_s = \mu_s \Rightarrow I\left(1 - \frac{\hat{S}}{S_{max}}\right) = \left(\frac{E}{S_{max}}\right)\hat{S} \tag{4.8}$$

Solving Eq.(4.8) for $\hat{S}$ gives Eq.(4.5). The third method is by finding the equilibrium points from the following differential equation:

$$\frac{dS}{dt} = \lambda_s - \mu_s \tag{4.9}$$

Substituting Eq.(4.6) and Eq.(4.7) in Eq.(4.9) for λ_s and μ_s:

$$\frac{dS}{dt} = I - \left(\frac{I}{S_{max}}\right) S - \left(\frac{E}{S_{max}}\right) S \tag{4.10}$$

At the equilibrium point, Eq.(4.10) will be equal to zero. If it is solved at that point, then $S = \hat{S}$ and Eq.(4.5) can be obtained. This

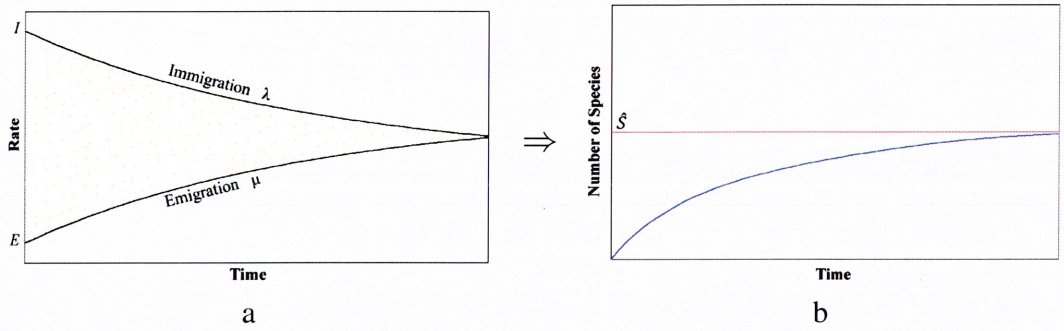


Figure (4.5)
The integrated difference through time of λ and μ in (a) will give the colonization curve in (b)

intersection point is the core of the equilibrium theory, where the island's biota will be at a state of dynamic equilibrium, and thus $P_s(t + \Delta t) = P_{max}$ (Simon, 2008a; MacArthur & Wilson, 1967; MacArthur & Connell, 1966). One of the interesting things when $S = \hat{S}$ is that the island will remain in its normal species density.

From the preceding information, the colonization curve can be plotted easily, either by subtracting the emigration rate from the immigration rate, as in Eq.(4.9), or by solving Eq.(4.10) for $S(t)$ as:

$$S(t) = \hat{S}\left[1 - e^{-\left(\frac{I+E}{S_{max}}\right)t}\right] \tag{4.11}$$

The derivation of Eq.(4.11) is in Appendix A, and the colonization curve is shown in Fig.(4.5).

Moreover, the birth and death rates can be considered, because the population density in the islands can be markedly affected by these two rates (MacArthur, 1972). Also, to conserve the communities, the predator-prey species should not have self-limitation (Svirezhev & Logofet, 1983).

All of these good fundamental information can be used as additional tools for future modifications on the existing BBO models, because BBO still is a new algorithm and offers a lot of rooms for further researches. For example, some of the complicated BBO models mainly focused on how to play on the location of the equilibrium point for increasing the probabilities to find the best solution (Ma et al., 2009; Ma, 2010; Ma & Simon, 2011a).

Now, from Eq.(4.2), to have S at time $(t+\Delta t)$, one of the following three conditions should hold:

1. S species at time t, and no immigration or emigration took place during the interval Δt;
2. $(S-1)$ species at time t, and one species immigrated;
3. $(S+1)$ species at time t, and one species emigrated.

To neglect the probability of more than one immigration or emigration, then Δt has to be set with small value. Also, from the basic of calculus, as Δt approaches 0, the ratio $\left(\frac{\Delta P_s}{\Delta t}\right)$ approaches $\dot{P}_s(t)$:

$$\begin{aligned}
\dot{P}_s(t) &\cong \lim_{\Delta t \to 0} \frac{P_s(t+\Delta t) - P_s(t)}{\Delta t} \\
&\cong \lim_{\Delta t \to 0} \left(\frac{[P_s(t) - P_s(t)\lambda_s \Delta t - P_s(t)\mu_s \Delta t + P_{s-1}(t)\lambda_{s-1}\Delta t + P_{s+1}(t)\mu_{s+1}\Delta t]}{\Delta t} \right. \\
&\qquad \left. - \frac{P_s(t)}{\Delta t} \right) \\
&\cong \lim_{\Delta t \to 0} \frac{[-P_s(t)\lambda_s - P_s(t)\mu_s + P_{s-1}(t)\lambda_{s-1} + P_{s+1}(t)\mu_{s+1}]\,\Delta t}{\Delta t} \\
&\cong -(\lambda_s + \mu_s)P_s(t) + \lambda_{s-1}P_{s-1}(t) + \mu_{s+1}P_{s+1}(t) \qquad (4.12)
\end{aligned}$$

By considering the above three conditions, Eq.(4.12) becomes:

$$\dot{P}_s(t) = \begin{cases} -(\lambda_s + \mu_s)P_s(t) + \mu_{s+1}P_{s+1}(t), & \text{if } S = 0 \\ -(\lambda_s + \mu_s)P_s(t) + \lambda_{s-1}P_{s-1}(t) + \mu_{s+1}P_{s+1}(t), & \text{if } 1 \leqslant S \leqslant S_{max} - 1 \\ -(\lambda_s + \mu_s)P_s(t) + \lambda_{s-1}P_{s-1}(t)), & \text{if } S = S_{max} \end{cases} \quad (4.13)$$

Thus, using the known values of $P_s(t)$ and $\dot{P}_s(t)$, the value of $P_s(t+\Delta t)$, that is given in Eq.(4.2), can be approximated as:

$$P_s(t + \Delta t) \cong P_s(t) + \dot{P}_s(t)\Delta t \quad (4.14)$$

Equation (4.14) is the final form that has to be used in the program of BBO for calculating $P_s(t + \Delta t)$. For finding $P_s(t)$, Dan. Simon in (Simon, 2008a) solved Eq.(4.13) iteratively.

The second approach, that was also proposed by Dan. Simon in (Simon, 2008a), is done through a direct way using eigenvector technique. It is described as:

For simplification, suppose S_{max} equal to the number of islands $\bar{k}$ minus one (i.e., $z = S_{max} = \bar{k} - 1$), then the value of $\dot{P}_s(t)$ in Eq.(4.13) can also be determined as:

$$\dot{P} = AP \ , \quad \text{where: } P = [P_0, P_1, \cdots, P_z]^T$$

$$A = \begin{bmatrix} -(\lambda_0 + \mu_0) & \mu_1 & 0 & \cdots & 0 \\ \lambda_0 & -(\lambda_1 + \mu_1) & \mu_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \lambda_{z-2} & -(\lambda_{z-1} + \mu_{z-1}) & \mu_z \\ 0 & \cdots & 0 & \lambda_{z-1} & -(\lambda_z + \mu_z) \end{bmatrix} \quad (4.15)$$

With the special case of the given simplified migration model (where $E = I$), the previous matrix can be defined as:

$$A = EA' = E \begin{bmatrix} -1 & \frac{1}{z} & 0 & \cdots & 0 \\ \frac{z}{z} & -1 & \frac{2}{z} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \frac{2}{z} & -1 & \frac{z}{z} \\ 0 & \cdots & 0 & \frac{1}{z} & -1 \end{bmatrix} \tag{4.16}$$

It can be observed that the zero eigenvalue of A' happens with the following eigenvector:

$$\begin{aligned} v &= [v_1, v_2, \cdots, v_{z+1}]^T \\ v_w &= \begin{cases} \dfrac{z!}{(z+1-w)!(w-1)!}, & \text{for } w = 1, 2, \cdots, w' \\ v_{z+2-w}, & \text{for } w = w'+1, w'+2, \cdots, z+1 \end{cases} \\ &\text{where: } w' = \left\lceil \frac{z+1}{2} \right\rceil \end{aligned} \tag{4.17}$$

Also, the $(z + 1)$ eigenvalues of the biogeography matrix A' are computed as:

$$e = \left\{ 0, \frac{-2}{z}, \frac{-4}{z}, \cdots, -2 \right\} \tag{4.18}$$

Equation (4.18) was proven later in (Igelnik & Simon, 2011). This matrix technique is easier than the iterative technique, because it can give a direct way to find the steady-state probability $P(\infty)$ by applying the following theorem (Simon, 2008a):

Theorem 1: *The steady-state value for the probability of the number of each species is given by:*

$$P(\infty) = \frac{v}{\sum_{w=1}^{z+1} v_w} \tag{4.19}$$

Although the second alternative is easier and $P(\infty)$ can be computed directly without any iteration, this method is not preferable in the most popular programs, such as C/C++, MATLAB, Octave, Maple, Python, etc, because it is valid only when $z \leqslant 170$, otherwise $z! = \infty$, unless an additional sub-algorithm is used to overcome this infinity problem. In addition, this approach will consume extra CPU time for dealing with long product operations. Based on that, the iterative method is flexible and more convenient.

4.3 Biogeography-Based Optimization (BBO)

BBO translates the natural distribution of species into a general problem solution (Simon, 2008a). Each island represents one solution, where the good problem solution means that the island has lots of good biotic "*living: diversity of prey, trees, shrubs, meadow, etc*" and abiotic "*non-living: distance of isolation, wind, temperature, humidity, water, area, etc*" factors (Svirezhev & Logofet, 1983), which attracts more species than the other islands (Lomolino et al., 2009). Each one of these features is called **suitability index variable** (SIV), which represents

the independent variable of such a problem in BBO. As these features change, the **island suitability index** (ISI) changes too; thus in BBO, ISI is the dependent variable (Simon, 2008a; Simon et al., 2009). Based on that, a problem with n-independent variables and $\bar{k}$-islands or population size can be expressed as:

$$ISI_i = f_i(SIV_1, SIV_2, \cdots, SIV_n) \quad , \quad i = 1, 2, \cdots, \bar{k} \tag{4.20}$$

The algorithm of BBO consists of two main sub-algorithms, called **migration** and **mutation**.

4.3.1 Migration Stage

Considering Eq.(4.20), the high ISI for island i represents a good solution, and also high ISI means large number of available species on that island, which forces immigration rate λ_s to be low and emigration rate μ_s to be high; while low ISI for island i represents a poor solution, which means a shortage indication in the availability of species on that island, where at this condition λ_s is high and μ_s is low.

Referring to Fig.(4.4), S_1 is located before $\hat{S}$, where λ_s is high, μ_s is low and the solution ISI_1 is poor; while S_2 is located after $\hat{S}$, where λ_s is low, μ_s is high and the solution ISI_2 is good. Thus, λ_s and μ_s are indications of poor and good solutions, respectively.

In migration process, the high ISI islands share their features to modify the low ISI islands, where the islands of both sides are proba-

bilistically selected. The high ISI islands become the source of modification, while the low ISI islands become the recipients to those emigrated species.

Although the species will emigrate from the rich islands to the poor islands, this phenomenon does not mean that the species will completely disappear from its home islands. However, only a few representatives emigrate (Simon, 2008a). Thus, the recipient islands are enhanced, and at the same time the source islands are kept away from any shortage on its richness of species.

In the early stages of introducing BBO, Dan. Simon proposed four different types of migration process, these types can be sorted as (Simon, 2011):

4.3.1.1 Partial Migration-Based BBO (PMB-BBO)

This is the first BBO algorithm that was presented in Dec. 2008 (Simon, 2008a). The pseudocode of this model is presented in Algorithm 2.

4.3.1.2 Single Migration-Based BBO (SMB-BBO)

This is the second BBO algorithm that was presented in Feb. 2009 (Simon, 2011). In this algorithm, only one randomly selected SIV is modified instead of doing *for-loop* for all n features. It can be described as in Algorithm 3.

Algorithm 2 The pseudocode of Partial Migration-Based BBO (PMB-BBO)

Require: Let ISI_i denote the ith population member and contains n features

for $i \leftarrow 1$ *to* $\bar{k}$ **do** {where $\bar{k}$ = number of islands or individuals}
  for $s \leftarrow 1$ *to* n **do** {where n = number of features "SIV" or design variables}
    Use λ_i to probabilistically select the immigrating island ISI_i
    for $j \leftarrow 1$ *to* $\bar{k}$ **do**
      Use μ_j to probabilistically decide whether to emigrate to ISI_i
      if ISI_j is selected **then**
        Randomly select an SIV σ from ISI_j
        Replace a random SIV s in ISI_i with SIV σ
      end if
    end for
    next SIV
  end for
  next island
end for

Algorithm 3 The pseudocode of Single Migration-Based BBO (SMB-BBO)

Require: Let ISI_i denote the ith population member and contains n features

for $i \leftarrow 1$ *to* $\bar{k}$ **do** {where $\bar{k}$ = number of islands or individuals}
  Use λ_i to probabilistically select the immigrating island ISI_i
  Pick a random SIV s {where $s = 1, 2, \cdots, n$}
  for $j \leftarrow 1$ *to* $\bar{k}$ **do**
    Use μ_j to probabilistically decide whether to emigrate to ISI_i
    if ISI_j is selected **then**
      Randomly select an SIV σ from ISI_j
      Replace a random SIV s in ISI_i with SIV σ
    end if
  end for
  next island
end for

4.3.1.3 Simplified Partial Migration-Based BBO (SPMB-BBO)

This simplified model is similar to the previous normal model "PMB-BBO", except that the simplified model will always select the best obtained island as the source of modification for those poor islands instead of doing an internal loop checking, see Algorithm 4.

Algorithm 4 The pseudocode of Simplified Partial Migration-Based BBO (SPMB-BBO)

Require: Let ISI_i denote the ith population member and contains n features

1: **for** $i \leftarrow 1$ *to* $\bar{k}$ **do** {where $\bar{k}$ = number of islands or individuals}
2: **for** $s \leftarrow 1$ *to* n **do** {where n = number of features "SIV" or design variables}
3: Use λ_i to probabilistically select the immigrating island ISI_i
4: eSelect the best obtained island (ISI_{best}) as the source of the emigrating SIV
5: **if** ISI_{best} is selected **then**
6: Randomly select an SIV σ from ISI_{best}
7: Replace a random SIV s in ISI_i with SIV σ
8: **end if**
9: next SIV
10: **end for**
11: next island
12: **end for**

It is apparent that the simplified model has two conflicting issues:

- It is faster (less CPU time) than PMB-BBO, because the internal looping for probabilistically selecting the good ISI is eliminated.
- However, it could trap in a local minima because it always depend on the best ISI, and consequently the probability of finding other better solutions reduces.

4.3.1.4 Simplified Single Migration-Based BBO (SSMB-BBO)

In the same way, this simplified model is similar to SMB-BBO, except the internal looping for selecting the source ISI that contains the good n SIV to be emigrated to the poor islands. This model is described in Algorithm 5.

Algorithm 5 The pseudocode of Simplified Single Migration-Based BBO (SSMB-BBO)

Require: Let ISI_i denote the ith population member and contains n SIV

1: **for** $i \leftarrow 1$ *to* $\bar{k}$ **do** {where $\bar{k}$ = number of islands or individuals}
2: Use λ_i to probabilistically select the immigrating island ISI_i
3: Pick a random SIV s {where $s = 1, 2, \cdots, n$}
4: Select the best obtained island (ISI_{best}) as the source of the emigrating SIV
5: **if** ISI_{best} is selected **then**
6: Randomly select an SIV σ from ISI_{best}
7: Replace a random SIV s in ISI_i with SIV σ
8: **end if**
9: next island
10: **end for**

4.3.2 Mutation Stage

In the island theory, the species at the equilibrium point $\hat{S}$ can be deviated dramatically due to some external random events called mutations (Floreano & Mattiussi, 2008). Events such as predators from other islands, diseases, tsunamis, volcanos and/or earthquakes cause negative deviation, and the total number of species will decrease steeply (MacArthur & Connell, 1966). On the other hand, there are some useful events that

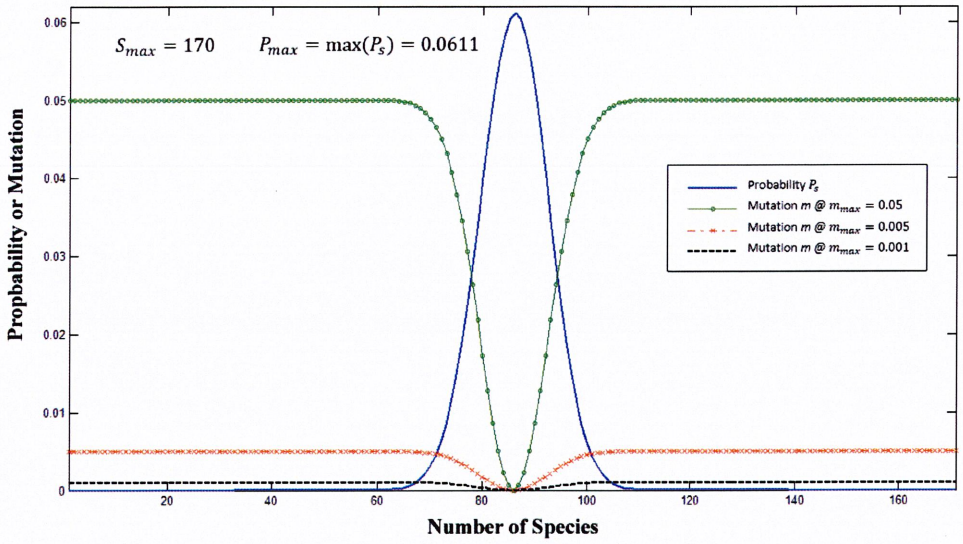


Figure (4.6)
Comparison between $P_s(t)$ and $m(t)$ at different m_{max}

can improve those n SIV to give better solutions, such as wind-carrying seeds (wind pollination) or flotsam (shipwreck) (MacArthur, 1972). In BBO, the mutation process is modeled as SIV mutation; and through species count probabilities P_s, the mutation rate m can be determined as (Simon, 2008a):

$$m = m_{max}\left(1 - \frac{P_s}{P_{max}}\right) \tag{4.21}$$

where $P_{max} = max(P_s)$ and m_{max} is a user-defined maximum mutation rate that m can reach.

From Eq.(4.21), m reaches to its minimum "zero" at the maximum value of P_s, and vice versa. Thus, m is inversely proportional to P_s ($m \propto^{-1} P_s$). This process can be graphically described as in Fig.(4.6), where the species count S starts from zero to S_{max}. As m_{max} increases, the chance to let the solutions be mutated increases too.

During the mutation stage, the low and high ISI solutions are likely to mutate, and then could be enhanced more than what they already have, where the solutions at the equilibrium point "$\hat{S}$" are not mutated (Simon, 2008a). Even if the mutated solutions become worse, the optional stage, called **elitism**, will store the best solutions from one generation to the next (Simon et al., 2009). The mutation process can be described as in Algorithm 6.

Algorithm 6 The pseudocode of the mutation stage

1: **for** $i \leftarrow 1$ *to* $\bar{k}$ **do** {where $\bar{k}$ = number of islands or individuals, see Eq.(4.20)}
2: Calculate probability P_s based on λ_s and μ_s {by iterative or eigenvector method}
3: Calculate mutation rate m {using Eq.(4.21)}
4: **if** rand $< m(i)$ and $i \geqslant \%R_m$ **then** {$\%R_m$ is a user defined mutation range}
5: Replace SIV of $ISI_s(i)$ with a randomly generated SIV
6: **end if**
7: **end for**

4.3.3 Clear Duplication Stage

This stage is an optional in the BBO algorithm. Although it was included in the code of the original BBO (Simon, 2008b), there is no enough details about it in the literature. This stage is very essential, because it increases the diversity of the generated solutions by avoiding the features to be duplicated on the other $(\bar{k} - 1)$ islands. Thus, without this stage, BBO algorithm may lose the chance to get better solutions due to these inactive individuals. It can be simply described in Algorithm 7.

Algorithm 7 The pseudocode of the clear duplication stage

Require: Check all n SIV on all $\bar{k}$ ISI

```
while there is a duplicated SIV do
    for i ← 1 to k̄ do {where k̄ = number of islands or individuals, see Eq.(4.20)}
        if any duplicated SIV s is detected then
            Replace the duplicated SIV s in ISI_i with a randomly generated SIV σ
        end if
    end for
end while
```

4.3.4 The Overall BBO Algorithm

The steps of the general BBO algorithm can be listed as:

1. Initialize the BBO parameters ($\bar{k}$, I, E, m_{max}, etc).
2. Find species count probabilities P_s and mutation rate m based on the calculated emigration rate μ_s and immigration rate λ_s by Eqs.(4.6)-(4.7).
3. Generate $\bar{k}$ random islands, where each island represents one solution to a given n-dimensional problem with non duplicated SIV.
4. Sort the solutions (i.e., $\bar{k}$ ISI), so that the first best solution ISI_1 should be mapped with the highest number of species and the lowest immigration rate λ_s (or the highest emigration rate μ_s), and continue the descending order till reaching to the worst solution which is $ISI_{\bar{k}}$.
5. Do elitism process for saving the required best solutions for the next generation; it is an optional step (Simon et al., 2009).

6. Probabilistically select the source islands based on μ_s, and the islands that need to be modified "the recipient islands" based on λ_s, and do the migration process. Then, update all $\bar{k}$ ISI before ending this step.

7. Do mutation process for the islands based on their probabilities that are listed in the probability vector after calculated in step (2). After that, clear any duplicated SIV once the mutation process is completed. Then, update all $\bar{k}$ ISI.

8. Return to step (4) for the next iteration. This loop can be terminated either if it reaches to an acceptable tolerance (*i.e., error on the optimized objective function*) or after the desired number of generations is completed.

These steps of the BBO algorithm are summarized in the flowchart of Fig.(4.7).

4.3.5 Performance Comparison of the four Original BBO Forms

The main problem associated with all the modified BBOs is that the modifications were done on an arbitrary selected form of the four original BBO forms. There is no clarification on which form the proposed modification is based on and why.

The four original forms of BBO have been tested through 23 benchmark functions with different dimensions and complexities. These functions can be classified into three groups: **unimodal**, **multimodal with**

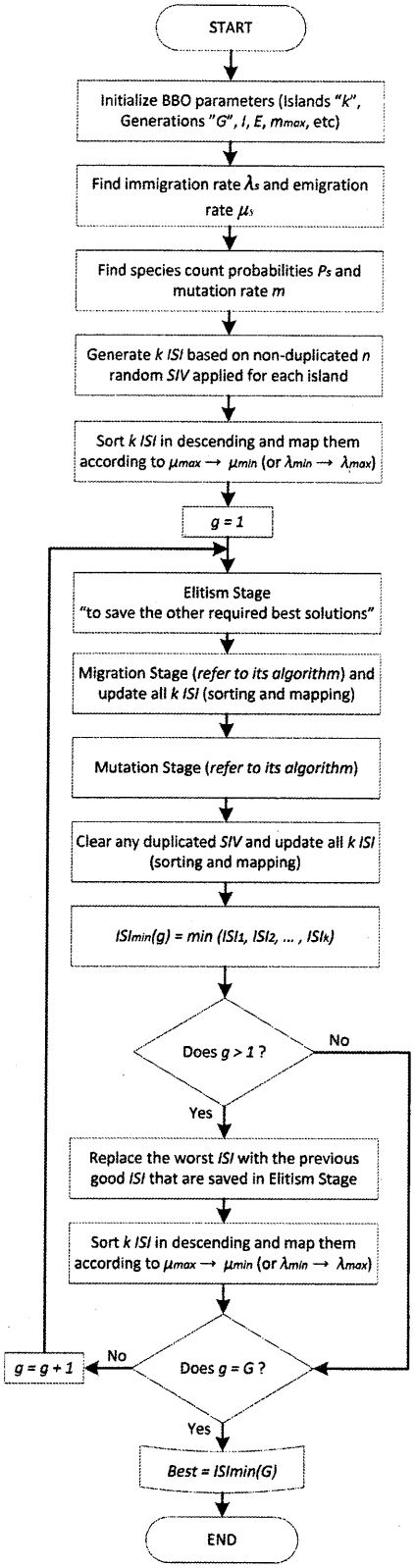


Figure (4.7)
General flowchart of the BBO algorithm

few local minima and **multimodal with many local minima**. Functions $\{f30\text{-}01$ to $f30\text{-}15\}$ are $high$-dimensional problems. Functions $\{f30\text{-}01$ to $f30\text{-}05\}$ and $f30\text{-}07$ are $high$-dimensional and multimodal with many local minima. Functions $f30\text{-}06$ and $\{f30\text{-}10$ to $f30\text{-}14\}$ are $high$-dimensional and unimodal. Function $f30\text{-}15$ is a $high$-dimensional step function with one discontinuous minimum. The remaining functions are low-dimensional and multimodal with few local minima (Gong, Cai, & Ling, 2010). More details about these benchmark functions are given in Appendix E.

The parameters that have been used here are similar to those used in (Ma, 2010; Ma & Simon, 2011a): population size of 50, $I = E = 1$, $m_{max} = 0.01$, generations limit of 20,000 for the $high$-dimensional problems and 1000 for the low-dimensional problems, elitism parameter of 1, and Monte-Carlo simulation with 30 trials.

Table 4.1 summarizes the performance of PMB, SMB, SPMB and SSMB models for 23 benchmark functions. The highlighted cells in the tables represent the best results among the four BBO algorithms. It can be clearly seen that the performance of PMB and SPMB are superior as compared to SMB and SSMB. For $high$-dimensional problems, PMB wins with 7 best solutions, 10 mean and 8 standard deviation out of 13; while SPMB wins with 6 best solutions, 3 mean and 5 standard deviation. On the other hand, for low-dimensional problems, SSMB enters this competition, and gives better Best, Mean and Standard deviation than

Table (4.1)

Comparison of the results for 30 trials of the four original BBO models, where Best, Mean, and StdDev stands for the smallest error, the mean of all the errors, and the standard deviation, respectively

f#	*n*	**Biogeography Based Optimization (BBO)**											
		Partial Migration Based			**Single Migration Based**			**Simplified Partial Migration Based**			**Simplified Single Migration Based**		
		Best	**Mean**	**StdDev**	**Best**	**Mean**	**StdDev**	**Best**	**Mean**	**StdDev**	**Best**	**Mean**	**StdDev**
f1-03	**1**	9.8030E-08	**2.8723E-05**	**4.2939E-05**	For 1-dimensional problems, SMB-BBO is not applicable; (SMB=PMB)			**3.8113E-09**	3.3121E-05	6.0656E-05	For 1-dimensional problems, SSMB-BBO is not applicable; (SSMB=SPMB)		
f1-04	**1**	**8.1958E-10**	**2.1546E-05**	5.2185E-05				2.2393E-08	3.4060E-05	**4.6055E-05**			
f1-05	**1**	**1.7454E-08**	7.4624E-05	1.8726E-04				1.0113E-07	**3.7714E-05**	**5.3672E-05**			
f2-10	**2**	5.6480E-07	9.6466E-05	1.6743E-04	3.6639E-06	3.4432E-04	3.9477E-04	**1.2486E-07**	**1.9438E-05**	**2.7087E-05**	4.4129E-07	7.3364E-05	9.1485E-05
f2-29	**2**	2.7778E-05	1.5749E-03	1.6278E-03	6.0816E-05	5.8628E-03	6.6169E-03	**1.2784E-07**	**8.1736E-05**	**8.7004E-05**	9.2851E-06	2.9315E-04	3.7882E-04
f2-48	**2**	2.1720E-11	8.0558E-08	2.1865E-07	6.4942E-10	1.4595E-04	5.4076E-04	**6.1392E-12**	**3.8090E-10**	**3.2748E-10**	8.4939E-11	1.0718E-07	5.3020E-07
f2-60	**2**	2.3455E-07	9.1217E-05	1.2481E-04	1.6640E-05	4.1018E-04	4.3907E-04	**3.5869E-10**	**4.8485E-06**	**4.7243E-06**	5.7770E-08	2.7089E-05	3.9141E-05
f3-02	**3**	7.6177E-06	5.0374E-04	3.8211E-04	2.5507E-04	1.6343E-03	1.2888E-03	**5.9005E-07**	**5.0688E-05**	**5.9545E-05**	1.6327E-06	5.2029E-04	4.9816E-04
f4-02	**4**	3.9927E-04	8.0573E-04	3.2529E-04	5.6999E-04	1.3729E-03	4.5011E-04	**1.8923E-04**	**7.0160E-04**	**3.1929E-04**	4.5865E-04	1.3658E-03	5.8181E-04
f6-01	**6**	**1.8781E-03**	**6.9158E-02**	6.0706E-02	3.6659E-02	1.6534E-01	7.1746E-02	3.3727E-03	7.9192E-02	**5.6884E-02**	2.1696E-02	1.5974E-01	6.5199E-02
f30-01	**30**	6.3026E-01	**9.9236E-01**	**2.3065E-01**	4.3719E+00	5.7651E+00	6.6010E-01	**4.9817E-01**	1.0345E+00	2.5990E-01	4.6872E+00	5.9257E+00	6.9267E-01
f30-02	**30**	**8.6708E-01**	**1.0263E+00**	3.4402E-02	2.1095E+00	4.4318E+00	1.3255E+00	9.4245E-01	1.0357E+00	**2.7730E-02**	2.1708E+00	5.0205E+00	1.9755E+00
f30-03	**30**	5.0934E-03	**2.9591E-02**	**3.2559E-02**	1.0493E+00	3.5616E+00	3.8957E+00	**3.8794E-03**	3.1228E-02	3.5340E-02	1.4325E+00	6.3746E+01	3.2648E+02
f30-04	**30**	9.3001E-02	**1.6875E-01**	**6.0757E-02**	5.3779E+00	3.1883E+03	1.3051E+04	**8.1548E-02**	2.0015E-01	8.0590E-02	6.9544E+00	2.3081E+03	8.1029E+03
f30-05	**30**	**9.4594E-01**	1.9351E+00	6.4580E-01	2.3807E+01	3.3102E+01	5.6152E+00	1.0236E+00	**1.9211E+00**	**5.7860E-01**	2.6082E+01	3.4796E+01	6.4496E+00
f30-06	**30**	**1.0517E+02**	**2.6058E+02**	**7.9555E+01**	5.0807E+03	2.3564E+04	1.6310E+04	1.3597E+02	2.7246E+02	1.1116E+02	7.4418E+03	3.9130E+04	3.2640E+04
f30-07	**30**	**4.4291E-06**	1.3876E-05	**5.7466E-06**	3.5004E-04	1.6463E-03	1.2669E-03	5.0828E-06	**1.3322E-05**	7.4610E-06	5.9656E-04	1.8382E-03	1.0377E-03
f30-10	**30**	2.8562E-07	**1.2406E-06**	1.2872E-06	7.6867E-04	1.2135E-02	1.4628E-02	**1.9837E-07**	1.3833E-06	**1.1343E-06**	1.0547E-03	1.3451E-02	1.4276E-02
f30-11	**30**	**4.2433E+02**	**1.9527E+04**	**1.1268E+04**	3.3679E+03	3.0638E+04	1.3227E+04	1.9048E+03	3.2307E+04	8.3231E+03	4.4645E+02	2.7268E+04	1.4724E+04
f30-12	**30**	3.2007E+00	6.2387E+00	**1.1469E+00**	2.3978E+01	3.4745E+01	6.0946E+00	**2.8725E+00**	**5.6856E+00**	1.2431E+00	2.8262E+01	3.8791E+01	4.1634E+00
f30-13	**30**	**4.0024E-01**	**7.0184E-01**	1.3458E-01	3.8978E+00	6.1515E+00	1.3348E+00	5.1678E-01	7.6268E-01	**1.0545E-01**	5.4535E+00	7.3952E+00	1.1782E+00
f30-14	**30**	**1.8518E+00**	**3.8843E+00**	1.3964E+00	1.6999E+02	3.1652E+02	9.9742E+01	2.2588E+00	4.3027E+00	**1.3073E+00**	8.1588E+01	3.5592E+02	1.5209E+02
f30-15	**30**	2.0000E+00	**4.7333E+00**	**1.8245E+00**	1.7400E+02	3.9163E+02	2.0465E+02	**1.0000E+00**	4.7667E+00	2.4315E+00	1.3900E+02	4.5263E+02	2.8296E+02

Table (4.2)
Normalized CPU time of the *high*-dimensional problems "f30-01 to f30-15"

f#	BBO Models			
	PMB-BBO	SMB-BBO	SPMB-BBO	SSMB-BBO
f30-01	1.4299E+00	1.0052E+00	1.2815E+00	**1.0000E+00**
f30-02	1.4256E+00	1.0070E+00	1.2876E+00	**1.0000E+00**
f30-03	1.3703E+00	1.0056E+00	1.2437E+00	**1.0000E+00**
f30-04	1.3878E+00	1.0021E+00	1.2505E+00	**1.0000E+00**
f30-05	1.4621E+00	1.0068E+00	1.3041E+00	**1.0000E+00**
f30-06	1.4717E+00	1.0074E+00	1.3133E+00	**1.0000E+00**
f30-07	1.4521E+00	1.0074E+00	1.2969E+00	**1.0000E+00**
f30-10	1.4089E+00	1.0070E+00	1.2769E+00	**1.0000E+00**
f30-11	1.1672E+00	1.0082E+00	1.1104E+00	**1.0000E+00**
f30-12	1.5018E+00	1.0086E+00	1.3355E+00	**1.0000E+00**
f30-13	1.4687E+00	1.0086E+00	1.3103E+00	**1.0000E+00**
f30-14	1.4804E+00	1.0084E+00	1.3181E+00	**1.0000E+00**
f30-15	1.4721E+00	1.0079E+00	1.3113E+00	**1.0000E+00**
Avg CPU Time	1.4230E+00	1.0070E+00	1.2800E+00	**1.0000E+00**

that of the PMB for the functions $\{f2\text{-}10, f2\text{-}29$ and $f2\text{-}60\}$, but it does not win as compared to SPMB. Single and simplified single migration based models are not valid for $\{f1\text{-}03$ to $f1\text{-}05\}$, because these problems are 1-dimensional problems, and the migration process is done within only one independent variable.

Although, in overall, SSMB has respectively the first and second worst performance for *high* and *low*-dimensional problems, it achieved the fastest algorithm as shown in Table 4.2. This is logical, because of two reasons. First, it does a migration on one randomly selected SIV for each island rather than all n SIV as in PMB and SPMB. Second, it will always select the best found solution as a source island for migration instead of doing a loop checking as in PMB and SMB. This is why the simplified versions of PMB and SMB trip in local minima particularly as the complexity, side constraints and/or dimensions increases and as the number of islands or population size decreases. In this situation, PMB

Table (4.3)
The overall performance of PMB and SPMB with/without mutation stage for $f30$-06, $f30$-07, $f2$-29 and $f2$-48

$f\#$	Function Name	n	Mutation Stage	Biogeography Based Optimization (BBO)					
				Partial Migration Based			Simplified Partial Migration Based		
				Best	Mean	StdDev	Best	Mean	StdDev
f30-06	Generalized Rosenbrock	30	Activated	**1.0517E+02**	**2.6058E+02**	**7.9555E+01**	1.3597E+02	2.7246E+02	1.1116E+02
			Not-Activated	**1.7036E+02**	**6.0492E+02**	**6.4746E+02**	2.2803E+02	7.5845E+02	8.8977E+02
f30-07	Generalized Schwefel F2.26	30	Activated	**4.4291E-06**	1.3876E-05	**5.7466E-06**	5.0828E-06	**1.3322E-05**	7.4610E-06
			Not-Activated	**8.1593E-06**	**2.9257E-05**	**1.1255E-05**	1.2890E-05	3.7150E-05	1.3349E-05
f2-29	Goldstein-Price	2	Activated	2.7778E-05	1.5749E-03	1.6278E-03	**1.2784E-07**	**8.1736E-05**	**8.7004E-05**
			Not-Activated	**1.4627E-04**	**2.5692E-02**	**2.2876E-02**	2.0330E-03	4.0344E-02	5.0549E-02
f2-48	Shekel's Foxholes	2	Activated	2.1720E-11	8.0558E-08	2.1865E-07	**6.1392E-12**	**3.8090E-10**	**3.2748E-10**
			Not-Activated	**5.4152E-08**	**2.2467E-03**	**8.6667E-03**	1.6396E-07	7.6697E-03	1.3175E-02

Table (4.4)
Performance Test I – $f30$-06 with different dimensions

$f\#$	Name	Dimensions	Biogeography Based Optimization (BBO)					
			Partial Migration Based			Simplified Partial Migration Based		
			Best	Mean	StdDev	Best	Mean	StdDev
f30-06a	Generalized Rosenbrock's Function	2	5.0569E-06	**3.0404E-04**	**2.9421E-04**	**1.1454E-06**	4.3368E-04	7.0482E-04
f30-06b		4	2.6016E+00	1.3546E+02	1.0058E+02	**3.8176E-01**	**3.0585E+01**	**3.2854E+01**
f30-06c		6	4.5589E+01	**8.1933E+02**	**7.3780E+02**	**4.0158E+01**	8.3004E+02	7.5869E+02
f30-06d		10	**2.8495E+01**	**2.1690E+02**	1.8675E+02	7.1550E+01	2.4654E+02	**1.7535E+02**
f30-06e		20	**8.7400E+01**	**3.3766E+02**	**2.4278E+02**	1.3913E+02	3.5894E+02	4.0126E+02
f30-06f		30	**1.0517E+02**	**2.6058E+02**	**7.9555E+01**	1.3597E+02	2.7246E+02	1.1116E+02

has the best exploration and exploitation, especially when the mutation stage that compensates the weakness of the migration algorithm is absent, and consequently the performance of SPMB is significantly decreased, as shown in Table 4.3.

To verify this conclusion, three performance tests have been conducted as shown in Tables 4.4, 4.5 and 4.6. Each one of these three tests is focussing on one criteria.

Test I is shown in Table 4.4, and it is used to study the performance of PMB and SPMB algorithms as the problem's dimension decreases. The parameters used for this test are similar to the previously used in Table 4.1, except that the generations limit are set as: 1000 for $n = 2, 4, 6$; 5000 for $n = 10$; $10,000$ for $n = 20$ and $20,000$ for $n = 30$.

Table (4.5)
Performance Test II – $f2$-60 and $f4$-02 with different population size

$f\#$	Name	Population Size	Biogeography Based Optimization (BBO)					
			Partial Migration Based			Simplified Partial Migration Based		
			Best	Mean	StdDev	Best	Mean	StdDev
f2-60a	6-Hump Camel-Back Function	5	**1.0557E-05**	**6.5798E-04**	**7.0515E-04**	1.1389E-05	8.7138E-04	8.0479E-04
f2-60b		10	**1.1365E-07**	2.9963E-04	4.1067E-04	2.4790E-06	**1.8534E-04**	**2.8880E-04**
f2-60c		50	2.3455E-07	9.1217E-05	1.2481E-04	**5.7770E-08**	**2.7089E-05**	**3.9141E-05**
f2-60d		200	1.7238E-07	2.2757E-05	2.7229E-05	**5.7634E-09**	**5.9924E-07**	**7.5759E-07**
f4-02a	Kowalik's Function	5	**3.0566E-04**	**1.1017E-03**	**5.6436E-04**	4.2186E-04	1.4561E-03	7.5693E-04
f4-02b		10	5.1379E-04	1.2970E-03	7.7744E-04	**2.1192E-04**	**1.0605E-03**	**7.2553E-04**
f4-02c		50	3.9927E-04	8.0573E-04	3.2529E-04	**1.8923E-04**	**7.0160E-04**	**3.1929E-04**
f4-02d		200	1.9935E-04	6.2320E-04	2.2726E-04	**1.4958E-04**	**4.4200E-04**	**1.7749E-04**

Table (4.6)
Performance Test III – $f30$-02 with different side constraints

$f\#$	Name	Side Constraints	Biogeography Based Optimization (BBO)					
			Partial Migration Based			Simplified Partial Migration Based		
			Best	Mean	StdDev	Best	Mean	StdDev
f30-02a	Generalized Griewank's Function	Xi ∈ [- 60,60]	8.6106E-02	2.3591E-01	8.7067E-02	**5.3899E-02**	**2.1482E-01**	**7.9380E-02**
f30-02b		Xi ∈ [- 600,600]	**8.7314E-01**	**1.2508E+00**	2.3871E-01	9.6562E-01	1.3044E+00	**2.2903E-01**
f30-02c		Xi ∈ [- 6000,6000]	**4.9293E+00**	**3.2079E+01**	**2.4023E+01**	6.0255E+00	3.7802E+01	2.7235E+01

Whereas, **Test II** shown in Table 4.5 is used to study the performance of PMB and SPMB algorithms as the number of islands or population size increases for two *low*-dimensional problems.

Finally, **Test III** shown in Table 4.6 is used to study the performance of PMB and SPMB algorithms under different upper and lower limits of the search space.

As can be seen from Table 4.4, the SPMB performs better as the problem's dimension decreases. But when the population size is small, the PMB will perform better even for the *low*-dimensional problems, as shown in Table 4.5. From Table 4.6, if the search space is large, the PMB wins too.

Figure (4.8) shows the curves of fitness functions of PMB, SMB,

SPMB, and SSMB for the **Generalized Griewank's function "f30-02"**, **Generalized Rastrigin's function "f30-05"**, **Generalized Rosenbrock's function "f30-06"** and **Schwefel's problem 1.2 "f30-11"**. For the functions f30-02, f30-05, f30-06 and f30-11, it can be clearly seen that the PMB and SPMB algorithms outperform the SMB and SSMB algorithms, and the gap between the convergence is large. For the function f30-11, SSMB enters this competition and shows better fitness as compared to SPMB. However, the PMB algorithm is again has the best convergence.

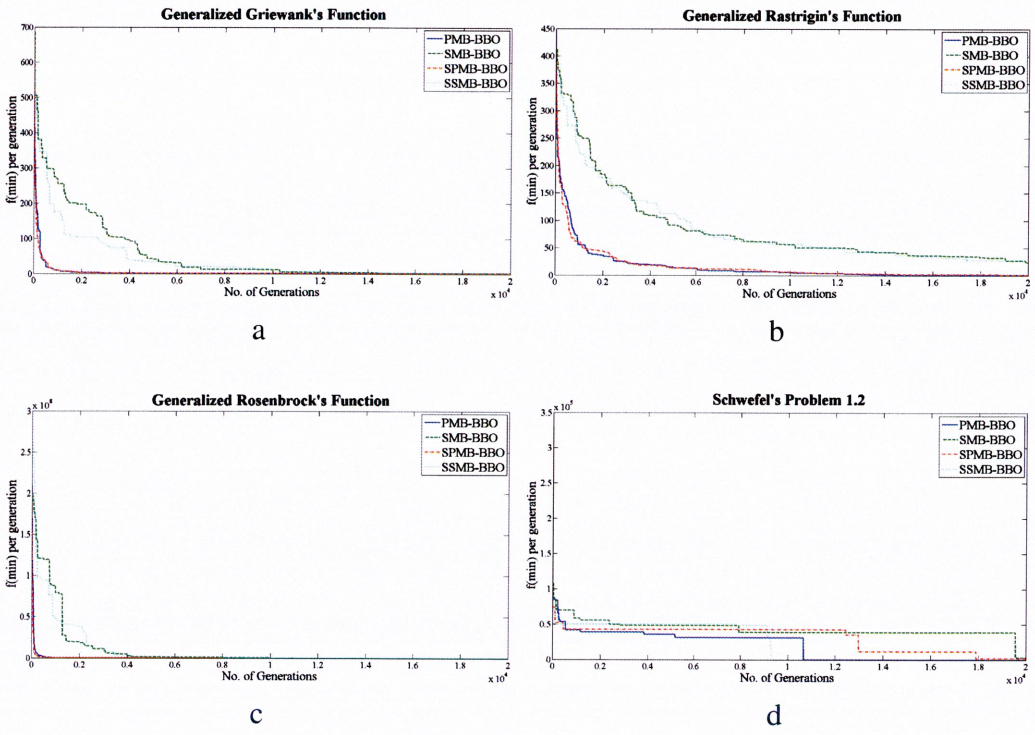


Figure (4.8)
Curves of fitness functions of PMB, SMB, SPMB, and SSMB for some selected functions. (a) f30-02, (b) f30-05, (c) f30-06 and (d) f30-11

4.3.5.1 Conclusions and Suggestions

The performance of the four original forms of BBO algorithms (PMB, SMB, SPMB and SSMB) has been extensively tested and compared using 23 benchmark functions of different dimensions and complexities, as well as different scenarios have been done for some selected functions. The results show that the PMB, which is the first presented model in 2008, gives the best performance as the complexity, side constraints and/or dimensions of a given problem increases and as the number of islands or population size decreases. However, the PMB is found to be the slowest algorithm which requires around 42.30% more CPU time than that of the SSMB as shown in Table 4.2. To compromise between the cost and convergence speed under the above scenarios, SPMB is found to be the best choice as it requires around 28.00% more CPU time, but its exploration and exploitation will decrease significantly. The performance of the SMB algorithm found to be the worst as compared to the others.

The results obtained in this study can be used as a foundation and a first step for enhancing any prospective modification on the BBO algorithm including the existing modifications that are described in the literature.

4.4 Biogeography Based Optimization with Essential Modifications (BBO-EM)

As mentioned at the beginning of this Chapter, it was found that the original BBO algorithm has some weaknesses on its exploration. This Section tries to solve the root problem itself instead of using different techniques to solve its effect. Two essential modifications are proposed. Firstly, modifying the probabilistic selection process of the migration and mutation stages to give a fairly randomized selection for all the features of the islands. Secondly, the clear duplication process, which is located after the mutation stage, is sized to avoid any corruption on the vector of SIV of those non-mutated islands. The proposed modifications are extensively tested using 120 test functions with different dimensions and complexities. The results reveal that the BBO performance can be enhanced effectively without embedding any additional sub-algorithm, and without using any complicated form of the immigration and emigration rates. In addition, the new BBO algorithm with essential modifications **BBO-EM** requires less CPU time and becomes even faster than the original simplified partial migration-based BBO. These essential modifications have to be considered as an initial step for any other modifications.

Based on the preceding conclusions, PMB-BBO model outperforms the others based on the given evaluation criteria (*hardness, search space, dimensions, population size*), while SPMB-BBO model could trap in a local or at least a near-global optima. For this reason, PMB-BBO is selected as a final confirmed BBO model for applying the proposed es-

sential modifications, which will be explained in the next Sub-Sections.

4.4.1 The Proposed Essential Modifications on BBO

This correction will cover two fundamental problems that are associated with the original version (the raw form) of BBO. It will show that the linear immigration and emigration rates, in Fig.(4.4), still can do well if the associated root problems are solved.

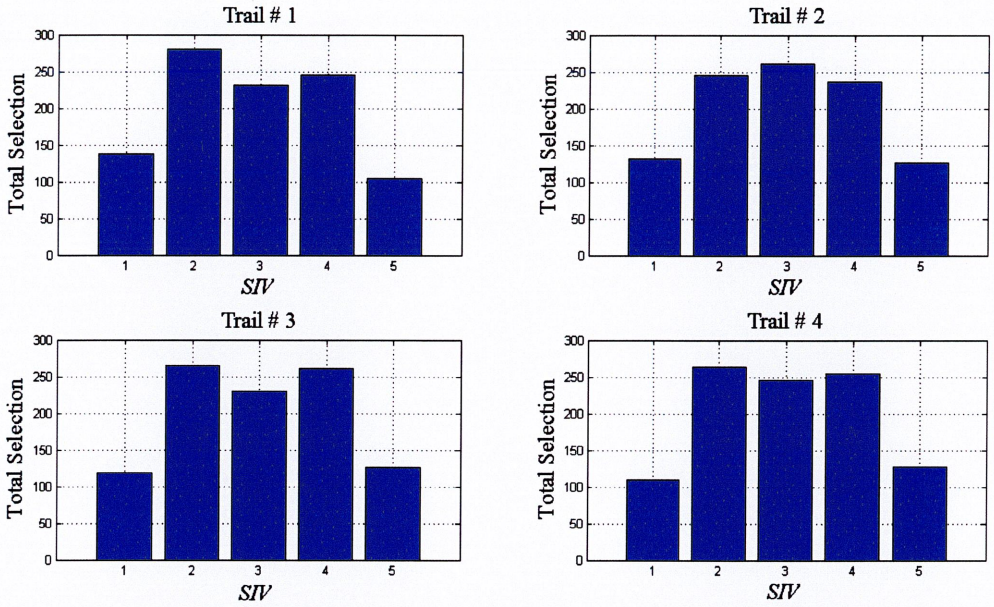
According to the original BBO program that was designed by Dan. Simon in (Simon, 2008b), there are two essential parts that need to be corrected:

4.4.1.1 Probabilistic Selection Process of the Migration and Mutation Stages

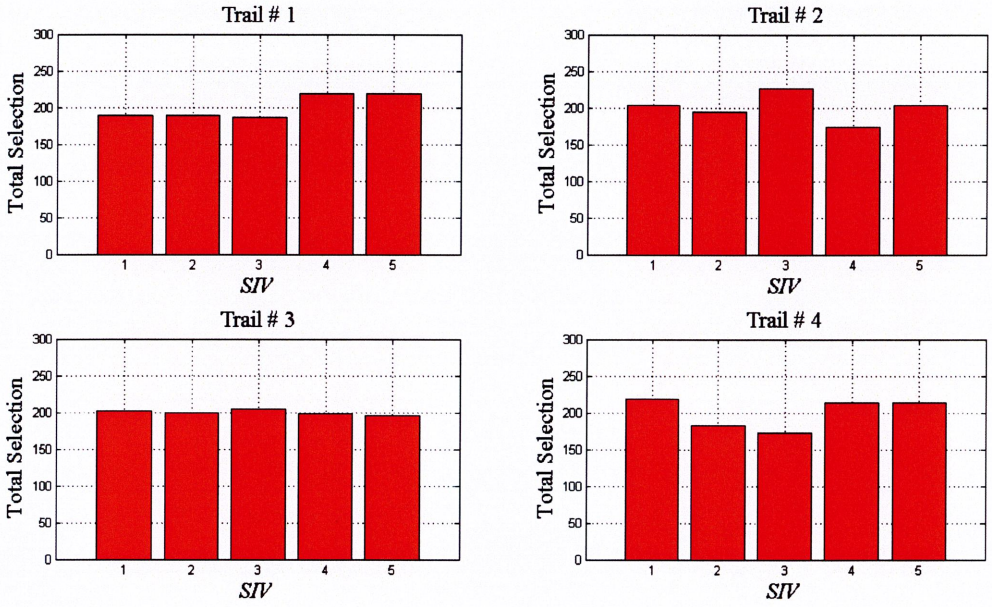
By referring to that program, the process for selecting $SIV\ s$ of an island i that needs to be migrated is done probabilistically, and the general code for this task is:

$$SIV_s(1:\bar{k}) = \text{round}\left[1 + (n-1) \times \text{rand}(1,n)\right] \qquad (4.22)$$

If the design features of each ISI are represented as a vector of $[SIV_1, SIV_2, \cdots, SIV_n]$, then the analysis of the above code shows that the $SIVs$ at the beginning and at the end (i.e. SIV_1 and SIV_n) have less weight than the other $(n-2)$ SIV that are located in between.



a: Original SIV-selection process



b: Modified SIV-selection process

Figure (4.9)
The original and modified SIV-selection process for migration and mutation stages

Figure (4.9a) shows the unfair selection (*non-uniform distribution*) for each SIV of island i. In this example; $n = 5$ and $\bar{k} = 1000$, with 4 trials since the process is done randomly.

In MATLAB, the proposed modification is to use integer random "**randi**" instead of using rounded real random "**rand**" with n SIV alignment, as in Eq.(4.22). This integer random function provides pseudorandom integers from a uniform discrete distribution on 1 to n. For getting integer random values with fairly selection for all n SIV:

$$SIV_s(1:\bar{k}) = \text{randi}\,(n, 1, n) \tag{4.23}$$

The MATLAB code given in Eq.(4.23) is equivalent to the previous code in Eq.(4.22), but with significant enhancement. The same analysis is done for this code, and the result is shown in Fig.(4.9b). It can be clearly seen that the selection process for the migrated and mutated n SIV is enhanced.

Note that, this part of modification is not available for 1-dimensional problems, and has less effect for 2-dimensional problems.

4.4.1.2 Clear Duplication Process of the Mutation Stage

From the original BBO program (Simon, 2008b), only the worst individuals are mutated. The range of these mutated individuals can be

defined in Algorithm 6 as:

$$R_m = \text{round}\left[\left(\frac{\bar{k}}{d}\right) : (\bar{k})\right] \tag{4.24}$$

where d is a user defined value, and $\bar{k}$ is the total number of islands, see Eq.(4.20). If $d = 2$, then the worst half individuals are to be mutated; and as d increases, the percentage of the total mutated individuals increases too.

The problem happens when the process of the mutation stage is completed, because the next clear duplication stage will cover all the individuals. The process of the last stage will corrupt the non-mutated individuals, and can affect the migration performance during the next generations.

The proposed modification to solve this problem is to perform the clear duplication process only on the mutated individuals, so that the $n\ SIV$ of the non-mutated individuals are kept away from any change.

4.4.2 Performance Comparison Between BBO and BBO-EM

The original and modified versions (*i.e., BBO and BBO-EM*) have been extensively tested through 120 of test functions with different dimensions and complexities, where all the details of each test function are given in Appendix E.

Table (4.7)
The simulation parameters for BBO-EM (for more details refer to (Ma, 2010; Ma & Simon, 2011a))

Parameter	Value
Population size, or k	50
λ_{max}, or I	1
μ_{max}, or E	1
m_{max}	0.01
Elitism	1
Mutation range	round $\left[\left(\frac{k}{2}\right) : \bar{k}\right]$
Number of trials	30

Table (4.8)
The required generation number for various n-dimensional problems

Problem's dimension	No. of generations
1, 2, 3, 4, 5 or 6	1,000
8, 9 or 10	5,000
15, 17 or 20	10,000
30	20,000
60	50,000

Table 4.7 shows the parameters used for both BBO models. These parameters are similar to those used in (Ma, 2010; Ma & Simon, 2011a), but with more restriction on the generations limits, which are listed in Table 4.8.

The performance comparison are evaluated using 120 test functions. This huge number of test functions will cover a gradient of difficulties of unimodal and multimodal functions in conjunction with few and many local minima as traps, where the diversity of variable bounds through narrow and wide search spaces will provide other challenges. These test functions are split into two equally groups; the first group contains only 2-dimensional test functions which are listed in Table 4.9, while the second group contains the other n-dimensional test functions which

are listed in Table 4.10. The reason for this arrangement is because the 2-dimensional are widely available comparing to the other test functions.

Table 4.9 and Table 4.10 give the best, mean and standard deviation of the 120 test functions. As an overall, the obtained results shows that the performance of PMB-BBO can be enhanced effectively by applying the proposed essential modifications on its probabilistic selection process and the range of the clear duplication process.

However, for 1-dimensional problems, this proposed modification is not effective because they have only one SIV, which means that the migration and mutation stages are processed within one independent variable. Thus, the first part of the proposed modification is absent. Based on this study, the new version of PMB-BBO provides better performance if both parts of the essential modifications are embedded. This can be seen from the result of the 1-dimensional problems that are shown in Table 4.10. This issue could be solved if the blinded or/and binary-coded BBO program is used instead of using the real-coded BBO program, although it has not yet been proven. Similarly, for other n-dimensional problems (where $n \geqslant 2$), the modified PMB-BBO shows better performance and wins in most test functions. From the results of the best error, the mean, and the standard deviation, it can be concluded that the proposed correction on the randomized selection process of the migration and mutation stages improves its exploration and exploitation. Fig.(4.10) shows the curves of fitness functions of both versions for the **Generalized Rastrigin's function "f30-05", Schwefel's problem 1.2 "f30-11", Qing's**

Table (4.9)
Comparison of results over 30 trials of the original and essentially modified versions of PMB-BBO – It contains just 2-dimensional test functions

f #	Function Name	n	Biogeography Based Optimization (BBO)					
			Original Partial Migration Based			Modified Partial Migration Based		
			Best	Mean	StdDev	Best	Mean	StdDev
f2-01	Aluffi-Pentini	2	1.4073E-07	4.3166E-05	5.3028E-05	**1.1278E-07**	**8.8268E-06**	**1.0718E-05**
f2-02	Banana Shape	2	**3.2053E-07**	**3.4468E-04**	**5.4340E-04**	2.9272E-06	7.4032E-04	8.3409E-04
f2-03	Beale	2	**2.8187E-06**	**1.6885E-04**	**1.6853E-04**	3.0033E-06	3.5367E-04	5.3948E-04
f2-04	Becker-Lago	2	9.1117E-08	1.4622E-05	1.7747E-05	**1.6084E-08**	**3.5743E-06**	**6.8117E-06**
f2-05	Bird	2	**7.0155E-07**	2.8262E-03	2.8437E-03	1.2209E-05	**3.5197E-04**	**4.4087E-04**
f2-06	Bohachevsky's F1	2	2.6931E-05	1.1511E-03	1.2958E-03	**8.4775E-07**	**2.4441E-04**	**3.2364E-04**
f2-07	Bohachevsky's F2	2	5.2095E-06	1.5371E-03	1.7717E-03	**1.5425E-06**	**1.2837E-04**	**1.6563E-04**
f2-08	Bohachevsky's F3	2	9.8338E-05	**2.5948E-03**	**4.0705E-03**	**1.8607E-05**	3.4955E-03	4.0741E-03
f2-09	Booth	2	2.3450E-05	1.3871E-03	1.3831E-03	**7.0317E-08**	**2.7318E-04**	**3.8789E-04**
f2-10	Branin RCOS	2	**5.6480E-07**	9.6466E-05	1.6743E-04	6.1344E-07	**5.2541E-05**	**1.3086E-04**
f2-11	Bukin's F4	2	**7.9693E-07**	6.6953E-05	6.2600E-05	9.5554E-07	**4.1762E-05**	**5.5448E-05**
f2-12	Bukin's F6	2	1.4263E-01	**5.5443E-01**	**2.3743E-01**	**1.2043E-01**	6.5545E-01	3.2602E-01
f2-13	Carrom Table	2	1.6143E-06	2.6772E-04	3.3997E-04	**4.6210E-08**	**5.5445E-05**	**5.1728E-05**
f2-14	Chichinadze	2	4.2929E-06	8.3095E-03	1.6882E-02	**3.6249E-06**	**4.4121E-03**	**6.0434E-03**
f2-15	Complex	2	**3.3448E-09**	1.1847E-05	1.4765E-05	1.1044E-08	**2.4003E-06**	**3.0318E-06**
f2-16	Cosine Mixture	2	6.6629E-08	4.9456E-06	5.9989E-06	**1.7583E-11**	**1.2499E-06**	**2.0690E-06**
f2-17	Cross In Tray	2	1.6206E-08	3.1863E-06	4.7941E-06	**5.1076E-09**	**5.2181E-07**	**5.9861E-07**
f2-18	Cross Leg Table	2	9.9939E-01	**9.9959E-01**	**8.6652E-05**	**9.9933E-01**	9.9960E-01	9.1172E-05
f2-19	Crowned Cross	2	**1.2851E-01**	2.6119E-01	**3.8380E-02**	1.5692E-01	**2.4251E-01**	4.1245E-02
f2-20	Davis	2	1.4899E-01	4.0370E-01	1.3762E-01	**8.7947E-02**	**2.6525E-01**	**1.1171E-01**
f2-21	Decanomial	2	**2.3936E-06**	**1.2007E-02**	**2.0762E-02**	3.6446E-05	3.0835E-02	8.4008E-02
f2-22	Dekkers-Aarts	2	4.9519E-01	2.6947E+00	3.6355E+00	**4.8362E-01**	**1.8882E+00**	**3.5944E+00**
f2-23	Drop Wave	2	**6.5366E-06**	**3.6741E-03**	**1.1802E-02**	1.7502E-05	7.8462E-03	1.9646E-02
f2-24	Easom	2	3.3330E-06	1.0643E-04	8.4788E-05	**8.7507E-08**	**1.8423E-05**	**3.1394E-05**
f2-25	Egg Holder	2	8.1682E-02	4.4751E+00	**2.5844E+00**	**4.1432E-02**	**3.6928E+00**	5.8969E+00
f2-26	EXP2	2	1.3177E-07	2.0891E-05	**3.0858E-05**	**9.5422E-08**	**4.4703E-06**	**6.1713E-06**
f2-27	Freudenstein-Roth	2	5.4859E-05	1.0514E-02	2.1059E-02	**2.0160E-06**	**4.2978E-03**	**6.6492E-03**
f2-28	Giunta	2	2.4751E-08	5.0673E-07	6.3542E-07	**7.6581E-10**	**8.1474E-08**	**1.1653E-07**
f2-29	Goldstein-Price	2	2.7778E-05	1.5749E-03	1.6278E-03	**3.1758E-06**	**2.7174E-04**	**4.1521E-04**
f2-30	Himmelblau	2	5.9826E-06	9.4920E-04	1.2093E-03	**1.4971E-06**	**7.2058E-05**	**9.5358E-05**
f2-31	Holder Table	2	6.7954E-07	7.5566E-05	1.2147E-04	**6.7237E-07**	**1.6798E-05**	**2.3888E-05**
f2-32	Hosaki	2	2.9862E-08	1.2363E-05	1.3592E-05	**1.9118E-08**	**2.2394E-06**	**2.3016E-06**
f2-33	Kearfott	2	1.4138E-06	1.7270E-05	2.3710E-05	**9.0621E-09**	**3.6849E-06**	**6.8312E-06**
f2-34	Inverted Cosine Wave	2	8.6479E-06	5.0150E-04	5.6270E-04	**2.5292E-06**	**4.7085E-05**	**1.0482E-04**
f2-35	Levy's F3 (or Hansen)	2	3.4917E-04	1.9752E-02	2.4033E-02	**2.1775E-04**	**1.7541E-03**	**2.2954E-03**
f2-36	Levy's F5	2	**5.5062E-05**	1.4691E-01	1.4975E-01	6.4056E-05	**2.2562E-02**	**2.5591E-02**
f2-37	Matyas	2	**2.5988E-07**	6.4073E-05	7.3529E-05	5.2695E-07	**3.8882E-05**	**4.2294E-05**
f2-38	McCormick	2	1.9661E-07	1.9014E-05	2.7088E-05	**9.8424E-08**	**3.2686E-06**	**4.4669E-06**
f2-39	Michalewicz	2	4.4897E-07	1.4864E-05	3.5212E-05	**9.5133E-09**	**3.6163E-06**	**5.3882E-06**
f2-40	Muller-Brown Surface	2	8.2645E-04	2.2639E-02	2.2129E-02	**8.5253E-06**	**8.2214E-03**	**1.0327E-02**
f2-41	Parsopoulos	2	3.3808E-12	4.5059E-08	5.6394E-08	**8.3716E-13**	**2.2079E-08**	**2.7074E-08**
f2-42	Peaks	2	7.0089E-07	2.2400E-04	2.7718E-04	**4.2985E-08**	**3.7351E-05**	**4.3122E-05**
f2-43	Pen Holder	2	8.5556E-09	2.7320E-07	4.6665E-07	**1.9586E-10**	**3.8556E-08**	**3.8049E-08**
f2-44	Powell's Badly Scaled	2	3.3630E-04	**6.4544E-01**	3.7524E-01	**9.7414E-06**	8.9186E-01	**3.1950E-01**
f2-45	Sawtoothxy	2	7.1264E-05	3.2655E-03	6.8834E-03	**1.6672E-06**	**3.1424E-04**	**4.5476E-04**
f2-46	Schaffer's F1	2	9.5860E-04	**8.7166E-03**	2.4978E-03	**1.0451E-04**	9.2323E-03	**1.9361E-03**
f2-47	Schaffer's F2	2	1.4803E+00	6.6148E+00	3.4823E+00	**1.3788E+00**	**3.3010E+00**	**1.2894E+00**
f2-48	Shekel's Foxholes	2	**2.1720E-11**	8.0558E-08	2.1865E-07	6.4642E-11	**6.4142E-10**	**5.9986E-10**
f2-49	Sinusoidal Problem	2	2.2129E-07	4.4691E-05	5.4481E-05	**2.6980E-08**	**3.1974E-06**	**4.2395E-06**
f2-50	Stenger	2	1.0055E-06	1.7379E-04	2.2777E-04	**5.6936E-07**	**6.4535E-05**	**9.4907E-05**
f2-51	Storn	2	**3.9437E-07**	1.3249E-06	1.3478E-06	3.9493E-07	**1.1236E-06**	**1.2236E-06**
f2-52	Stretched V	2	**2.9346E-26**	**1.4161E-16**	**4.4560E-16**	5.9429E-24	4.1241E-16	1.3925E-15
f2-53	Test Tube Holder	2	1.0791E-06	8.0866E-05	9.8752E-05	**7.8935E-08**	**2.8773E-05**	**4.7662E-05**
f2-54	Treccani	2	9.4387E-08	2.0880E-05	2.7157E-05	**3.8026E-08**	**2.2693E-06**	**1.8565E-06**
f2-55	Trefethen's F4	2	**4.6587E-04**	**9.5690E-02**	**8.1159E-02**	9.1985E-04	1.3107E-01	1.1657E-01
f2-56	Tripod	2	5.2656E-03	1.0925E-01	**6.3076E-02**	**3.5156E-03**	**5.3680E-02**	9.0101E-02
f2-57	Zakharov	2	**1.4286E-07**	6.8890E-05	6.9831E-05	6.0584E-07	**6.6225E-06**	**6.2012E-06**
f2-58	Zettl	2	2.6937E-08	8.7643E-06	1.0503E-05	**9.2581E-09**	**1.9735E-06**	**4.2604E-06**
f2-59	3-Hump Camel-Back	2	**5.7882E-08**	2.8687E-05	5.8124E-05	9.8715E-08	**5.3205E-06**	**7.9850E-06**
f2-60	6-Hump Camel-Back	2	2.3455E-07	9.1217E-05	1.2481E-04	**3.4081E-08**	**1.5711E-05**	**2.2781E-05**

Table (4.10)
Comparison of results over 30 trials of the original and essentially modified versions of PMB-BBO – (where $n \neq 2$)

f #	Function Name	n	Biogeography Based Optimization (BBO)					
			Original Partial Migration Based			Modified Partial Migration Based		
			Best	Mean	StdDev	Best	Mean	StdDev
f1-01	Mineshaft F1	1	6.2998E-01	7.5377E-01	**7.4896E-02**	**4.4885E-01**	**7.4142E-01**	1.0203E-01
f1-02	Mineshaft F2	1	**2.8719E-09**	4.6768E-04	**6.6885E-04**	3.9974E-08	**3.9916E-04**	8.2865E-04
f1-03	Shekel's F1	1	9.8030E-08	**2.8723E-05**	**4.2939E-05**	**6.7958E-09**	1.2670E-04	4.0902E-04
f1-04	Shekel's F2	1	**8.1958E-10**	**2.1546E-05**	**5.2185E-05**	8.7502E-09	5.4862E-05	8.6120E-05
f1-05	Shekel's F3	1	1.7454E-08	7.4624E-05	1.8726E-04	**1.4736E-10**	**5.1717E-05**	**8.7380E-05**
f1-06	Stron.-Zilin.-Shalt.	1	**5.7915E-11**	**1.0865E-07**	**2.4327E-07**	1.0768E-10	2.5748E-07	6.8094E-07
f1-07	Suharev	1	1.5944E-11	**9.3043E-08**	**1.2955E-07**	**1.1613E-11**	1.8424E-07	4.0888E-07
f1-08	Zilinskas' F2	1	1.4002E-09	**2.2730E-06**	**5.2141E-06**	**1.0064E-09**	9.9106E-06	2.2533E-05
f3-01	Box-Betts	3	2.3366E-08	1.4450E-06	**1.1712E-06**	**2.3614E-09**	**1.0091E-06**	1.2014E-06
f3-02	Hartman's F1	3	7.6177E-06	5.0374E-04	3.8211E-04	**1.0262E-06**	**3.8549E-05**	**7.4211E-05**
f3-03	Helical Valley	3	8.1299E-02	8.4147E-01	**6.2500E-01**	**2.3847E-03**	**5.2274E-01**	6.8708E-01
f3-04	Levy's F8	3	6.4096E-06	1.2455E-03	1.2789E-03	**4.8947E-07**	**5.2271E-05**	**6.1623E-05**
f3-05	Meyer and Roth	3	9.1336E-06	1.0188E-04	**8.3202E-05**	**4.5474E-06**	**7.0623E-05**	9.3791E-05
f3-06	Perm F1	3	1.6149E-01	1.2030E+00	7.6733E-01	**4.0283E-03**	**5.7516E-01**	**4.9994E-01**
f4-01	Corana (or Ingber)	4	3.5444E+00	8.6925E+01	6.1928E+01	**0.0000E+00**	**1.1788E+01**	**1.2559E+01**
f4-02	Kowalik	4	3.9927E-04	8.0573E-04	3.2529E-04	**2.5735E-04**	**6.2274E-04**	**3.0950E-04**
f4-03	Miele and Cantrell	4	8.0518E-09	**1.2520E-06**	**1.2340E-06**	**1.2172E-10**	1.4661E-06	3.9810E-06
f4-04	Powell's Quartic	4	2.8518E-02	1.8301E+00	2.0186E+00	**5.9062E-03**	**2.6144E-01**	**3.1940E-01**
f4-05	Neumaier's F2	4	9.7017E-03	3.9273E-02	2.4992E-02	**6.8860E-03**	**2.8289E-02**	**2.3559E-02**
f4-06	Wood (or Colville)	4	1.5438E+00	7.1389E+00	3.4273E+00	**1.2867E-01**	**1.9675E+00**	**1.3293E+00**
f5-01	AMGM	5	9.8524E-10	1.4399E-07	2.2968E-07	**3.1111E-11**	**4.1812E-09**	**8.7922E-09**
f5-02	Osborne's F1	5	1.1839E-02	**1.1988E-01**	**9.1466E-02**	**1.1325E-02**	1.3801E-01	1.0496E-01
f5-03	SODP	5	7.2433E-07	6.9565E-05	8.3596E-05	**6.3699E-09**	**1.6674E-05**	**2.4309E-05**
f5-04	Styblinski-Tang	5	8.1139E-02	8.6368E-01	8.0724E-01	**3.4485E-03**	**6.1622E-02**	**5.0114E-02**
f6-01	Hartman's F2	6	1.8781E-03	6.9158E-02	6.0706E-02	**7.1304E-04**	**3.8071E-02**	**5.4616E-02**
f6-02	Perm F2	6	2.9814E-01	1.0219E+00	6.1973E-01	**1.8974E-02**	**5.7573E-01**	**6.0020E-01**
f9-01	ANNs XOR	9	**7.5664E-04**	**5.7103E-03**	3.6158E-03	7.9878E-04	6.8751E-03	**2.2983E-03**
f9-02	Price's Transistor	9	2.5525E+01	1.3334E+02	5.4565E+01	**9.1948E+00**	**9.0453E+01**	**4.1274E+01**
f9-03	Storn's Chebyshev	9	9.7939E+03	5.4462E+04	3.6762E+04	**3.5246E+03**	**2.6181E+04**	**2.2022E+04**
f10-01	Epistatic Michalewicz	10	4.2122E-01	1.2966E+00	6.2987E-01	**1.5307E-01**	**6.7944E-01**	**3.4882E-01**
f10-02	Katsuura	10	2.1784E-01	5.0836E-01	1.4062E-01	**8.0454E-02**	**2.9182E-01**	**1.1279E-01**
f10-03	Odd Square	10	**9.0095E-01**	**1.0167E+00**	4.3942E-02	9.6634E-01	1.0366E+00	**3.0877E-02**
f10-04	Paviani	10	6.2946E-03	2.0765E-02	1.3830E-02	**1.3197E-03**	**5.5884E-03**	**4.0332E-03**
f15-01	Dixon-Price	15	**5.6066E-01**	2.1042E+00	1.3961E+00	7.3983E-01	**1.4613E+00**	**5.4173E-01**
f15-02	Neumaier's F3 (or Trid)	15	1.0026E+02	9.5503E+02	**6.7827E+02**	**8.9477E+01**	**8.7253E+02**	7.7831E+02
f15-03	Normalized Rana	15	**2.9324E+01**	5.6653E+01	1.2546E+01	2.9923E+01	**4.6883E+01**	**9.8307E+00**
f17-01	Bent Cigar	17	9.2454E+05	2.6185E+06	1.2377E+06	**5.3826E+05**	**1.3379E+06**	**5.7265E+05**
f17-02	Defl. Corrug. Spring	17	6.2664E-01	1.3368E+00	5.0800E-01	**6.2664E-01**	**9.9217E-01**	**3.9077E-01**
f17-03	Infinity (or Csendes)	17	4.6802E-14	1.6281E-11	2.4605E-11	**4.1422E-15**	**2.4643E-12**	**4.6031E-12**
f20-01	Alpine	20	1.7629E-02	3.6322E-02	1.2919E-02	**1.5741E-02**	**2.8971E-02**	**1.0286E-02**
f20-02	Quintic	20	2.0141E+00	3.2993E+00	7.3579E-01	**1.4342E+00**	**2.6384E+00**	**5.6576E-01**
f20-03	Pathological	20	**1.7044E+00**	2.6537E+00	4.2252E-01	2.1295E+00	**2.5968E+00**	**3.8864E-01**
f30-01	Ackley	30	6.3026E-01	9.9236E-01	2.3065E-01	**5.6622E-01**	**9.3734E-01**	**2.1466E-01**
f30-02	Gen. Griewank	30	8.6708E-01	1.0263E+00	**3.4402E-02**	**8.4581E-01**	**1.0131E+00**	4.7757E-02
f30-03	Gen. Penalized F1	30	5.0934E-03	2.9591E-02	3.2559E-02	**1.6288E-03**	**1.7078E-02**	**1.6484E-02**
f30-04	Gen. Penalized F2	30	9.3001E-02	**1.6875E-01**	**6.0757E-02**	**8.2945E-02**	1.8237E-01	7.5631E-02
f30-05	Gen. Rastrigin	30	9.4594E-01	1.9351E+00	**6.4580E-01**	**7.5521E-01**	**1.7540E+00**	7.1453E-01
f30-06	Gen. Rosenbrock	30	1.0517E+02	2.6058E+02	**7.9555E+01**	**8.5320E+01**	**2.4504E+02**	9.3473E+01
f30-07	Gen. Schwefel's F2.26	30	**4.4291E-06**	1.3876E-05	5.7466E-06	4.6460E-06	**1.0766E-05**	**3.8181E-06**
f30-08	Mishra's F1	30	1.3229E-01	1.8511E-01	3.4109E-02	**1.1668E-01**	**1.6764E-01**	**2.7709E-02**
f30-09	Mishra's F2	30	1.2195E-01	1.9327E-01	3.4082E-02	**1.1973E-01**	**1.6742E-01**	**3.0336E-02**
f30-10	Quartic	30	2.8562E-07	1.2406E-06	1.2872E-06	**4.9473E-08**	**9.6111E-07**	**1.1137E-06**
f30-11	Schwefel's F1.2	30	4.2433E+02	1.9527E+04	1.1268E+04	**1.0214E+00**	**2.6767E+02**	**2.9316E+02**
f30-12	Schwefel's F2.21	30	**3.2007E+00**	6.2387E+00	1.1469E+00	4.3096E+00	**5.5198E+00**	**9.0466E-01**
f30-13	Schwefel's F2.22	30	**4.0024E-01**	**7.0184E-01**	1.3458E-01	4.5480E-01	7.0286E-01	**1.3019E-01**
f30-14	Sphere Model	30	1.8518E+00	3.8843E+00	1.3964E+00	**1.5508E+00**	**3.3657E+00**	**1.2515E+00**
f30-15	Step	30	2.0000E+00	4.7333E+00	**1.8245E+00**	**0.0000E+00**	**4.1000E+00**	1.8859E+00
f60-01	Hyper-Ellipsoid	60	5.2404E-01	1.0131E+00	3.8482E-01	**5.1024E-01**	**9.0328E-01**	**2.6241E-01**
f60-02	Qing	60	3.5937E+03	**5.9785E+03**	**1.6586E+03**	**3.5094E+03**	6.4488E+03	2.0255E+03
f60-03	Salomon	60	2.2999E+00	3.0172E+00	3.5403E-01	**2.2999E+00**	**2.7949E+00**	**2.5704E-01**

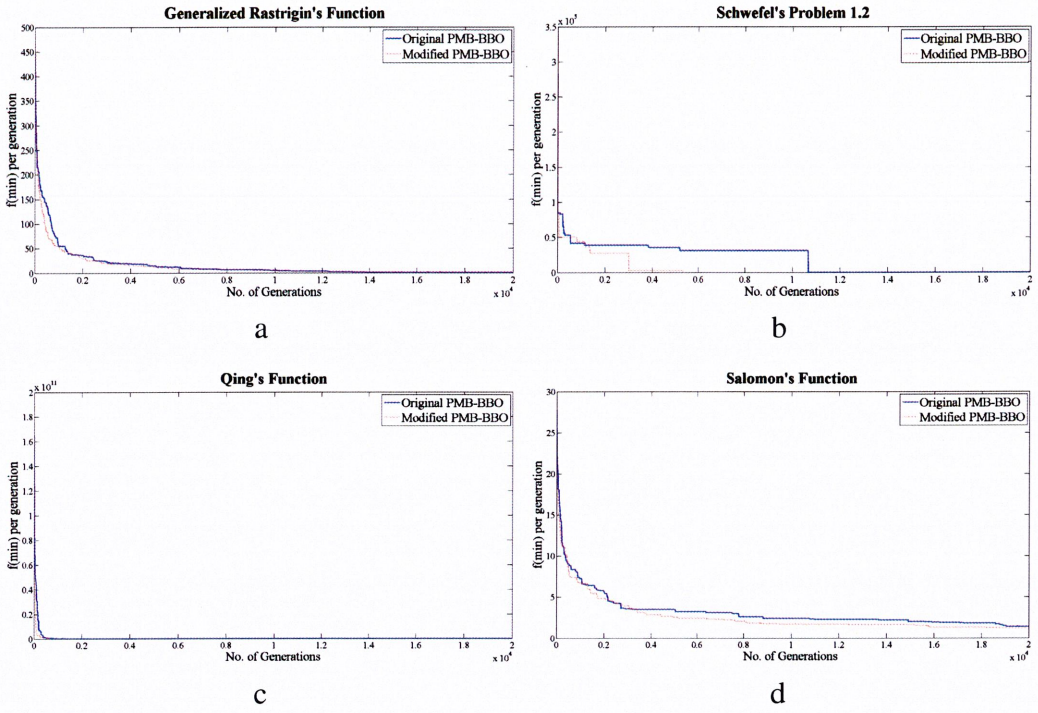


Figure (4.10)
Curves of fitness functions of BBO and BBO-EM for some selected functions. (a) $f30$-05, (b) $f30$-11, (c) $f60$-02 and (d) $f60$-03

function "f60-02"** and **Salomon's function "f60-03". It can be clearly seen that BBO-EM can converge to better solutions as compared to that of BBO.

Therefore, the results have proved that these essential modifications are highly recommended for problems with dimension higher than 1. Even, for the few test functions, where the original version shows better results, the modified version gives a competitive results.

Although, BBO-EM shows enhanced results, for some very hard test functions, like **Price's Transistor**, **Storn's Chebyshev function**, **Trid's** (or **Neumaier's problem 3**), **Normalized Rana's function with diagonal wrap**, **Bent Cigar function**, **Qing's function**, **Generalized Rosenbrock's function** and **Schwefel's problem 1.2**, both versions failed to

Table (4.11)
The overall performance of BBO and BBO-EM with/without mutation stage

f#	Function Name	*n*	Mutation Stage	Biogeography Based Optimization (BBO)					
				Partial Migration Based			Modified Partial Migration Based		
				Best	Mean	StdDev	Best	Mean	StdDev
f30-06	Generalized Rosenbrock	30	Activated	1.0517E+02	2.6058E+02	**7.9555E+01**	**8.5320E+01**	**2.4504E+02**	9.3473E+01
			Not-Activated	1.7036E+02	6.0492E+02	6.4746E+02	**1.6188E+02**	**2.5669E+02**	**7.3475E+01**
f30-07	Generalized Schwefel F2.26	30	Activated	**4.4291E-06**	1.3876E-05	5.7468E-06	4.6460E-06	**1.0766E-05**	**3.8181E-06**
			Not-Activated	8.1593E-06	2.9257E-05	1.1255E-05	**4.4413E-06**	**1.1069E-05**	**4.1198E-06**
f2-29	Goldstein-Price	2	Activated	2.7778E-05	1.5749E-03	1.6278E-03	**3.1758E-06**	**2.7174E-04**	**4.1521E-04**
			Not-Activated	1.4627E-04	2.5692E-02	2.2876E-02	**4.3193E-06**	**1.0116E-03**	**1.1351E-03**
f2-48	Shekel's Foxholes	2	Activated	**2.1720E-11**	8.0558E-08	2.1865E-07	6.4642E-11	**6.4142E-10**	**5.9986E-10**
			Not-Activated	5.4152E-08	2.2467E-03	8.6667E-03	**5.2873E-11**	**3.9904E-08**	**1.3873E-07**

Table (4.12)
Normalized CPU time of the 60-dimensional problems $\{f60\text{-}01$ to $f60\text{-}03\}$

Function	PMB-BBO Versions		CPU Time Saving (%)
	Original ver.	Modified ver.	
Hyper-Ellipsoid	1.4617E+00	**1.0000E+00**	**31.58437697**
Qing	1.4605E+00	**1.0000E+00**	**31.53056390**
Salomon	1.5104E+00	**1.0000E+00**	**33.79377207**
Avg. CPU Time	1.4775E+00	**1.0000E+00**	**32.31951040**

converge to the optimal solutions. However, as a comparison, the proposed version outperform the original version on most of these test functions.

Table 4.11 shows an extended comparison when the mutation stage is not considered, as it has been done in the preceding study between PMB-BBO and SPMB-BBO. Here, the performance of BBO-EM is clearly better than the original PMB-BBO.

Table 4.12 shows the CPU time comparison between the original PMB-BBO and the modified PMB-BBO. It can be clearly seen that the modified version can save around 32.32% of the CPU time, which means that it is even faster than the simplified partial migration model "SPMB-BBO" which was presented in the preceding Section by around 24.76%.

4.4.2.1 Discussion

The comparison between the performance of the four original models of BBO (PMB, SMB, SPMB and SSMB) reveals that PMB-BBO gives the best performance when the given problem is hard, has large upper and lower limits of search space, high-dimensional and/or the number of islands (individuals or population size) is small. However, PMB-BBO lacks the exploration (Gong, Cai, & Ling, 2010). Therefore, in this study, the root problem that causes the poor exploration is solved by using an integer random function which provides a pseudorandom integers from a uniform discrete distribution. Furthermore, the exploitation is improved by keeping the non-mutated solutions away from any corruption that can be caused by the clear duplication process.

4.4.3 Conclusions and Suggestions

This study proposed some modifications to improve the performance of the original form of PMB-BBO without using any complicated models for immigration and emigration rates. It is shown that the simplified linear model still can give good results if the root problems of the migration and mutation stages are solved. An extensive testing of the original and the proposed modified versions of PMB-BBO through 120 test functions shows that the performance of the modified version of PMB-BBO (*which is called BBO-EM*) is better on Best, Mean, and StdDev than that of the original version.

The proposed modification can be used as a basis for modifying the existing modified BBOs in the literature. For example, the blended-BBO which is presented in (Ma & Simon, 2011b) can be a great add-on to BBO-EM. Furthermore, even if it is required to employ the complicated models of the immigration and emigration rates, such as the generalized sinusoidal migration model in (Ma & Simon, 2011a); this complicated model shows a great performance, and it is very interesting to re-test this model with considering the proposed essential modifications that are described in this study.

4.5 Iterative Variable Bounds – Biogeography Based Optimization (IVB-BBO)

This Section proposes a new modification with three suggested models that can be involved in any existing model of BBO in the literature. This modification is based on one of the main features of BBO where the other best solutions are not erased. This feature will be utilized effectively in order to iterate the lower and upper bounds of each independent variable, where the global optimum solution is supposed to be located in between; such that the minimization of search space gives a richness of good solutions. All of these three models will be tested under various popular benchmark functions, and the obtained results will be compared with that of the other BBO models in the literature.

As covered before, the BBO performance can be enhanced either by modifying the BBO algorithm itself or by hybridizing it with other

algorithms, or even both; as it will be described in details later. Unfortunately, the existing modified BBOs are constructed with complicated models which are not easy to be implemented, and need more effort from the programmers. In contrast, the simplified linear model is easy to be programmed, but it gives poor performance.

From the previous Section, the root problems that are associated with the simplified linear model are solved and the results are improved markedly. But, it can still provide more accurate results if the strength side of its algorithm is utilized effectively, which will be the focus of this Section. A superior BBO algorithm will be presented at the end of this Section.

4.5.1 The Proposed Modifications on BBO

It can be clearly seen from the preceding literature review that all the modifications either complicate the BBO algorithm and/or consume extra CPU time as a result of dealing with the hybridization or the parallel computing. Instead, the proposed modifications will solve the weakness of the original BBO algorithm by just using the simplified linear model shown in Fig.(4.4); and at the same time, the algorithm processing speed becomes very fast.

It is important to remember that the model of PMB-BBO is selected to be the best original variation in comparison with SMB-BBO, SPMB-BBO and SSMB-BBO, as covered in Section 4.3. For simplicity, PMB-BBO will be referred to as BBO, here after. In addition, the essential

modifications in Section 4.4 are considered as a foundation for this new algorithm. Moreover, based on the performance that are obtained from RCBBO in (Gong, Cai, Ling, & Li, 2010) and EBBO in (Pattnaik et al., 2010), the proposed modifications will be based on the floating point with clear duplicated operator.

The following proposed modifications will be a practical involvement of one major feature available in BBO that the other best solutions will not be erased during the next new generations (Simon, 2008a).

The major novelty of this Section is to utilize that feature of BBO by iterating the bounds of the independent variables (i.e., SIV_{min} and SIV_{max}) without drifting away from the location of the optima.

The vector of the migrated best solutions are used to extract the updated search space, in which the optimal solution is located. The new obtained range among the lower and upper bounds of each SIV is smaller than that initially available in the given problem. Thus, more bounded side constraints become very rich and contain lots of good solutions, where the probability for finding the optimal solution is higher than that obtained through the original BBO.

Two different models are employed to do the previous task. From the obtained results, the advantages are collected, and the disadvantages are rejected. The third model is built based on the previous two models, and considered as the final algorithm. These three models are respectively called **Bounds Single Updater-BBO (BSU-BBO)**, **Bounds**

Multi Updater-BBO (BMU-BBO) and **Iterative Search Space-BBO (ISS-BBO)**.

4.5.1.1 Bounds Single Updater-BBO (BSU-BBO)

This is the basic model. It will iterate the search space just one time based on the collected best solutions after certain amount of generations. The total number of generations is divided into two stages, $G_T = [G_1, G_2]$, where G_1 is larger than G_2 ($G_1 >> G_2$) to increase the probability to cover all the local optimum points. In this study, these two stages are respectively taken as $G_1 = 0.75G_T$ and $G_2 = 0.25G_T$, or in other word $G_1 = 3G_2$.

Referring to Eq.(4.20) and the general flowchart of Fig.(4.7), there are ($\bar{k} \times G_1$) **number of function evaluations (NFE)** for the first stage. The vector of the best solutions can be obtained from:

$$ISI_{min,g} = \min\left(ISI_{1,g}, ISI_{2,g}, \cdots, ISI_{\bar{k},g}\right) \ , \ g = 1, 2, \cdots, G_1 \quad (4.25)$$

Because there are n SIV for each individual, then the vector of SIV_s can be obtain from the population of Eq.(4.25) as:

$$SIV_s = [SIV_{s,1}, SIV_{s,2}, \cdots, SIV_{s,G_1}] \ , \ s = 1, 2, \cdots, n \quad (4.26)$$

There are two remaining steps before be able to iterate $SIV_{min,s}$ and $SIV_{max,s}$. The n vectors of Eq.(4.26) have to be sorted in an ascending form, which can be expressed as:

$$SIV_s = \text{sort}\left(SIV_s\right) \quad (4.27)$$

Algorithm 8 The pseudocode of the BSU-BBO model

Require: Start BBO normally for the first stage G_1

for $g \leftarrow 1$ *to* G_1 **do**
  Apply Eq.(4.25)
end for
Apply Eq.(4.26), and then Eq.(4.27)
if $SIV_{min,s}^{old} \leqslant SIV_{s,1} \leqslant SIV_{max,s}^{old}$ **then**
  Apply Eq.(4.28)
end if
if $SIV_{min,s}^{old} \leqslant SIV_{s,G_1} \leqslant SIV_{max,s}^{old}$ **then**
  if $SIV_{min,s}^{new} < Eq.(4.29)$ **then**
    Apply Eq.(4.29)
  end if
else
  $SIV_{max,s}^{new} = SIV_{max,s}^{old}$
end if
Start BBO normally for the second stage G_2

Then, the new bounds can be updated as follows:

$$SIV_{min,s}^{new} = SIV_{min,s}^{old} + SF\left(SIV_{s,1} - SIV_{min,s}^{old}\right) \quad (4.28)$$

$$SIV_{max,s}^{new} = SIV_{max,s}^{old} - SF\left(SIV_{max,s}^{old} - SIV_{s,G_1}\right) \quad (4.29)$$

The **Safety Factor (SF)** is used to avoid the drift of the new search space away from the optimum point. If $SF = 0$, then there is no updating process, which means that BSU-BBO will work like BBO. This study takes $SF = 0.90$ to 0.98, and the model can be described through Algorithm 8.

4.5.1.2 Bounds Multi Updater-BBO (BMU-BBO)

The obtained data from the preceding model proves that the BBO performance can be improved significantly even with only one iteration

on its search space. In addition, Eq.(4.28) and Eq.(4.29) give a reasonable cropping, where the new lower bound is increased and the new upper bound is decreased, without losing the optimal location. However, the new updated search space is still relatively large.

BMU model is designed to collect the data during the first stage G_1. It works in a similar manner of BSU model, but when the second stage G_2 is initiated, then the search space is repeatedly iterated at each new generation.

To accomplish this mission, one vital problem has to be solved first. The SIV_s vectors length is always equal to one, since the iteration is done per each single generation. In this case, $g = 1$ in Eq.(4.25). Thus:

$$ISI_{min,1} = \min\left(ISI_{1,1}, ISI_{2,1}, \cdots, ISI_{\bar{k},1}\right) \tag{4.30}$$

Similarly for Eq.(4.26):

$$SIV_s = [SIV_{s,1}] = SIV_{s,1} \quad , \quad s = 1, 2, \cdots, n \tag{4.31}$$

For Eq.(4.28) and Eq.(4.29), if $SF = 1.0$ then $SIV^{new}_{min,s} = SIV^{new}_{max,s}$, and thus:

$$\Delta SIV^{new}_{s} = SIV^{new}_{max,s} - SIV^{new}_{min,s} = 0 \tag{4.32}$$

This indicates that the second stage becomes useless, as the obtained

solution at its first generation (*i.e., the beginning of* G_2) is always considered as a fitness.

Now, if $SF = 0.98$, as in BSU model, then:

$$\Delta SIV_s^{new} = 0.02\Delta SIV_s^{old} \tag{4.33}$$

It can be clearly seen that Eq.(4.33) quickly converges to zero, and thus Eq.(4.33) approaches Eq.(4.32). *The proof is given in Appendix B.*

To solve this problem, the $\bar{k}$-islands per each generation are considered as a vector of ISI, so Eq.(4.30) is replaced by:

$$ISI_{min} = [ISI_1, ISI_2, \cdots, ISI_{\bar{k}}] \tag{4.34}$$

After completing G_1, the search space is supposed to be very rich of good solutions, and hence Eq.(4.34) can be used. Similarly, Eq.(4.26) is reformulated as:

$$SIV_s = \left[SIV_{s,1}, SIV_{s,2}, \cdots, SIV_{s,\bar{k}}\right] \quad , \quad s = 1, 2, \cdots, n \tag{4.35}$$

The next step is to sort SIV_s in Eq.(4.35) by using Eq.(4.27). The new lower bound can be updated by Eq.(4.28), where Eq.(4.29) becomes:

$$SIV_{max,s}^{new} = SIV_{max,s}^{old} - SF\left(SIV_{max,s}^{old} - SIV_{s,\bar{k}}\right) \tag{4.36}$$

Algorithm 9 The pseudocode of the BMU-BBO model

Require: Start BBO normally for the first stage G_1

1: **for** $g \leftarrow 1$ *to* G_1 **do** {bounds are iterated one time at the end of this loop}
2: Apply Eq.(4.25)
3: **end for**
4: Apply Eq.(4.26), and then Eq.(4.27)
5: **if** $SIV^{old}_{min,s} \leqslant SIV_{s,1} \leqslant SIV^{old}_{max,s}$ **then**
6: Apply Eq.(4.28)
7: **end if**
8: **if** $SIV^{old}_{min,s} \leqslant SIV_{s,G_1} \leqslant SIV^{old}_{max,s}$ **then**
9: **if** $SIV^{new}_{min,s} < Eq.(4.29)$ **then**
10: Apply Eq.(4.29)
11: **end if**
12: **else**
13: $SIV^{new}_{max,s} = SIV^{old}_{max,s}$
14: **end if**
15: **for** $g \leftarrow 1$ *to* G_2 **do** {bounds are iterated continuously at each g}
16: Start BBO normally for the second stage G_2
17: Apply Eq.(4.34), Eq.(4.35), Eq.(4.27), Eq.(4.28) and Eq.(4.36)
18: **end for**

Algorithm 9 summarizes all the previous steps.

4.5.1.3 Iterative Search Space-BBO (ISS-BBO)

As described before, the main advantage of BSU model is that it has higher immunity against converging to non-global optimum as compared to BMU model, whereas its main disadvantage is the restriction on the search space updating process to be just one iteration, which is not enough.

The BMU model can solve that by using multi-iterations process, but the main disadvantage is the high probability of converging to a non-

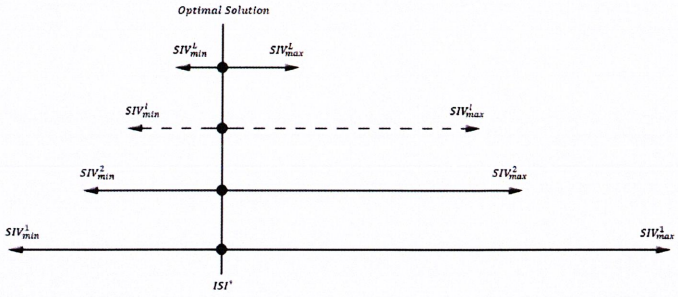
optimal solution. This may happen because the second stage is designed based on an assumption that the best global solution is still within the updated lower and upper bounds per each single generation. Sometimes, this guess does not work, so it converges to near-optimal points with some error. This error can be minimized effectively by increasing $\bar{k}$ and/or G_T. These observations led to merge the properties of both BSU and BMU in just one model called **Iterative Search Space (ISS)**.

The idea behind ISS is to let the BBO algorithm to run in multi-stages or levels, as shown in Fig.(4.11), where each stage is assigned with certain number of generations as expressed by Eq.(4.37). This will help BBO to collect enough data for cropping the search space at the end of each stage. Here, the vector length of SIV_s is equal to the number of generations of stage l, which represents the strength of the BSU model. Since the ISS model has multi-stages rather than just G_1 and G_2 stages, it can do multi-iterations process, which is the strength of the BMU model. The weakness of BMU model is due to the presence of the iteration in each generation. Instead, ISS model iterates the side constraints at the end of each stage that contains number of generations.

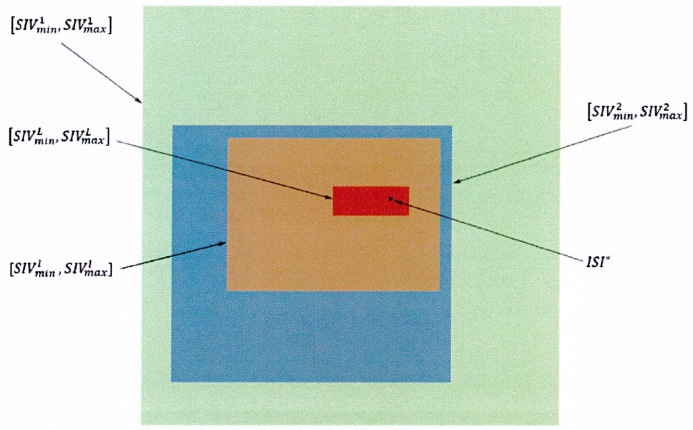
Many functions can be used to define the stages' generations G_l, such as a linear ramp function with a slop u and levels L as in Eq.(4.38), a rational function as in Eq.(4.39), and an exponential function as in Eq.(4.40):

$$G_T = \text{sum}(G_l) = \text{sum}\left([G_1, G_2, \cdots, G_L]\right) \tag{4.37}$$

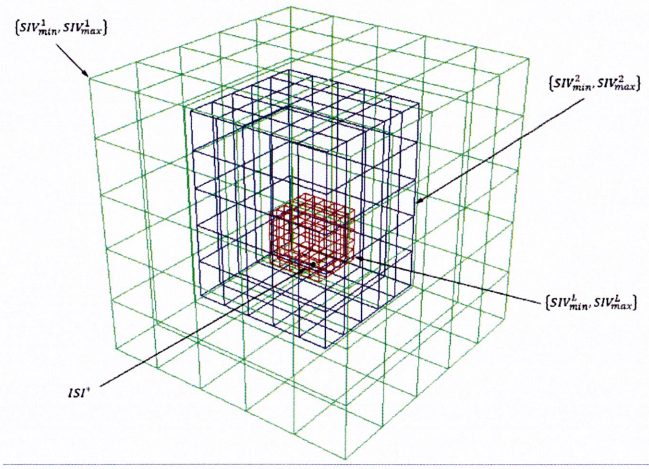
$$G_l = \text{ceil}\left[G_1 - u(l-1)\right] \quad , \quad l = 1, 2, \cdots, L \tag{4.38}$$



a: Search space of 1-dimensional problem



b: Search space of 2-dimensional problem



c: Search space of 3-dimensional problem

Figure (4.11)
Illustrations in the iterated search space and the location of the optimal solution. From (a) to (c): 1, 2 and 3-dimensional problems

$$G_l = \text{ceil}\left[\frac{G_1}{l}\right] \quad , \quad l = 1, 2, \cdots, L \tag{4.39}$$

$$G_l = \text{ceil}\left[G_1 e^{\beta(l-1)}\right] \quad , \quad l = 1, 2, \cdots, L \tag{4.40}$$

There are two main disadvantages that can be highlighted when examining Eq.(4.38). Firstly, the negative slop needs to be tuned carefully. Secondly, since the total generations G_T is distributed in a linear shape, so a small portion of G_T will be assigned for the initial stage G_1, which is not enough. As a result, the error will increase.

The rational function of Eq.(4.39) is similar to a harmonic series (Arfken, Weber, & Harris, 2012):

$$G_T = \text{ceil}\left[G_1 \sum_{l=1}^{L}\left(\frac{1}{l}\right)\right] = \text{ceil}\left[G_1\left(1 + \frac{1}{2} + \cdots + \frac{1}{L}\right)\right] \tag{4.41}$$

The term $\left(\frac{1}{L}\right)$ approaches zero for large L, i.e., $\lim_{L\to\infty}\left(\frac{1}{L}\right) = 0$. The drawback of this function is the repeatability of G_l for some levels, which means it does not distribute G_T in an efficient way.

All the above problems can be solved easily by employing the exponential function of Eq.(4.40). It has a pretty nice distribution shape for G_T, where more generations are assigned at the beginning in order to make sure that the new search space contains the location of the optimum solution, and then the weight is reduced in a decaying form, as shown in

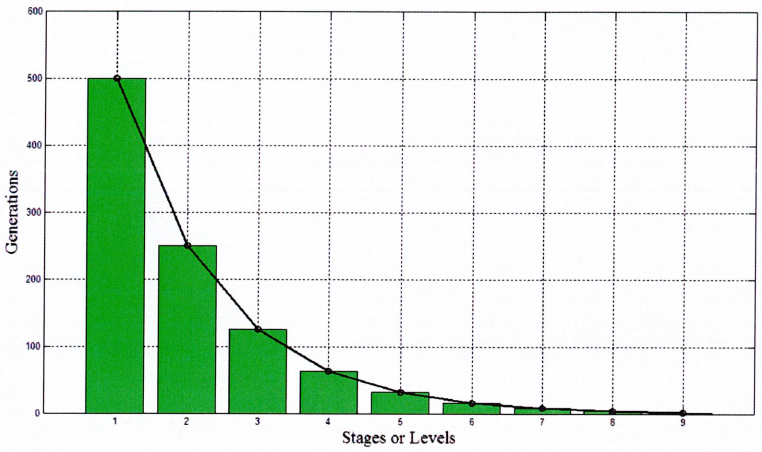


Figure (4.12)
Distribution of generation levels in a decaying form ($G_T = 1000$, $R = 50\%$)

Fig.(4.12). The decay constant β can be calculated as:

$$\beta = -\ln(1-R) \quad \text{where} \quad R = \frac{G_1}{G_T} \tag{4.42}$$

where R is the ratio between the generations at the first stage and the total generations of all the stages. *The proof is given in Appendix C.*

As mentioned before, the number of generations that are assigned for the first stage G_1 should be large enough to prevent, or at least minimize, the drifting error. Thus, it is preferable to set the ratio R at a high value. In this thesis, R is set at 50%, which means that the first half of G_T is utilized just during the first stage, while the second half of G_T is distributed exponentially for the other stages, i.e., $[G_2, G_3, \cdots, G_L]$. Therefore, Eq.(4.25), Eq.(4.26) and Eq.(4.29) will become respectively as:

$$ISI_{min,g_l} = \min\left(ISI_{1,g_l}, ISI_{2,g_l}, \cdots, ISI_{\bar{k},g_l}\right) \;,\; g_l = 1, 2, \cdots, G_l \tag{4.43}$$

$$SIV_s = \left[SIV_{s,1}, SIV_{s,2}, \cdots, SIV_{s,G_l}\right] \;,\; s = 1, 2, \cdots, n \tag{4.44}$$

$$SIV^{new}_{max,s} = SIV^{old}_{max,s} - SF\left(SIV^{old}_{max,s} - SIV_{s,G_l}\right) \tag{4.45}$$

Algorithm 10 The pseudocode of the ISS-BBO model

for $l \leftarrow 1$ *to* L **do**
  Apply Eq.(4.40) for computing G_l
  for $g \leftarrow 1$ *to* G_l **do** {bounds are iterated once time at the end of this loop}
    Start BBO normally for stage G_l
    Apply Eq.(4.43)
  end for
  if $G_l \geqslant$ MGN **then** {MGN: Minimum Generation Number}
    Apply Eq.(4.44), and then Eq.(4.27)
    if $SIV^l_{min,s} \leqslant SIV_{s,1} \leqslant SIV^l_{max,s}$ **then**
      Apply Eq.(4.28) for $SIV^{l+1}_{min,s}$
    end if
    if $SIV^l_{min,s} \leqslant SIV_{s,G_l} \leqslant SIV^l_{max,s}$ **then**
      if $SIV^{l+1}_{min,s} < Eq.(4.45)$ **then**
        Apply Eq.(4.45) for $SIV^{l+1}_{max,s}$
      end if
    else
      $SIV^{l+1}_{max,s} = SIV^l_{max,s}$
    end if
  end if
end for

The ISS model is described in Algorithm 10, and its flowchart is shown in Fig.(4.13).

Note that, from Algorithm 10, MGN is very important factor that can enhance the performance and increase the algorithm's immunity to avoid trapping into local optima. The concept is very simple; once $G_l < MGN$, the iteration process will stop. In this study, MGN is taken as 10, so the iteration process will stop once the new level l contains less than 10 generations.

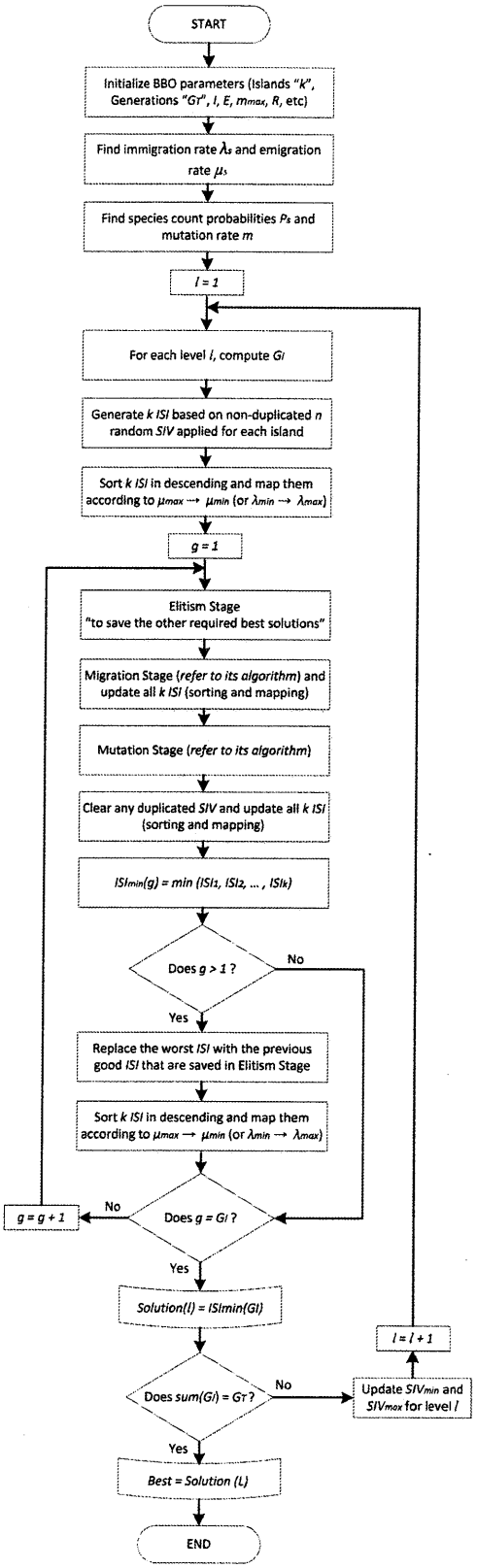


Figure (4.13)
Flowchart of the ISS-BBO algorithm

4.5.2 Comparison Between ISS-BBO and Other Models

The three proposed models (BSU-BBO, BMU-BBO and ISS-BBO) have been extensively tested using 36 benchmark functions. *The details are given in Appendix E.*

The parameters that have been used here are similar to those used in (Ma, 2010; Ma & Simon, 2011a): population size of 50, $I = E = 1$, $m_{max} = 0.01$, generations limit of 1000 for ($n = 1, 2, 3, 4$ and 6), 5000 for ($n = 9$ and 10), 10,000 for ($n = 15, 17$ and 20), and 20,000 for ($n = 30$), and with 30 Monte-Carlo simulation runs. At the beginning of this analysis, the parameters of elitism and safety factor are respectively taken as ($Elit = 1$ or 4) and ($SF = 0.9$ or 0.98) and obtained by trail and error approach.

Table 4.13 shows the best, mean and standard deviation of the 36 test functions with $Elit = 1$ and $SF = 0.9$. The bolded numbers indicates the best result achieved for each test function. As an overall observation, it can be clearly seen that the third proposed model (ISS-BBO) outperforms the others in most cases (31 cases out of 36 achieved lowest error). BMU model achieved good result in some few cases, but it is likely to trap into local minima; **Shekel family** and **Schwefel's F1.2** are obvious examples of this condition. On the other hand, ISS model has high immunity to avoid trapping into local minima as compared to BMU model. However, it may happen in few trials out of 30 Monte-Carlo simulation runs, which gives poor means for that few test functions, such as **Schaffer's F1**, **Box-Betts**, **Fletcher-Powell** and **Schwefel's F1.2**. BSU model

has the highest immunity against trapping into local minima. However, it shows the least performance as compared to BMU and ISS models, because it modifies the search space only one time at the end of the first stage. Based on that, ISS model combines the advantages of both BSU and BMU models, and at the same time, minimizes the disadvantages of both models.

Table 4.14 shows the CPU time performance of all the proposed models that are simulated using the same machine. The proposed modifications on the original BBO form enhanced its processing speed and accuracy. The BSU model achieved the fastest CPU time due to the fact that it modifies the variables bounds only one time. ISS model achieved the second fastest model with only 0.3% higher CPU time as compared to BSU model. The original BBO algorithm is the slowest among the four models. It takes 31.9% more CPU time than BSU model. Therefore, ISS model achieved the best performance that compromise between the speed and accuracy.

Figure (4.14) shows log-log curves of the fitness functions of the original BBO algorithm in comparison with that of the BSU, BMU and ISS models for **Ackley's function "f30-01"**, **Generalized Rastrigin's function "f30-05"**, **Schwefel's problem 2.21 "f30-12"** and **Sphere Model "f30-14"**. It can be clearly seen that the ISS model can converge to a better solution than the other models.

With fixed population size $\bar{k}$ and total number of generations G_T, it has been observed that the performance can also be effectively improved

Table (4.13)
Comparison of results over 30 trials of the PMB-BBO, BSU-BBO, BMU-BBO and ISS-BBO Models

f #	Function Name	n	Models of Biogeography Based Optimization (BBO)											
			PMB-BBO			BSU-BBO			BMU-BBO			ISS-BBO		
			Best	Mean	StdDev	Best	Mean	StdDev	Best	Mean	StdDev	Best	Mean	StdDev
f1-03	Shekel's F1	1	9.803E-08	2.872E-05	4.294E-05	1.349E-08	3.185E-06	4.578E-06	**1.776E-15**	1.612E-05	4.009E-05	3.553E-15	**8.056E-12**	**1.372E-11**
f1-04	Shekel's F2	1	8.196E-10	2.155E-05	5.219E-05	5.979E-09	5.945E-06	1.032E-05	**1.776E-15**	1.122E-05	1.510E-05	2.132E-14	**1.483E-11**	**2.802E-11**
f1-05	Shekel's F3	1	1.745E-08	7.462E-05	1.873E-04	4.844E-09	5.608E-06	1.232E-05	1.101E-13	3.666E-05	7.964E-05	**1.599E-14**	**9.119E-12**	**1.725E-11**
f2-10	Branin RCOS	2	1.348E-08	1.478E-04	1.251E-04	1.045E-07	4.538E-05	8.533E-05	1.247E-06	7.717E-05	2.458E-04	**6.075E-13**	**3.777E-11**	**5.936E-11**
f2-12	Bukin's F6	2	1.614E-01	5.680E-01	3.168E-01	9.639E-02	5.040E-01	2.794E-01	**7.830E-03**	**3.769E-02**	**2.828E-02**	8.203E-03	7.718E-02	5.874E-02
f2-20	Davis	2	1.193E-01	4.260E-01	1.553E-01	1.262E-02	2.039E-01	1.156E-01	1.163E-02	2.000E-01	1.381E-01	**2.504E-03**	**8.700E-03**	**3.197E-03**
f2-29	Goldstein-Price	2	2.417E-05	2.054E-03	2.883E-03	1.146E-07	6.626E-05	7.512E-05	3.253E-06	1.766E-04	3.213E-04	**5.513E-12**	**2.328E-10**	**2.626E-10**
f2-46	Schaffer's F1	2	1.699E-04	8.805E-03	**2.594E-03**	6.463E-05	7.934E-03	3.588E-03	1.021E-05	7.725E-03	3.431E-03	**4.548E-11**	**6.153E-03**	4.682E-03
f2-47	Schaffer's F2	2	7.907E-01	7.162E+00	4.076E+00	2.649E-01	1.909E+00	1.097E+00	**2.205E-05**	**1.088E-02**	5.383E-02	1.073E-02	3.859E-02	**1.866E-02**
f2-60	6-Hump Camel-Back	2	2.456E-06	9.025E-05	8.037E-05	1.455E-09	6.655E-06	9.800E-06	8.253E-08	1.558E-05	2.604E-05	**6.350E-14**	**3.153E-11**	**5.193E-11**
f3-01	Box-Betts	3	1.098E-07	1.781E-06	1.475E-06	5.346E-09	3.778E-07	4.665E-07	7.155E-09	4.562E-07	6.002E-07	**9.628E-14**	**3.035E-09**	**7.566E-09**
f3-03	Fletcher-Powell	3	3.431E-02	9.500E-01	9.346E-01	2.163E-03	1.141E-01	**1.078E-01**	4.221E-03	2.731E-01	3.880E-01	**4.657E-07**	**3.541E-02**	1.088E-01
f3-05	Meyer-Roth	3	1.669E-05	1.121E-04	6.532E-05	5.121E-06	2.713E-05	2.251E-05	4.312E-06	3.301E-05	2.959E-05	**3.553E-06**	**4.634E-06**	**2.462E-06**
f3-06	Perm F1	3	1.508E-01	1.510E+00	8.583E-01	8.136E-03	2.795E-01	**3.112E-01**	8.127E-03	8.101E-01	7.902E-01	**4.582E-05**	**2.398E-01**	3.635E-01
f4-05	Neumaier's F2	4	8.240E-04	4.683E-02	3.343E-02	2.619E-03	2.810E-02	2.680E-02	3.581E-03	2.882E-02	2.698E-02	**3.160E-05**	**6.014E-03**	**9.226E-03**
f4-06	Wood (or Colville)	4	4.117E-01	5.865E+00	4.252E+00	7.177E-02	1.131E+00	8.345E-01	2.579E-02	1.570E+00	1.680E+00	**2.297E-04**	**3.209E-01**	**7.247E-01**
f6-02	Perm F2	6	8.175E-02	1.174E+00	8.417E-01	2.731E-02	5.922E-01	6.195E-01	3.870E-02	6.853E-01	6.417E-01	**3.594E-04**	**4.163E-01**	**4.330E-01**
f9-02	Price's Transistor	9	4.323E+01	1.317E+02	4.864E+01	1.141E+01	5.860E+01	**3.621E+01**	5.861E+00	8.451E+01	4.746E+01	**1.003E-01**	**4.857E+01**	3.816E+01
f9-03	Storn's Chebyshev	9	5.599E+03	5.383E+04	3.142E+04	1.461E+03	1.782E+04	1.410E+04	2.987E+03	1.587E+04	**9.405E+03**	**6.707E+02**	**1.018E+04**	1.176E+04
f10-01	Epistatic Michalewicz	10	3.029E-01	1.440E+00	5.471E-01	2.209E-01	6.969E-01	4.126E-01	2.329E-01	9.141E-01	4.799E-01	**1.213E-01**	**5.702E-01**	**3.275E-01**
f15-02	Neumaier's F3 (or Trid)	15	9.680E+01	7.499E+02	6.248E+02	4.720E+01	7.369E+02	7.479E+02	1.151E+02	5.523E+02	**3.406E+02**	**1.082E+01**	**4.075E+02**	3.835E+02
f15-03	Normalized Rana	15	3.523E+01	5.751E+01	1.549E+01	3.911E+01	5.301E+01	8.523E+00	3.376E+01	5.198E+01	**1.121E+01**	**2.435E+01**	**4.686E+01**	1.223E+01
f17-01	Bent Cigar	17	1.037E+06	2.825E+06	1.363E+06	1.246E+05	8.877E+05	5.017E+05	3.840E+05	2.018E+06	9.441E+05	**4.557E+00**	**1.827E+02**	**5.364E+02**
f20-01	Alpine	20	1.800E-02	3.517E-02	9.710E-03	1.250E-02	2.086E-02	5.101E-03	2.116E-02	3.886E-02	1.396E-02	**2.798E-04**	**1.112E-03**	**8.593E-04**
f30-01	Ackley	30	5.044E-01	9.553E-01	3.153E-01	3.100E-01	6.666E-01	1.914E-01	7.437E-01	1.217E+00	2.554E-01	**1.229E-03**	**7.433E-03**	**8.657E-03**
f30-02	Gen. Griewank	30	1.017E+00	1.036E+00	**1.240E-02**	7.591E-01	9.929E-01	6.330E-02	1.020E+00	1.047E+00	1.347E-02	**8.314E-04**	**3.787E-02**	3.403E-02
f30-03	Gen. Penalized F1	30	2.418E-03	2.148E-02	2.608E-02	1.736E-03	1.434E-02	1.276E-02	3.425E-03	2.952E-02	4.129E-02	**1.057E-07**	**2.203E-06**	**3.327E-06**
f30-04	Gen. Penalized F2	30	8.998E-02	1.933E-01	5.812E-02	4.416E-02	1.227E-01	3.884E-02	1.236E-01	2.354E-01	7.022E-02	**1.313E-06**	**3.922E-04**	**1.974E-03**
f30-05	Gen. Rastrigin	30	6.513E-01	1.860E+00	6.546E-01	6.006E-01	1.308E+00	4.193E-01	1.357E+00	2.608E+00	8.232E-01	**6.993E-06**	**1.999E-01**	**3.978E-01**
f30-06	Gen. Rosenbrock	30	1.192E+02	3.690E+02	4.932E+02	1.194E+02	2.380E+02	**1.283E+02**	1.585E+02	3.460E+02	1.590E+02	**2.274E+00**	**2.242E+02**	4.082E+02
f30-07	Gen. Schwefel's F2.26	30	4.716E-06	1.333E-05	5.012E-06	3.928E-06	8.821E-06	3.868E-06	5.670E-06	1.713E-05	8.031E-06	**3.568E-07**	**3.555E-06**	**3.027E-06**
f30-11	Schwefel's F1.2	30	2.110E+01	2.238E+04	1.418E+04	1.536E+00	4.604E+02	5.234E+02	**6.196E-13**	2.350E+00	7.025E+00	3.588E-07	**4.328E-03**	**1.115E-02**
f30-12	Schwefel's F2.21	30	4.292E+00	6.292E+00	1.013E+00	3.093E+00	4.759E+00	7.136E-01	1.501E+00	3.378E+00	1.053E+00	**9.004E-02**	**2.516E-01**	**8.386E-02**
f30-13	Schwefel's F2.22	30	4.379E-01	7.235E-01	1.234E-01	3.682E-01	5.526E-01	1.061E-01	5.095E-01	8.358E-01	1.506E-01	**2.274E-03**	**6.357E-03**	**4.137E-03**
f30-14	Sphere Model	30	1.383E+00	3.950E+00	1.349E+00	7.558E-01	2.141E+00	1.067E+00	1.811E+00	5.095E+00	2.069E+00	**6.086E-05**	**2.196E-03**	**3.258E-03**
f30-15	Step	30	0.000E+00	5.033E+00	2.025E+00	0.000E+00	2.000E+00	1.265E+00	2.000E+00	6.700E+00	2.340E+00	**0.000E+00**	**0.000E+00**	**0.000E+00**
Total Winnings out of 36 Functions			1/36	0/36	2/36	1/36	0/36	4/36	5/36	2/36	4/36	**31/36**	**34/36**	**26/36**

Table (4.14)
Normalized CPU time of the 30-dimensional problems $\{f30\text{-}01$ to $f30\text{-}15\}$

f #	Function Name	n	Models of Biogeography-Based Optimization			
			BBO	BSU-BBO	BMU-BBO	ISS-BBO
f30-01	Ackley	30	6839.283	**5046.704**	6311.529	5048.630
f30-02	Gen. Griewank	30	6916.988	**5167.159**	6396.068	5180.183
f30-03	Gen. Penalized F1	30	7621.550	5816.850	7097.600	**5717.747**
f30-04	Gen. Penalized F2	30	7511.773	5708.535	6877.032	**5601.522**
f30-05	Gen. Rastrigin	30	6637.605	**4803.442**	6093.325	4806.490
f30-06	Gen. Rosenbrock	30	6465.583	**4725.401**	5885.246	4761.199
f30-07	Gen. Schwefel's F2.26	30	6651.042	**4897.813**	6092.103	4933.595
f30-11	Schwefel's F1.2	30	10387.646	**9233.976**	11270.307	9434.560
f30-12	Schwefel's F2.21	30	6216.846	**4642.369**	5685.971	4662.106
f30-13	Schwefel's F2.22	30	6583.103	4755.856	5896.386	**4702.883**
f30-14	Sphere Model	30	6440.180	**4706.285**	5804.437	4941.985
f30-15	Step	30	6504.682	4774.090	5874.912	**4690.629**
Avg. CPU Time (sec)			7064.690	**5356.540**	6607.076	5373.461
Normalized Avg. CPU Time			1.319	**1.000**	1.233	1.003

by changing the values of $Elit$, m_{max}, R_m, R and/or SF parameter. Now, let's select $Elit$ to be 4 and SF to be 0.98, while the other parameters are kept without any change. Table 4.15 shows the comparison of ISS-BBO with different values of $Elit$ and SF. From that table, ISS-BBO gives better mean and standard deviation at $(Elit = 4)$ and $(SF = 0.90)$ for $high$-dimensional problems; while for low-dimensional problems, it works better at $(SF = 0.98)$. The best results are obtained when $Elit$ and SF are respectively set at 4 and 98%. However, still there is no enough study to select the optimum parameters that can compromise between the different parts of the performance criteria.

ISS-BBO with the parameters of $(Elit = 4$ and $SF = 0.90)$ is selected in order to be compared with the results of other BBO models in the literature. The Sin-BBO and the GenSin-BBO algorithms are selected here (Ma & Simon, 2011a). GenSin-BBO is the most advanced model available in the literature that comes with one of the most complicated migration rates.

Table (4.15)
Performance comparison of ISS model with different values of $Elit$ and SF

f #	Function Name	n	Iterative Search Space - Biogeography Based Optimization (ISS-BBO)											
			Elitisim = 1 & SF = 90%			Elitisim = 4 & SF =90%			Elitisim = 1 & SF = 98%			Elitisim = 4 & SF = 98%		
			Best	Mean	StdDev	Best	Mean	StdDev	Best	Mean	StdDev	Best	Mean	StdDev
f1-03	Shekel's F1	1	3.553E-15	8.056E-12	1.372E-11	1.776E-14	4.805E-12	5.962E-12	**0.000E+00**	**2.368E-15**	1.059E-15	1.776E-15	**2.368E-15**	**8.374E-16**
f1-04	Shekel's F2	1	2.132E-14	1.483E-11	2.802E-11	3.553E-15	6.334E-12	1.121E-11	**1.776E-15**	3.257E-15	8.054E-16	**1.776E-15**	**3.079E-15**	**7.855E-16**
f1-05	Shekel's F3	1	1.599E-14	9.119E-12	1.725E-11	5.329E-15	7.699E-12	1.261E-11	**1.776E-15**	4.026E-15	1.019E-15	**1.776E-15**	**3.790E-15**	**9.979E-16**
f2-10	Branin RCOS	2	6.075E-13	**3.777E-11**	**5.936E-11**	2.528E-12	7.207E-11	2.137E-10	**0.000E+00**	1.231E-06	6.597E-06	**0.000E+00**	2.446E-08	1.206E-07
f2-12	Bukin's F6	2	8.203E-03	7.718E-02	5.874E-02	1.397E-02	6.981E-02	4.632E-02	**4.041E-03**	**3.649E-02**	1.810E-02	7.232E-03	3.976E-02	**1.732E-02**
f2-20	Davis	2	2.504E-03	8.700E-03	3.197E-03	4.317E-03	9.252E-03	3.441E-03	4.820E-04	**1.359E-03**	**2.053E-03**	**4.178E-04**	1.606E-03	2.254E-03
f2-29	Goldstein-Price	2	5.513E-12	2.328E-10	2.626E-10	1.560E-12	2.599E-10	3.450E-10	**7.683E-14**	**4.669E-14**	**3.378E-14**	7.683E-14	8.427E-14	2.030E-13
f2-46	Schaffer's F1	2	4.548E-11	6.153E-03	4.682E-03	4.300E-10	**5.182E-03**	4.847E-03	**2.509E-14**	8.744E-03	**2.915E-03**	3.653E-14	8.420E-03	3.303E-03
f2-47	Schaffer's F2	2	1.073E-02	3.859E-02	1.866E-02	9.123E-03	4.191E-02	1.708E-02	**1.790E-04**	**1.435E-03**	**1.100E-03**	2.602E-04	2.075E-03	2.443E-03
f2-60	6-Hump Camel-Back	2	6.350E-14	3.153E-11	5.193E-11	4.019E-14	1.223E-11	1.465E-11	**6.661E-16**	**5.462E-15**	**1.743E-14**	8.882E-16	7.960E-14	2.687E-13
f3-01	Box-Betts	3	9.628E-14	**3.035E-09**	**7.566E-09**	2.520E-14	1.207E-08	2.817E-08	**1.684E-15**	5.832E-07	9.691E-07	1.447E-13	6.101E-07	9.396E-07
f3-03	Fletcher-Powell	3	4.657E-07	**3.541E-02**	**1.088E-01**	**2.990E-08**	5.136E-02	1.197E-01	2.355E-07	5.992E-01	6.141E-01	9.172E-07	7.750E-01	1.037E+00
f3-05	Meyer-Roth	3	**3.553E-06**	**4.634E-06**	**2.462E-06**	3.553E-06	2.853E-05	1.027E-04	3.553E-06	4.836E-05	5.923E-05	3.558E-06	8.686E-05	1.290E-04
f3-06	Perm F1	3	4.582E-05	2.398E-01	3.635E-01	1.092E-05	3.122E-01	**3.626E-01**	1.438E-05	**2.381E-01**	3.882E-01	**4.945E-08**	3.340E-01	4.562E-01
f4-05	Neumaier's F2	4	3.160E-05	**6.014E-03**	**9.226E-03**	6.251E-05	7.429E-03	1.180E-02	3.285E-05	1.179E-02	1.361E-02	**4.438E-08**	1.300E-02	1.435E-02
f4-06	Wood (or Colville)	4	2.297E-04	3.209E-01	7.247E-01	**9.467E-05**	**1.766E-01**	**1.882E-01**	9.740E-04	9.315E-01	1.502E+00	1.280E-04	7.358E-01	1.068E+00
f6-02	Perm F2	6	**3.594E-04**	4.163E-01	4.330E-01	1.155E-03	**1.666E-01**	**1.869E-01**	3.444E-02	5.794E-01	5.122E-01	1.270E-03	1.901E-01	2.624E-01
f9-02	Price's Transistor	9	**1.003E-01**	4.857E+01	3.816E+01	5.235E+00	**4.063E+01**	**3.379E+01**	5.470E-01	4.447E+01	3.515E+01	2.217E+00	4.876E+01	4.076E+01
f9-03	Storn's Chebyshev	9	6.707E+02	1.018E+04	1.176E+04	3.654E+02	**6.347E+03**	**6.843E+03**	4.149E+02	2.143E+04	1.450E+04	**2.159E+02**	1.346E+04	1.095E+04
f10-01	Epistatic Michalewicz	10	1.213E-01	5.702E-01	3.275E-01	6.198E-02	**4.436E-01**	**2.172E-01**	8.377E-02	6.651E-01	4.082E-01	**3.985E-02**	4.945E-01	2.703E-01
f15-02	Neumaier's F3 (or Trid)	15	1.082E+01	4.075E+02	3.835E+02	1.019E+01	**2.463E+02**	**2.585E+02**	**7.827E+00**	4.094E+02	4.045E+02	1.214E+01	4.977E+02	5.949E+02
f15-03	Normalized Rana	15	2.435E+01	4.686E+01	1.223E+01	1.604E+01	3.657E+01	**1.029E+01**	2.040E+01	4.870E+01	1.444E+01	**6.145E+00**	**3.559E+01**	1.374E+01
f17-01	Bent Cigar	17	4.557E+00	1.827E+02	5.364E+02	5.054E-02	**9.007E-01**	**1.800E+00**	1.407E+01	3.944E+03	6.111E+03	**6.398E-03**	9.409E+00	3.039E+01
f20-01	Alpine	20	2.798E-04	1.112E-03	8.593E-04	**5.298E-05**	**2.209E-04**	**1.959E-04**	1.750E-04	1.529E-03	1.451E-03	6.253E-05	3.292E-04	3.260E-04
f30-01	Ackley	30	1.229E-03	7.433E-03	8.657E-03	1.236E-04	**6.872E-04**	**6.596E-04**	4.946E-03	6.491E-02	1.083E-01	**1.117E-04**	2.485E-03	2.939E-03
f30-02	Gen. Griewank	30	8.314E-04	3.787E-02	3.403E-02	**3.417E-06**	3.139E-02	3.267E-02	7.705E-03	6.509E-02	4.640E-02	4.860E-06	**2.363E-02**	**2.142E-02**
f30-03	Gen. Penalized F1	30	1.057E-07	2.203E-06	3.327E-06	1.660E-09	**1.261E-07**	**5.408E-07**	7.813E-07	6.932E-03	2.586E-02	**1.063E-09**	9.987E-07	2.421E-06
f30-04	Gen. Penalized F2	30	1.313E-06	3.922E-04	1.974E-03	**1.282E-08**	**1.893E-07**	**2.007E-07**	4.672E-06	9.559E-04	2.643E-03	3.730E-08	1.440E-05	4.851E-05
f30-05	Gen. Rastrigin	30	6.993E-06	1.999E-01	3.978E-01	**4.602E-07**	**4.715E-06**	**6.427E-06**	1.726E-03	4.707E-01	4.932E-01	5.182E-07	1.186E-04	2.382E-04
f30-06	Gen. Rosenbrock	30	2.274E+00	2.242E+02	4.082E+02	**4.423E-01**	9.416E+01	6.290E+01	4.575E+01	1.730E+02	1.625E+02	3.507E+00	**6.788E+01**	**3.797E+01**
f30-07	Gen. Schwefel's F2.26	30	3.568E-07	3.555E-06	3.027E-06	**3.038E-10**	**1.155E-07**	**1.381E-07**	1.918E-06	6.164E-06	4.049E-06	1.148E-08	7.539E-07	6.147E-07
f30-11	Schwefel's F1.2	30	3.588E-07	4.328E-03	1.115E-02	2.204E-11	**3.001E-06**	**5.216E-06**	3.838E-08	1.990E-01	5.609E-01	**1.652E-15**	4.600E-06	2.379E-05
f30-12	Schwefel's F2.21	30	9.004E-02	2.516E-01	8.386E-02	2.706E-02	**4.690E-02**	**1.609E-02**	1.108E-01	8.613E-01	1.080E+00	**8.582E-03**	7.019E-02	2.162E-01
f30-13	Schwefel's F2.22	30	2.274E-03	6.357E-03	4.137E-03	2.969E-04	**7.600E-04**	**4.141E-04**	5.895E-03	1.497E-02	8.287E-03	**2.149E-04**	1.939E-03	1.709E-03
f30-14	Sphere Model	30	6.086E-05	2.196E-03	3.258E-03	5.526E-07	**1.177E-05**	**2.154E-05**	6.304E-04	1.456E-02	1.922E-02	**1.241E-07**	1.633E-04	2.709E-04
f30-15	Step	30	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
Total Winnings out of 36 Functions			4/36	6/36	6/36	9/36	**19/36**	**20/36**	12/36	8/36	6/36	**18/36**	7/36	7/36

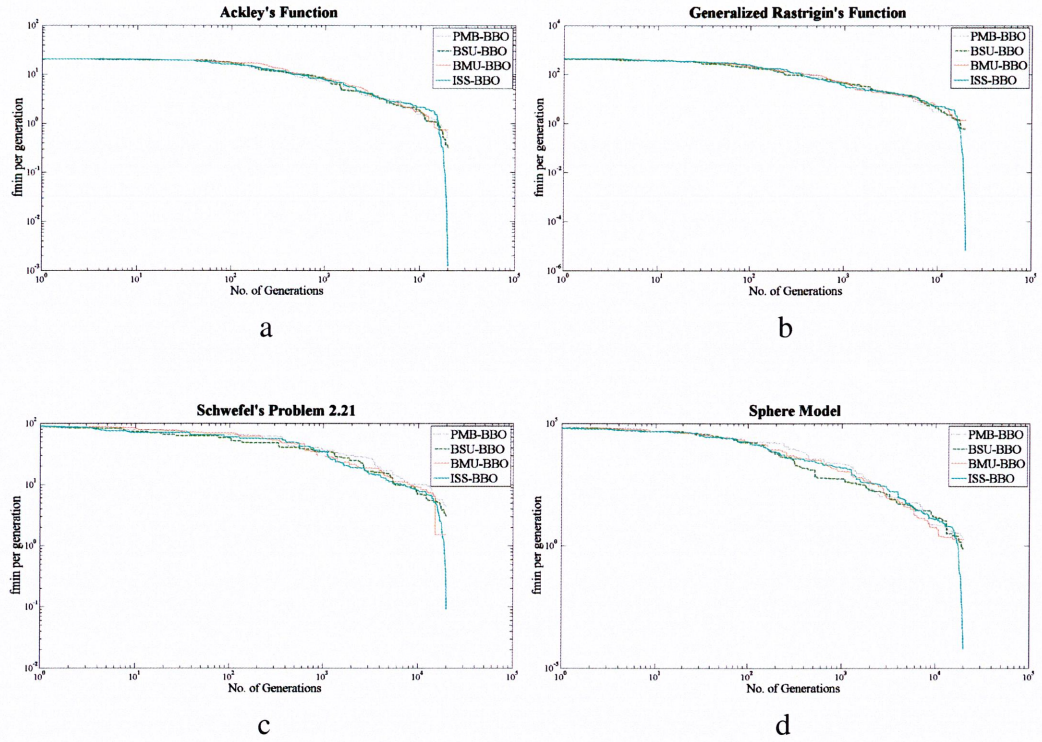


Figure (4.14)
Log-Log curves of fitness functions of BBO, BSU, BMU and ISS for some selected functions. (a) $f30$-01, (b) $f30$-05, (c) $f30$-12 and (d) $f30$-14

Table 4.16 gives the best, mean and standard deviation of similar 18 benchmark functions. As an overall performance evaluation, ISS-BBO wins with 12 lowest errors, 9 best mean and 6 best standard deviation. The comparison with Sin-BBO shows that the ISS model wins in all performance criteria, while the comparison with GenSin-BBO shows that the last one wins 10 best standard deviation out of 18. The difference between ISS-BBO and GenSin-BBO in terms of mean and standard deviation is very small (10-8 and 8-10). However, for the best errors, ISS-BBO outperforms GenSin-BBO with total winning of 13 out of 18.

Note that, the beauty of this comparison is that the GenSin-BBO is considered as the most advanced BBO model that takes into account many aspects, like complicated migration rates, mutation rates, different

Table (4.16)

Comparison of results over 30 trials of ISS-BBO with linear migration rates and BBO with sinusoidal and generalized sinusoidal migration rates

f #	Function Name	n	Models of Biogeography-Based Optimization									ISS vs. Sin			ISS vs. GenSin		
			Sin-BBO			GenSin-BBO			ISS-BBO								
			Best	Mean	StdDev	Best	Mean	StdDev	Best	Mean	StdDev	Best	Mean	StdDev	Best	Mean	StdDev
f1-03	Shekel's F1	1	5.29E-08	6.14E-07	6.60E-08	2.69E-08	3.57E-08	6.96E-08	**1.78E-14**	**4.80E-12**	**5.96E-12**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**
f1-04	Shekel's F2	1	9.60E-12	7.89E-10	2.31E-10	6.65E-15	**3.47E-14**	**5.73E-15**	**3.55E-15**	6.33E-12	1.12E-11	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	0 - 1	0 - 1
f1-05	Shekel's F3	1	3.55E-12	7.34E-11	**5.78E-12**	1.75E-10	4.54E-10	1.58E-10	**5.33E-15**	**7.70E-12**	1.26E-11	**1 - 0**	**1 - 0**	0 - 1	**1 - 0**	**1 - 0**	**1 - 0**
f2-10	Branin RCOS	2	**2.17E-15**	**1.44E-14**	**4.77E-15**	5.34E-10	3.94E-09	3.89E-10	2.53E-12	7.21E-11	2.14E-10	0 - 1	0 - 1	0 - 1	**1 - 0**	**1 - 0**	**1 - 0**
f2-29	Goldstein-Price	2	6.06E-15	7.05E-15	2.56E-16	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.56E-12	2.60E-10	3.45E-10	0 - 1	0 - 1	0 - 1	0 - 1	0 - 1	0 - 1
f2-60	6-Hump Camel-Back	2	2.67E-09	1.51E-08	7.32E-09	7.11E-12	4.02E-11	**1.22E-12**	**4.02E-14**	**1.22E-11**	1.47E-11	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	0 - 1
f30-01	Ackley	30	2.54E-01	9.71E-01	7.16E-02	1.14E-03	2.54E-03	8.25E-04	**1.24E-04**	**6.87E-04**	**6.60E-04**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**
f30-02	Gen. Griewank	30	7.49E-01	1.97E+00	9.33E-01	2.55E-01	3.65E-01	**1.72E-02**	**3.42E-06**	**3.14E-02**	3.27E-02	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	0 - 1
f30-03	Gen. Penalized F1	30	**2.26E-30**	**4.11E-30**	**8.45E-31**	3.74E-25	7.81E-25	5.38E-26	1.66E-09	1.26E-07	5.41E-07	0 - 1	0 - 1	0 - 1	0 - 1	0 - 1	0 - 1
f30-04	Gen. Penalized F2	30	1.28E-10	7.36E-10	6.32E-11	**1.07E-11**	**9.05E-11**	**6.72E-12**	1.28E-08	1.89E-07	2.01E-07	0 - 1	0 - 1	0 - 1	0 - 1	0 - 1	0 - 1
f30-05	Gen. Rastrigin	30	1.55E-03	1.21E-02	3.78E-03	1.15E-04	4.38E-04	9.33E-05	**4.60E-07**	**4.71E-06**	**6.43E-06**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**
f30-06	Gen. Rosenbrock	30	9.24E-01	3.47E+00	4.33E-01	6.54E-01	**9.36E-01**	**2.44E-01**	**4.42E-01**	9.42E+01	6.29E+01	**1 - 0**	0 - 1	0 - 1	**1 - 0**	0 - 1	0 - 1
f30-07	Gen. Schwefel's F2.26	30	2.63E-06	5.08E-06	2.74E-06	6.85E-09	**9.38E-09**	**4.76E-10**	**3.04E-10**	1.15E-07	1.38E-07	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	0 - 1	0 - 1
f30-11	Schwefel's F1.2	30	6.33E-02	8.31E-02	1.17E-02	2.71E-03	3.08E-03	5.71E-04	**2.20E-11**	**3.00E-06**	**5.22E-06**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**
f30-12	Schwefel's F2.21	30	5.68E-15	5.34E-14	2.54E-15	**5.36E-15**	**4.64E-14**	**1.07E-15**	2.71E-02	4.69E-02	1.61E-02	0 - 1	0 - 1	0 - 1	0 - 1	0 - 1	0 - 1
f30-13	Schwefel's F2.22	30	1.84E-04	8.03E-04	3.84E-04	**1.22E-04**	**1.47E-04**	**4.86E-05**	2.97E-04	7.60E-04	4.14E-04	0 - 1	**1 - 0**	0 - 1	0 - 1	0 - 1	0 - 1
f30-14	Sphere Model	30	2.17E-02	6.38E-02	5.46E-03	9.55E-04	8.24E-03	9.78E-04	**5.53E-07**	**1.18E-05**	**2.15E-05**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**
f30-15	Step	30	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.16E-15	8.21E-15	1.83E-16	**0.00E+00**	**0.00E+00**	**0.00E+00**	1 - 1	1 - 1	1 - 1	**1 - 0**	**1 - 0**	**1 - 0**
Total Winnings out of 36 Benchmark Functions			3/18	3/18	4/18	4/18	7/18	**9/18**	**12/18**	**9/18**	6/18	**12 - 7**	**12 - 7**	**10 - 9**	**13 - 5**	**10 - 8**	8 - 10

bias of sinusoidal angles and is tested after completing further analysis by using BBO Markov model. In contrast, ISS-BBO uses only the simplest BBO model, as shown in Fig.(4.4), as a basis of its algorithm. In addition, ISS-BBO can save 31.6% of the total consumed CPU time as compared to the regular BBO algorithm.

Although, GenSin-BBO is the winner in terms of the total best standard deviations, ISS-BBO showed competitive performance. For low-dimensional problems there is no winner, because each one of them will win with 3 best results out of 6. For $high$-dimensional problems, ISS-BBO wins with 5 best results out of 12, and it is near to the balanced number which is 6.

4.5.2.1 ISS-BBO versus Scan-Zoom

One important note that has to be clarified in this study is that: this modification has similar concept to the **Scan and Zoom Optimization Technique**, which is introduced by P. Venkataraman (Venkataraman, 2009), but each works in different way. Firstly, Scan and Zoom starts at the mid-point of each SIV (center of design region, SIV_c), while ISS-BBO is a population-based algorithm. Secondly, Scan and Zoom depends on a linearly and equally spaced points between lower and upper bounds of each SIV, while ISS-BBO selects these points randomly. Thirdly, Scan and Zoom depends on the resolution between these points to avoid trapping into local optima, while ISS-BBO does not require

this restriction. Fourthly, for $l = 1, 2, \cdots, L$ of zooming levels, updated centralized SIV_c^l is equivalent to SIV_s^l at ISI_{best}^l in ISS-BBO, and updated ΔSIV_s^l is equal to the half of ΔSIV_s^{l-1} where $SIV_{min,s}^l = SIV_c^l - \Delta SIV_s^l$ and $SIV_{max,s}^l = SIV_c^l + \Delta SIV_s^l$; while ISS-BBO selects $SIV_{min,s}^l$ and $SIV_{max,s}^l$ based on vectors of n SIV that have been collected from the best individuals of $\bar{k}$-islands' vectors and after completing G_l generations, see pseudocode of ISS-BBO in Algorithm 10.

4.5.3 conclusions and Suggestions

This Section proposed new model called ISS-BBO that solves the associated weakness by updating the problem's side constraints to create rich search space without violating its initial bounds. The updating process is done based on the enough collected information from the unerased best individuals of the past generations, which is one of the main features of BBO. In the proposed ISS-BBO, the updating process is achieved in an iterative way, where the new lower and upper bounds will always be bigger and smaller than their old values, respectively. The experimental results show that the ISS-BBO outperforms both the original BBO form and the other two proposed models (BSU and BMU). Also, it can perform better than the most advanced BBO algorithms available in the literature that are equipped with some of the most complicated migration rates (Sin-BBO and GenSin-BBO). In addition, ISS-BBO can save 31.6% of CPU time as compared to the other BBO models with same number of function evaluations (NFE).

In this study, the basic BBO form, where λ_s and μ_s are linear and $I = E$, is used. It is highly recommended to study the ISS model under different migration and mutation rates. Also, Oppositional and Blended versions (OBBO and BBBO) can be used as advanced add-ons to improve the overall performance of the ISS-BBO algorithm.

4.6 Adaptive Variable Bounds - Biogeography Based Optimization (AVB-BBO)

From Section 4.5, the variable bounds of each independent variable are updated in an iterative way. This process will be replaced with other process called **Adaptive Variable Bounds (AVB)**. It can be done by just changing the conditions of the "if-statements" in Algorithm 10. In AVB model, the values of the new lower and upper bounds are adaptively increased or decreased (*i.e., zooming-in or zooming-out*), which may raise the level of algorithm's immunity against trapping into local minima. To differentiate between these two updating mechanisms, see the difference between Fig.(4.11b) and Fig.(4.15).

The adaptive process can be expressed in many ways. In this thesis, AVB model is done by modifying Eq.(4.28) and Eq.(4.45) to be respectively as Eq.(4.46) and Eq.(4.47), so the new bounds can be updated as follows:

$$SIV^{new}_{min,s} = SIV_{s,1} \times SF \tag{4.46}$$

$$SIV^{new}_{max,s} = SIV_{s,G_l} \times SF^{-1} \tag{4.47}$$

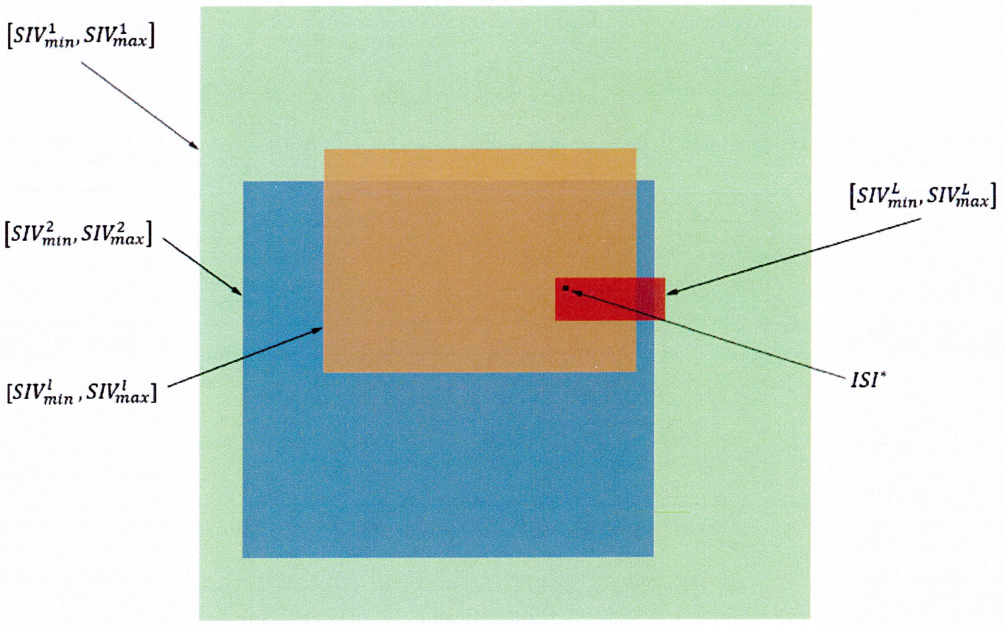


Figure (4.15)
Illustration in the adaptive variable bounds and the location of the optimal solution

The complete process of AVB model is described through Algorithm 11.

Note that, to make Algorithm 11 works smoothly, it is important to set ζ and ξ as follows:

- ζ is a constant or random value in the range of $(0 < \zeta < 1)$.
- ξ is a constant or random value in the range of $(\xi > 1)$.

This means that the AVB model depends on ζ and ξ whether they are constant or variable for each generation and how much their values. In this research, ζ and ξ are simplified as *rand* and *rand+1*, respectively.

4.6.1 Comparison Between AVB and Iterative Models

The proposed AVB model has been compared with the obtained results from the previously proposed models (*i.e., BSU-BBO, BMU-BBO*

and ISS-BBO). It is tested using the same 36 benchmark functions and same initialization parameters ($Elit$ and SF are taken as 1 and 98%, respectively).

Algorithm 11 The pseudocode of the AVB-BBO model

for $l \leftarrow 1$ *to* L **do**
  Apply Eq.(4.40) for computing G_l
  for $g \leftarrow 1$ *to* G_l **do** {bounds are iterated once time at the end of this loop}
    Start BBO normally for stage G_l
    Apply Eq.(4.43)
  end for
  if $G_l \geqslant$ MGN **then** {MGN: Minimum Generation Number}
    Apply Eq.(4.44), and then Eq.(4.27)
    if $SIV^l_{min,s} \leqslant SIV_{s,1} \leqslant SIV^l_{max,s}$ **then**
      Apply Eq.(4.46) for $SIV^{l+1}_{min,s}$
    end if
    if $SIV^l_{min,s} \leqslant SIV_{s,G_l} \leqslant SIV^l_{max,s}$ **then**
      if $SIV^{l+1}_{min,s} < Eq.(4.47)$ **then**
        Apply Eq.(4.47) for $SIV^{l+1}_{max,s}$
      end if
    else
      $SIV^{l+1}_{max,s} = SIV^l_{max,s}$
    end if
    while $|SIV^{l+1}_{max,s} - SIV^{l+1}_{min,s}| < \epsilon$ **do**
      if sign($SIV^{l+1}_{min,s}$) is +ve **then**
        $SIV^{l+1}_{min,s} = SIV^{l+1}_{min,s} \times \zeta$
      else
        $SIV^{l+1}_{min,s} = SIV^{l+1}_{min,s} \times \xi$
      end if
      if sign($SIV^{l+1}_{max,s}$) is +ve **then**
        $SIV^{l+1}_{max,s} = SIV^{l+1}_{max,s} \times \xi$
      else
        $SIV^{l+1}_{max,s} = SIV^{l+1}_{max,s} \times \zeta$
      end if
    end while
  end if
end for

Table 4.17 shows the best, mean and standard deviation, where the bolded numbers indicates the best result achieved for each test function. As indicated from that table, AVB model outperforms BSU and BMU models. But unfortunately, it cannot win against ISS model.

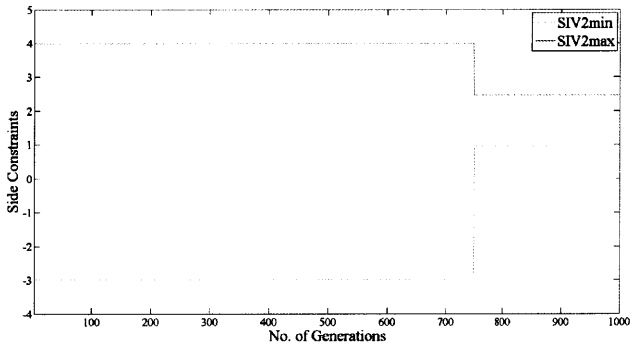
Figure (4.16) shows how the lower and upper bounds of the second independent variable (*i.e.,* SIV_2) of **Perm No.1 Function "f3-06"** be updated by iterative and adaptive models.

4.6.2 Conclusions and Suggestions

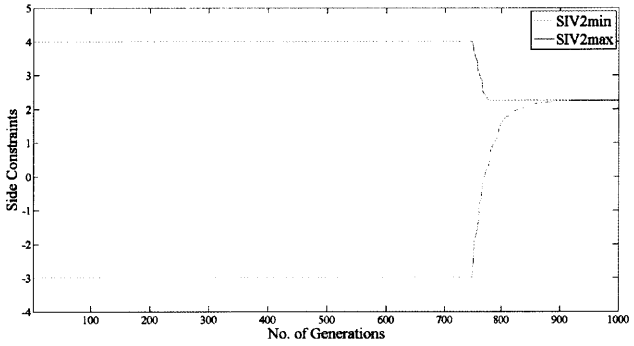
This Section proposed new model named AVB-BBO that has the ability to update the search space of such a problem adaptively. Although, AVB model outperforms BSU and BMU models, it did not perform better than the ISS model. However, the algorithm of AVB model contains simple equations, and it may perform better if some future modifications are involved. One of the main properties of this model is that the search space can be zoomed-in if the range between the lower and upper limits of each SIV is very large, and can be zoomed-out if that range becomes very narrow. With this property, the immunity against trapping into local minima is increased.

Table (4.17)
Comparison of results over 30 trials of the AVB-BBO, BSU-BBO, BMU-BBO and ISS-BBO Models

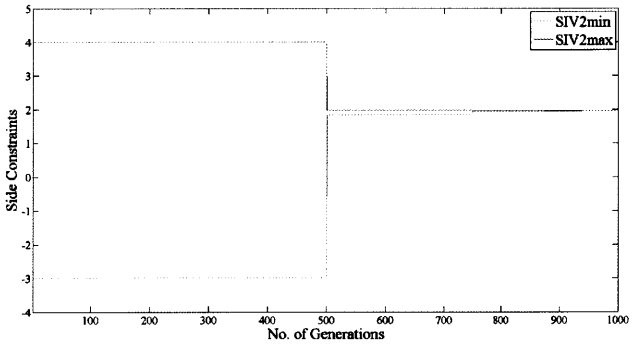
f #	Function Name	n	Updatable Search Space - Biogeography Based Optimizations											
			AVB-BBO			AVB-BBO vs. BSU-BBO			AVB-BBO vs. BMU-BBO			AVB-BBO vs. ISS-BBO		
			Best	Mean	StdDev	Best	Mean	StdDev	Best	Mean	StdDev	Best	Mean	StdDev
f1	Shekel's F1	1	2.051E-09	3.056E-06	5.415E-06	**1 - 0**	**1 - 0**	0 - 1	0 - 1	**1 - 0**	**1 - 0**	0 - 1	0 - 1	0 - 1
f2	Shekel's F2	1	5.426E-10	3.785E-06	6.475E-06	**1 - 0**	**1 - 0**	**1 - 0**	0 - 1	**1 - 0**	**1 - 0**	0 - 1	0 - 1	0 - 1
f3	Shekel's F3	1	1.332E-10	4.233E-06	5.127E-06	**1 - 0**	**1 - 0**	**1 - 0**	0 - 1	**1 - 0**	**1 - 0**	0 - 1	0 - 1	0 - 1
f4	Branin RCOS	2	2.414E-08	5.277E-06	1.387E-05	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	0 - 1	0 - 1	0 - 1
f5	Bukin's F6	2	1.105E-01	3.539E-01	1.537E-01	0 - 1	**1 - 0**	**1 - 0**	0 - 1	0 - 1	0 - 1	0 - 1	0 - 1	0 - 1
f6	Davis	2	2.186E-03	1.656E-01	1.012E-01	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	0 - 1	0 - 1
f7	Goldstein-Price	2	2.587E-08	8.701E-05	1.978E-04	**1 - 0**	0 - 1	0 - 1	**1 - 0**	**1 - 0**	**1 - 0**	0 - 1	0 - 1	0 - 1
f8	Schaffer's F1	2	2.279E-07	8.775E-03	2.826E-03	**1 - 0**	0 - 1	**1 - 0**	**1 - 0**	0 - 1	**1 - 0**	0 - 1	0 - 1	**1 - 0**
f9	Schaffer's F2	2	3.432E-03	2.099E+00	2.164E+00	**1 - 0**	0 - 1	0 - 1	0 - 1	0 - 1	0 - 1	**1 - 0**	0 - 1	0 - 1
f10	6-Hump Camel-Back	2	3.407E-11	4.734E-07	1.322E-06	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	0 - 1	0 - 1	0 - 1
f11	Box-Betts	3	1.992E-08	1.190E-06	1.263E-06	0 - 1	0 - 1	0 - 1	0 - 1	0 - 1	0 - 1	0 - 1	0 - 1	0 - 1
f12	Fletcher-Powell	3	5.775E-05	3.275E-01	5.953E-01	**1 - 0**	0 - 1	0 - 1	**1 - 0**	0 - 1	0 - 1	0 - 1	0 - 1	0 - 1
f13	Meyer-Roth	3	3.565E-06	1.968E-05	3.759E-05	**1 - 0**	**1 - 0**	0 - 1	**1 - 0**	**1 - 0**	0 - 1	0 - 1	0 - 1	0 - 1
f14	Perm F1	3	4.132E-05	1.999E-02	1.776E-02	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**
f15	Neumaier's F2	4	2.824E-04	6.827E-03	8.276E-03	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	0 - 1	0 - 1	**1 - 0**
f16	Wood (or Colville)	4	6.498E-03	5.943E-01	6.678E-01	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	0 - 1	0 - 1	**1 - 0**
f17	Perm F2	6	1.441E-03	8.903E-02	1.123E-01	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	0 - 1	**1 - 0**	**1 - 0**
f18	Price's Transistor	9	5.993E+00	5.986E+01	3.587E+01	**1 - 0**	0 - 1	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	0 - 1	0 - 1	0 - 1
f19	Storn's Chebyshev	9	6.051E+02	1.235E+04	1.015E+04	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	0 - 1	**1 - 0**	0 - 1	**1 - 0**
f20	Epistatic Michalewicz	10	2.198E-01	6.129E-01	2.995E-01	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	0 - 1	0 - 1	0 - 1
f21	Neumaier's F3 (or Trid)	15	2.185E+01	2.694E+01	5.093E+00	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**
f22	Normalized Rana	15	3.333E+01	4.036E+01	7.025E+00	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	0 - 1	**1 - 0**	0 - 1
f23	Bent Cigar	17	2.761E+04	1.922E+05	1.755E+05	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	0 - 1	0 - 1	0 - 1
f24	Alpine	20	2.962E-03	9.839E-03	4.299E-03	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	0 - 1	0 - 1	0 - 1
f25	Ackley	30	1.402E-01	2.960E-01	1.140E-01	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	0 - 1	0 - 1	0 - 1
f26	Gen. Griewank	30	1.452E-01	5.156E-01	1.686E-01	**1 - 0**	**1 - 0**	0 - 1	**1 - 0**	**1 - 0**	0 - 1	0 - 1	0 - 1	0 - 1
f27	Gen. Penalized F1	30	1.280E-05	5.979E-04	1.214E-03	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	0 - 1	0 - 1	0 - 1
f28	Gen. Penalized F2	30	1.599E-04	1.994E-02	2.763E-02	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	0 - 1	0 - 1	0 - 1
f29	Gen. Rastrigin	30	7.999E-02	3.863E-01	3.557E-01	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	0 - 1	0 - 1	**1 - 0**
f30	Gen. Rosenbrock	30	3.424E+01	1.660E+02	2.152E+02	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	0 - 1	0 - 1	**1 - 0**	**1 - 0**
f31	Gen. Schwefel's F2.26	30	4.391E-06	1.104E-05	4.431E-06	0 - 1	0 - 1	0 - 1	**1 - 0**	**1 - 0**	**1 - 0**	0 - 1	0 - 1	0 - 1
f32	Schwefel's F1.2	30	4.685E-03	1.824E+02	2.565E+02	**1 - 0**	**1 - 0**	**1 - 0**	0 - 1	0 - 1	0 - 1	0 - 1	0 - 1	0 - 1
f33	Schwefel's F2.21	30	1.454E-01	5.229E-01	4.480E-01	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	0 - 1	0 - 1	0 - 1
f34	Schwefel's F2.22	30	1.455E-01	2.650E-01	7.022E-02	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	0 - 1	0 - 1	0 - 1
f35	Sphere Model	30	2.059E-01	7.087E-01	3.958E-01	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	**1 - 0**	0 - 1	0 - 1	0 - 1
f36	Step	30	0.000E+00	1.133E+00	1.310E+00	1 - 1	**1 - 0**	0 - 1	**1 - 0**	**1 - 0**	**1 - 0**	1 - 1	0 - 1	0 - 1
Total Winnings out of 36 Benchmark Functions						**33 - 4**	**29 - 7**	**27 - 9**	**29 - 7**	**30 - 6**	**27 - 9**	5 - 32	5 - 31	9 - 27



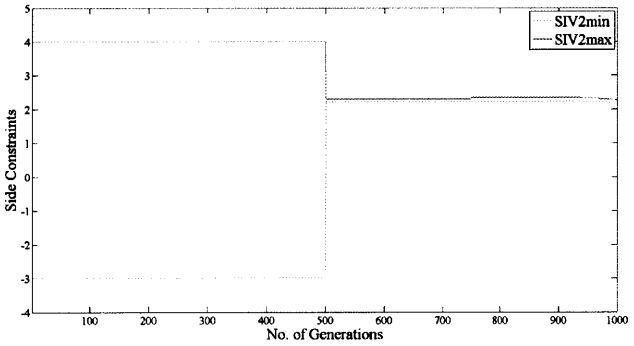
a



b



c



d

Figure (4.16)
The actual updating manner of iterative and adaptive models on SIV_2 of Perm No.1 function. (a) BSU Model, (b) BMU Model, (c) ISS Model and (d) AVB Model

Chapter 5

Optimal Coordination of DOCRs Using BBO

5.1 The Proposed Modifications

5.1.1 Constraint-Handling Techniques

5.1.2 Avoiding Infeasibility Condition

5.1.3 Hybridization with Linear Programming

5.2 Performance Comparison and Discussions

5.2.1 Test Case 1: 3-Bus System

5.2.2 Test Case 2: 4-Bus System with Normal Configuration

5.2.3 Test Case 3: 8-Bus System with Normal Configuration

5.2.4 Test Case 4: 9-Bus System with Normal Configuration

5.2.5 Test Case 5: 15-Bus System

Chapter 5
Optimal Coordination of DOCRs Using BBO

Many variations have been proposed in the literature to the basic BBO algorithm. From the preceding studies reported in Chapter 4 on BBO, the performance evaluation among the four original forms of the BBO algorithm found that the first presented model in (Simon, 2008a), which is known as the partial migration based BBO (PMB-BBO), gives the best results as the complexity, side constraints and/or dimensions of a given problem increases and as the number of islands or population size decreases. However, it is the slowest algorithm which requires around 42.30% more CPU time than that of the fastest algorithm. By solving the associated weakness in its migration and mutation sub-algorithms, the enhanced BBO version (which is called BBO-EM) can save about 32.32% of its total CPU time with better performance. Thus, BBO-EM is used here as a basis to solve the optimal coordination problem of DOCRs.

5.1 The Proposed Modifications

All the presented BBO algorithms in Chapter 4 can only solve unconstrained problems, while the coordination problem is considered as a constrained problem. In addition, the conventional EAs are slow and can make big issue in the protective designs, where the processing time is very important factor. Moreover, obtaining a feasible solution is not an easy task because the coordination problem is highly nonconvex due to

its large number of constraints and limited number of discrete PS. Based on that, the following Sub-Sections will list some proposed modifications that can overcome all the previous stated problems:

5.1.1 Constraint-Handling Techniques

Various techniques have been proposed to handle and satisfy the constraints of the coordination problem, such as **Karush-Kuhn-Tucker (KKT)** (Urdaneta et al., 1988) and death penalty function (Bedekar & Bhide, 2011). In addition, the constraint-handling part of the overall optimal coordination program is either not mentioned at all in some papers, or insufficient details are given such as the unknown exterior penalty function that was reported in (Noghabi et al., 2009; Amraee, 2012). Therefore, it is important to carry further investigation on this issue, because it directly affects the performance of the optimization algorithm in terms of the overall cost and convergence speed (CPU time) (Eiben & Smith, 2003). The ten constraint-handling techniques that have been presented in Section 2.3 will be selected here for carrying this investigation.

5.1.2 Avoiding Infeasibility Condition

It is very hard to find an optimal and feasible solution of the coordination problem, because it is considered as a highly constrained **mixed integer non-linear programming** (**MINLP**) problem. Some researchers got good results but with some few constraint violations, as in (Noghabi et al., 2009; Bedekar & Bhide, 2011).

The nature of this infeasibility condition comes from the complexity of the network or/and the lack of diversity of PS values. If the network is large and PS is provided with limited number of discrete values, then finding a solution that satisfies all the selectivity constraints of P/B DOCR pairs becomes very difficult and requires high number of generations. This motivated the researchers to study the sources of that problem in order to overcome the infeasibility issue.

In (Karegar, Abyaneh, Ohis, & Meshkin, 2005), the selectivity constraints are extensively investigated, and the infeasibility range of each P/B relay pair is identified by dividing the overall range into four categories: non-valid, pre-obtained, redundant and valid, as shown in Fig.(5.1). Based on that, the total number of constraints reduced and hence the chance to get feasible solutions is increased. In addition, when an infeasible solution is detected, a search for a feasible solution is carried out through a while loop unit, in which either the relays' TCCCs or their PSs are changed randomly until a feasible solution is found. However, this technique cannot be applied in this thesis because it is assumed that all the DOCRs are modeled with only IDMT characteristic curve, while the previous technique uses different TCCCs in order to find a feasible P/B relay pair. Although, there are other proposed techniques, such as using complicated equations (Ezzeddine & Kaczmarek, 2011a), they are either hard to be implemented or failed to reach a feasible solution.

Consequently, this Sub-Section proposes some enhancements to the technique proposed in (Karegar et al., 2005). In this thesis, both PS

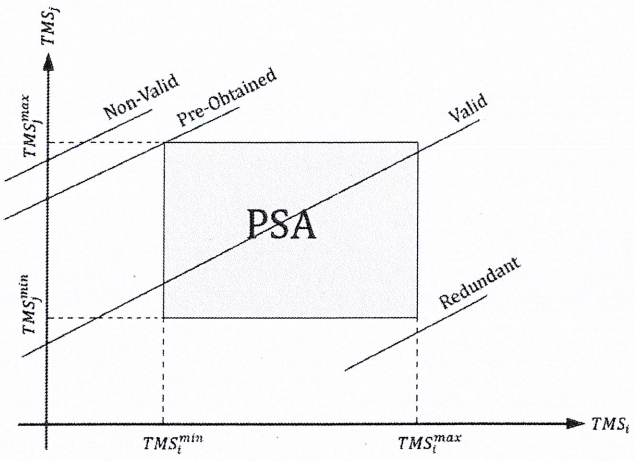


Figure (5.1)
Four categories of selectivity constraint lines

and TMS are randomly generated through the while loop unit, which is shown in Fig.(5.2) as the pre-processing unit. The proposed pre-processing unit is incorporated within the conventional BBO (*it is also incorporated within the hybrid BBO-LP as will be seen later*) to solve very complex and nonconvex problems, and it gives feasible solutions even within few generations which was not achieved with the conventional GA in (Noghabi et al., 2009; Bedekar & Bhide, 2011).

5.1.3 Hybridization with Linear Programming

Unlike gradient-based algorithms, EAs are probabilistic-based multipoint search techniques, the fact that makes them very slow convergent and time consuming algorithms. Thus, many researchers prefer to use only LP and NLP techniques, while the others hybridized them with EAs (Noghabi et al., 2009; Bedekar & Bhide, 2011).

In order to accelerate the convergence speed and accuracy, a hybrid

BBO with LP is proposed in this thesis as shown in Fig.(5.2). LP is selected instead of NLP for the sake of speed and simplicity.

The algorithm of LP can be constructed by considering the objective function of DOCRs coordination problem given by Eq.(3.10) or Eq.(3.11), and the corresponding constraints given by Eqs.(3.12)-(3.17). First, the BBO algorithm is executed normally, and the fitness per generation is selected. The obtained TMS values are set as initial points to the LP, while PS values are set as predetermined setting (i.e., constants), in order to linearize Eqs.(3.8)-(3.13).

The overall flowchart of the proposed BBO-LP algorithm is shown in Fig.(5.2), which includes all the preceding proposed modifications. It should be noted that the clear duplication process covers only TMS (if PS is discrete) in order to avoid infinite loops.

5.2 Performance Comparison and Discussions

The performance of the proposed BBO algorithm with/without hybridization is evaluated for five test systems with different scenarios and complexities. The algorithm's parameters for each system are listed in Table 5.1.

For fair comparison with work done in (Noghabi et al., 2009; Amraee, 2012), the algorithms are coded using MATLAB R2011a and the simulation is carried out using Pentium IV, 3.06 GHz and 512 MB RAM with 32 bit Windows XP SP3 operating system.

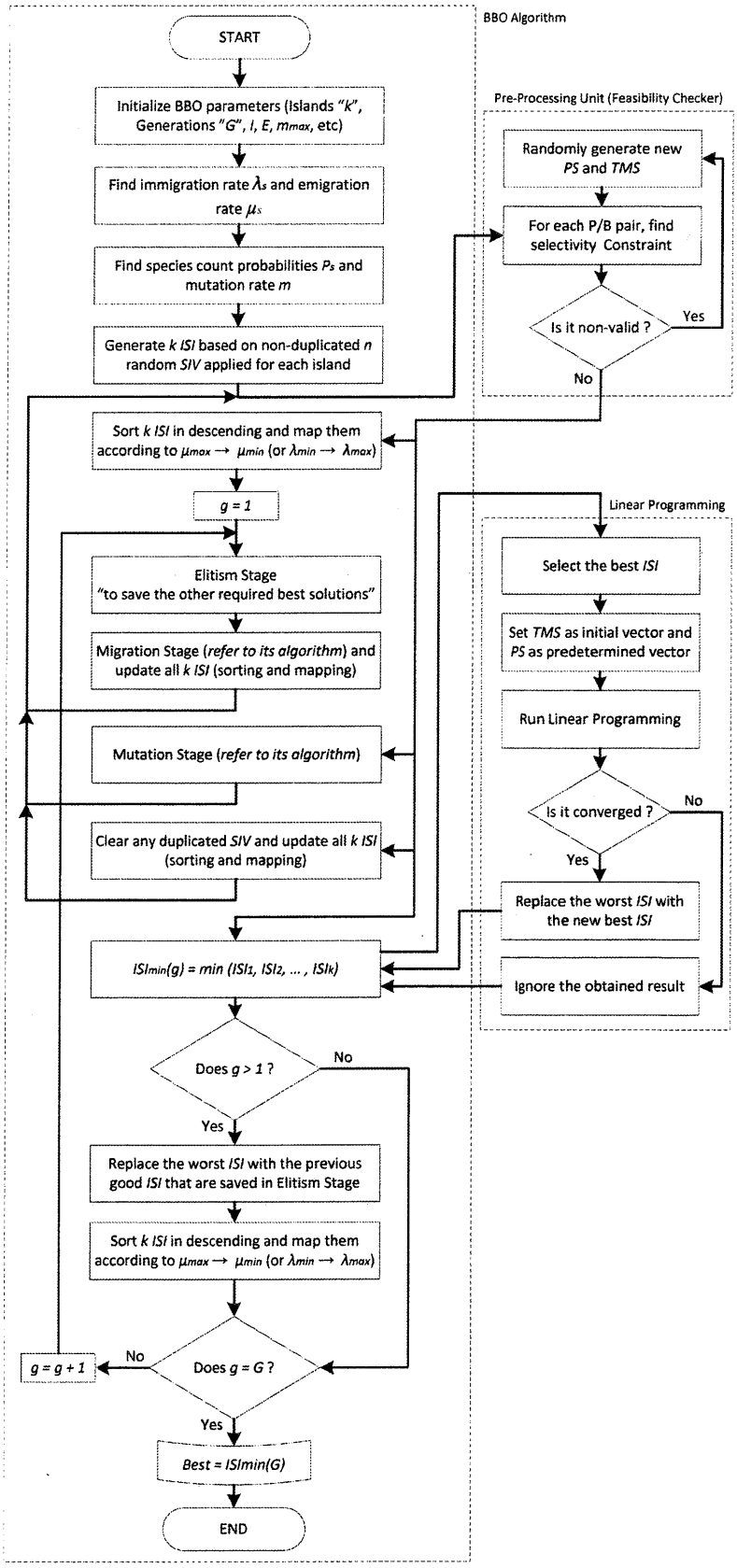


Figure (5.2)
Flowchart of the proposed BBO-LP algorithm

Table (5.1)
The BBO parameters for each test case (Amraee, 2012; Thangaraj et al., 2010; Noghabi et al., 2009; Bedekar & Bhide, 2011)

Parameters	3-Bus	4-Bus	8-Bus	9-Bus	15-Bus
TMS^{min}	0.1	0.05	0.1	0.025	> 0.1
TMS^{max}	1.1	1.1	1.1	1.2	1.1
PS^{min} (A)	1.5	1.25	0.5	Eq. (3.18)	0.5
PS^{max} (A)	5.0	1.5	2.5	Eq. (3.19)	2.5
PS Mode	discrete	continuous	discrete	continuous	discrete
CTI (s)	0.2	0.3	0.3	0.2	0.2
m_{max}	0.05	0.05	0.05	0.1	0.05

5.2.1 Test Case 1: 3-Bus System

The 3-bus system is shown in Fig.(5.3). The $CTRs$ for the relays R_1 to R_6, the listed P/B pairs, the corresponding currents for 3ϕ fault at the middle of each line, and the rest of information all are given in Appendix D. The PS is considered discrete in uniform steps of 0.5 A.

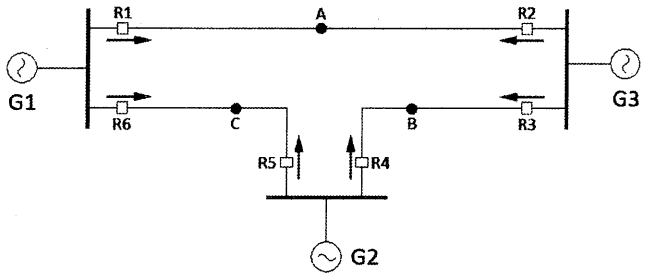


Figure (5.3)
Single-line diagram of the 3-bus system for test case 1

5.2.1.1 Scenario 1.1: Normal Configuration with BBO

The coordination problem of the 3-bus system with fixed topology and six IDMT-based DOCRs will have 12 variables and 30 constraints. By using the expression of Eq.(2.2), Eq.(3.11) can be formulated into a mathematical model as:

$$\min_{TMS,PS} Z(TMS_1, \cdots, TMS_6, PS_1, \cdots, PS_6)$$

$$\begin{aligned} \text{Subjected to: } T_{pr,i} + CTI - T_{bc,j} &\leqslant 0 \\ T_i^{min} - T_{pr,i} &\leqslant 0 \\ T_{pr,i} - T_i^{max} &\leqslant 0 \end{aligned}$$

$$\begin{aligned} TMS_i^{min} \leqslant TMS_i \leqslant TMS_i^{max} \\ PS_i^{min} \leqslant PS_i \leqslant PS_i^{max} \end{aligned} \tag{5.1}$$

where i and j notations are respectively represent the primary and backup relays (*i.e., P/B pairs*), which are given in Appendix D.

The effect of the ten constraint-handling techniques on the performance of the conventional BBO is evaluated based on the population size of 50, generation number of 85 and 30 Monte-Carlo simulations as shown in Table 5.2, where "Violations" stands for ($T_{bc} - T_{pr} - CTI \geqslant 0$) and "Time" stands for the relays' operating time when they operate as primary relays. It can be clearly seen that the binary static penalty provides the second best result after the self-adaptive penalty, but it is faster than the last one as will be seen in the next scenario. It is designed to have only one element of the penalty multiplier vector, which can be tuned easily and quickly. In order to select one penalty method that balances between the simplicity, quality and CPU time, further investigations will be carried out in the next scenario.

5.2.1.2 Scenario 1.2: Transient Configuration with BBO/BBO-LP

The data of this scenario is similar to the previous one, except that both the transient configurations and the normal configuration are considered. The transient configurations happen if only one primary relay is operated during the fault isolation process (Urdaneta et al., 1988). Thus, the transient selectivity constraints of Eq.(3.14) have to be considered too. This will duplicate the number of selectivity constraints in Eq.(5.1), which in turns make the coordination problem more difficult to be optimized.

The effect of the previous ten constraint-handling techniques on the performance of the conventional BBO is evaluated based on 50 Monte-Carlo simulations as shown in Table 5.3, where "Norm Viol" stands for normal violations ($T_{bc} - T_{pr} - CTI \geqslant 0$) and "Trans Viol" stands for transient violations ($T'_{bc} - T'_{pr} - CTI \geqslant 0$). Also, Table 5.4 gives additional information about these techniques, where the "Best" and "Worst" stand for the best and the worst solutions, the "Mean" and "StdDev" stand for the mean and the standard deviation of all the 50 trials, and the "nCPU" stands for the average normalized CPU time that is calculated by dividing all the actual CPU times by the lowest obtained CPU time (*which is the CPU time of the exponential dynamic penalty function*). It can be clearly seen that the binary static penalty outperforms the others, and achieved the third fastest method. Adaptive and self-adaptive penalties still show competitive results as in Table 5.2. However, their speed

performance, which is very important factor in protective relaying, is not good. The same performance results are obtained for other simulated larger systems. Thus, the binary static penalty is chosen for solving the other case studies. Table 5.5 gives the singular value of the penalty multiplier, which are obtained by trail and error tuning for each case study.

As stated before in Sub-Section 5.1.3, EAs are slow convergent techniques, and hence the conventional BBO needs high number of generations (Noghabi et al., 2009; Bedekar & Bhide, 2011). As a result, the hybrid BBO-LP algorithm is proposed. Table 5.6 shows that the performance of the proposed BBO-LP is better than the performance of the conventional BBO with respect to all of the performance criteria. Similarly, the proposed BBO-LP achieved better performance as compared to the Seeker algorithm proposed in (Amraee, 2012) as shown in Table 5.7. In addition, it can be clearly seen from Table 5.7 that the BBO-LP has similar PS and more tuned TMS than the Seeker algorithm, and consequently acheived a better feasible solution.

5.2.1.3 Scenario 1.3: Transient Configurations with ISS-BBO

In this scenario, the new proposed ISS model that is presented in Chapter 4 will be used here to solve the 3-bus system with the transient configurations. It is important to mention that the iterative process will be done only on the upper bounds (*i.e.,* TMS^{max} *and* PS^{max}), because $T \downarrow \propto TMS \downarrow \ \& \ PS \downarrow$. The tuning parameters are taken as; $Elit = 4$, $R = 50\%$, $SF = 98\%$ and $MGN = 3$. Also, because PS is discrete, its

upper bound will decrease by steps if there is any update (*see Algorithm 10 in Chapter 4*).

Table (5.2)
Comparison of the cost functions of the 10 constraint-handling techniques for the test case 1 - scenario 1

Relay No.	Random Search "Direct Method"			
	TMS	PS	Violations	Time
1	0.10503456	2.5	0.22385380	0.27772572
2	0.10226221	1.5	0.01658281	0.21413812
3	0.11388465	2	0.05868374	0.25375462
4	0.11283684	1.5	0.14811266	0.25509389
5	0.10844235	1.5	0.07195008	0.22833366
6	0.11854236	1.5	0.30161369	0.30034841
Total Operating Time: 1.52939441616869 s				

Relay No.	Constant Penalty			
	TMS	PS	Violations	Time
1	0.14587203	1.5	0.24145327	0.32029654
2	0.10377297	1.5	0.00381712	0.21730167
3	0.11828675	2	0.05662043	0.26356325
4	0.11032135	1.5	0.07906559	0.24940703
5	0.1177428	1.5	0.07170523	0.24791647
6	0.10385562	1.5	0.35067289	0.26313691
Total Operating Time: 1.56162186789885 s				

Relay No.	Binary Static Penalty			
	TMS	PS	Violations	Time
1	0.1344352	1.5	0.16745708	0.29518427
2	0.10330469	1.5	0.06127103	0.21632109
3	0.10135841	2.5	0.03515852	0.24424111
4	0.10482652	2	0.05649279	0.2628434
5	0.10242372	1.5	0.10460644	0.21566097
6	0.10206013	1.5	0.35154976	0.25858771
Total Operating Time: 1.4928385528808 s				

Relay No.	The Superiority of Feasible Points - Type I			
	TMS	PS	Violations	Time
1	0.11202541	3	0.12924215	0.31932426
2	0.11167353	1.5	0.02850729	0.23384553
3	0.112681	2.5	0.15703474	0.27152491
4	0.10148165	2	0.15362085	0.25445643
5	0.10024817	1.5	0.16730551	0.21108019
6	0.11949956	1.5	0.37299405	0.30277364
Total Operating Time: 1.5930049616981 s				

Relay No.	The Superiority of Feasible Points - Type II			
	TMS	PS	Violations	Time
1	0.11475844	2	0.38559059	0.27865096
2	0.10373202	1.5	0.04516652	0.21721592
3	0.1133702	2	0.02990281	0.25260834
4	0.12113127	1.5	0.06791851	0.27384537
5	0.13358484	1.5	0.01675074	0.28127308
6	0.10646763	1.5	0.34373377	0.26975489
Total Operating Time: 1.5733485665321 s				

Relay No.	The Eclectic Evolutionary Algorithm			
	TMS	PS	Violations	Time
1	0.13257812	2	0.20297821	0.32191986
2	0.10738799	1.5	0.04872794	0.22487155
3	0.10120884	2.5	0.11355474	0.24388071
4	0.10395018	2	0.19715588	0.26064605
5	0.11204667	1.5	0.08357688	0.23592283
6	0.12927147	1.5	0.31462699	0.32753253
Total Operating Time: 1.6147735277978 s				

Relay No.	Typical Dynamic Penalty			
	TMS	PS	Violations	Time
1	0.12911381	1.5	0.78397086	0.2834999
2	0.10637369	1.5	0.01847025	0.2227476
3	0.10781232	2	0.02019897	0.24022443
4	0.11558674	1.5	0.09023022	0.26131067
5	0.10189726	2.5	0.01726223	0.25634639
6	0.10838901	1.5	0.35958214	0.27462306
Total Operating Time: 1.53875205265928 s				

Relay No.	Exponential Dynamic Penalty			
	TMS	PS	Violations	Time
1	0.12856798	1.5	0.18313621	0.28230141
2	0.1070644	1.5	0.09888867	0.22419395
3	0.10125142	2	0.03287135	0.22560561
4	0.13703302	1.5	0.10123045	0.30979497
5	0.10285594	1.5	0.02821622	0.21657104
6	0.12007893	1.5	0.33538029	0.30424158
Total Operating Time: 1.56270855261936 s				

Relay No.	Modified Adaptive Multiplication Penalty			
	TMS	PS	Violations	Time
1	0.11132616	2.5	0.19502540	0.29436156
2	0.10268099	1.5	0.04196157	0.21501504
3	0.11605168	1.5	0.10754532	0.23558822
4	0.11971509	1.5	0.06416459	0.27064378
5	0.10655776	1.5	0.00518578	0.22436551
6	0.10510073	1.5	0.33895463	0.26629161
Total Operating Time: 1.50626572547173 s				

Relay No.	Self-Adaptive Penalty			
	TMS	PS	Violations	Time
1	0.10234653	2	0.22469496	0.24851297
2	0.11667454	1.5	0.05460052	0.24431768
3	0.10179488	2	0.00350777	0.22681653
4	0.10950737	2	0.05688289	0.27458024
5	0.10405699	1.5	0.02807467	0.21909994
6	0.10444332	1.5	0.45036068	0.26462595
Total Operating Time: 1.47795329920184 s				

Table (5.3)
Comparison of the cost functions of the 10 constraint-handling techniques for the test case 1 - scenario 2

Relay No.	Random Search "Direct Method"				
	TMS	PS	Norm Viol	Trans Viol	Time
1	0.13782	2	1.33782	0.05216	0.33465
2	0.10671	2.5	0.06316	0.00312	0.26672
3	0.12792	2.5	0.07123	0.01085	0.30824
4	0.13881	1.5	0.17682	0.03359	0.31382
5	0.10131	3	0.18284	0.04436	0.27376
6	0.10664	2	1.48688	0.05412	0.30345
Total Operating Time: 1.80063789318958 s					

Relay No.	Constant Penalty				
	TMS	PS	Norm Viol	Trans Viol	Time
1	0.10455	3	0.56474	0.01995	0.29802
2	0.14846	1.5	0.09617	0.00190	0.31087
3	0.10081	3.5	0.10988	0.03017	0.27673
4	0.11334	2.5	0.24286	0.08715	0.31031
5	0.11978	2	0.21461	0.03101	0.27776
6	0.11630	2	0.63329	0.02799	0.33092
Total Operating Time: 1.80460605178158 s					

Relay No.	Binary Static Penalty				
	TMS	PS	Norm Viol	Trans Viol	Time
1	0.10433	3	0.75593	0.00964	0.29738
2	0.11279	2	0.21512	0.09437	0.25999
3	0.10077	3	0.12531	0.04509	0.26005
4	0.10804	3	0.14742	0.01094	0.31972
5	0.10076	2.5	0.14591	0.00381	0.25348
6	0.10302	2	0.82143	0.01118	0.29313
Total Operating Time: 1.68375366938206 s					

Relay No.	The Superiority of Feasible Points - Type I				
	TMS	PS	Norm Viol	Trans Viol	Time
1	0.11674	3	1.34487	0.05653	0.33276
2	0.10072	2.5	0.07199	0.00196	0.25174
3	0.10026	4	0.16380	0.07476	0.29121
4	0.11495	2	0.16068	0.02649	0.28824
5	0.10159	3	0.32013	0.08648	0.27452
6	0.10020	2	1.39345	0.04112	0.28512
Total Operating Time: 1.72358543660502 s					

Relay No.	The Superiority of Feasible Points - Type II				
	TMS	PS	Norm Viol	Trans Viol	Time
1	0.11656	2.5	0.44287	0.02921	0.30820
2	0.14360	1.5	0.15878	0.04127	0.30071
3	0.10142	3.5	0.09026	0.02244	0.27840
4	0.10554	3	0.17550	0.03439	0.31232
5	0.14701	1.5	0.18701	0.00237	0.30953
6	0.10621	2	0.62395	0.03806	0.30221
Total Operating Time: 1.8113627304257 s					

Relay No.	The Eclectic Evolutionary Algorithm				
	TMS	PS	Norm Viol	Trans Viol	Time
1	0.10795	3	0.44184	0.02925	0.30771
2	0.11691	2	0.08273	0.00855	0.26948
3	0.12195	3	0.09099	0.00894	0.31471
4	0.12121	2	0.14703	0.01270	0.30391
5	0.14677	1.5	0.21634	0.04342	0.30904
6	0.10051	2	0.87653	0.03634	0.28601
Total Operating Time: 1.79086442124039 s					

Relay No.	Typical Dynamic Penalty				
	TMS	PS	Norm Viol	Trans Viol	Time
1	0.12061	2.5	0.92318	0.06360	0.31890
2	0.13741	1.5	0.09911	0.00773	0.28773
3	0.10147	3.5	0.10987	0.03948	0.27854
4	0.10957	2.5	0.19314	0.05015	0.29999
5	0.11594	2.5	0.20523	0.02112	0.29166
6	0.10703	2	0.57304	0.01274	0.30454
Total Operating Time: 1.78137295960041 s					

Relay No.	Exponential Dynamic Penalty				
	TMS	PS	Norm Viol	Trans Viol	Time
1	0.10538	3	0.46524	0.04541	0.30039
2	0.10077	3	0.07717	0.00453	0.27043
3	0.10369	3.5	0.10667	0.02645	0.28462
4	0.12019	2	0.14998	0.03969	0.30137
5	0.14926	1.5	0.19784	0.00904	0.31427
6	0.12800	1.5	3.15064	0.08564	0.32432
Total Operating Time: 1.79540469824163 s					

Relay No.	Modified Adaptive Multiplication Penalty				
	TMS	PS	Norm Viol	Trans Viol	Time
1	0.10111	3	0.76526	0.01870	0.28820
2	0.12009	2	0.09955	0.02203	0.27682
3	0.10282	3	0.10194	0.02447	0.26535
4	0.12651	2	0.15574	0.01693	0.31721
5	0.10077	2.5	0.15810	0.01299	0.25351
6	0.10391	2	0.90395	0.04093	0.29568
Total Operating Time: 1.69676105703778 s					

Relay No.	Self-Adaptive Penalty				
	TMS	PS	Norm Viol	Trans Viol	Time
1	0.11533	2.5	0.77951	0.01427	0.30494
2	0.10210	2.5	0.06558	0.00642	0.25520
3	0.10526	3	0.09099	0.02346	0.27166
4	0.13643	1.5	0.15757	0.01955	0.30843
5	0.10326	2.5	0.16636	0.01781	0.25978
6	0.10284	2	1.41175	0.04092	0.29262
Total Operating Time: 1.69262888130894 s					

Table (5.4)
Performance criteria of the 10 constraint-handling techniques for the test case 1 - scenario 2

Constraint-Handling Type	Performance Criteria				
	Best (s)	**Worst (s)**	**Mean (s)**	**StdDev (s)**	**nCPU**
Random Search	1.80064	2.30832	2.02439	0.12475	9.76674
Constant	1.80461	2.15603	1.95493	0.09632	1.01090
Binary Static	**1.68375**	**1.99249**	**1.80228**	**0.06996**	1.19760
SFP - Type I	1.72359	2.24728	2.00115	0.13033	1.65472
SFP - Type II	1.81136	2.25699	2.00936	0.10646	1.68448
Eclectic EA	1.79086	2.34414	2.05208	0.11947	1.26876
Typical Dynnamic	1.78137	2.22096	1.96757	0.10735	1.21053
Exponential Dynamic	1.79540	2.29990	1.99988	0.11371	**1.00000**
Adaptive Multiplication	1.69676	2.18442	1.88424	0.09899	1.65089
Self-Adaptive	1.69263	2.17629	1.87734	0.11186	1.73128

Table (5.5)
The value of the binary static penalty multiplier for each network system

System	**3-Bus**	**4-Bus**	**8-Bus**	**9-Bus**	**15-Bus**
r	0.15	0.35	10	10	10

Table (5.6)
Comparison between BBO and BBO-LP for the test case 1 - scenario 2

Algorithm Type	Performance Criteria				
	Best (s)	**Worst (s)**	**Mean (s)**	**StdDev (s)**	**nCPU**
BBO	1.68375	1.99249	1.80228	0.06996	1.19760
BBO-LP	**1.59871**	**1.74746**	**1.66487**	**0.03601**	**0.21688**

Table (5.7)
Comparison with Seeker algorithm for the test case 1 - scenario 2

Primary Relay	**Seeker** (Amraee, 2012)		**BBO**		**BBO-LP**	
	PS	TMS	PS	TMS	PS	TMS
R_1	2.5	0.1070	3.0	0.1043	2.5	0.1067
R_2	2.0	0.1080	2.0	0.1128	2.0	0.1083
R_3	3.0	0.1000	3.0	0.1008	3.0	0.1000
R_4	2.5	0.1000	3.0	0.1080	2.5	0.1000
R_5	2.5	0.1000	2.5	0.1008	2.5	0.1000
R_6	1.5	0.1120	2.0	0.1030	1.5	0.1119
Pop. #	50		50		**20**	
Gen. #	85		1000		**20**	
Best (s)	1.599		1.68375		**1.59871**	
CPU (s)	10.45		16.23		**2.99**	

Table (5.8)
Comparison between BBO and ISS-BBO for the test case 1 - scenario 3

Primary Relay	BBO		ISS-BBO*		ISS-BBO**	
	PS	TMS	PS	TMS	PS	TMS
R_1	3.0	0.1043	2.5	0.1067	3.0	0.1020
R_2	2.0	0.1128	2.5	0.1000	2.5	0.1001
R_3	3.0	0.1008	3.0	0.1000	3.0	0.1017
R_4	3.0	0.1080	2.5	0.1000	2.5	0.1033
R_5	2.5	0.1008	2.5	0.1000	2.5	0.1006
R_6	2.0	0.1030	2.0	0.1000	2.0	0.1015
Pop. #	**50**		**50**		**50**	
Gen. #	1000		*1000		**85	
Best (s)	1.6838		**1.6002**		1.6284	
CPU (s)	16.23		17.16		**1.69**	

Table 5.8 shows the performance comparison between BBO and ISS-BBO. For the same number of generations, BBO is slightly faster, which is already mentioned in the performance evaluation of ISS-BBO in Chapter 4. However, ISS-BBO outperforms BBO in terms of solution quality. Moreover, ISS-BBO can perform better than BBO even with very few number of generations; so, it will save high amount of CPU time, and will be faster than BBO-LP by 56.52%.

5.2.2 Test Case 2: 4-Bus System with Normal Configuration

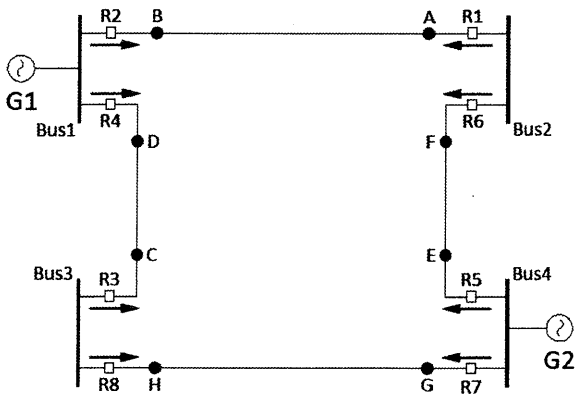


Figure (5.4)
Single-line diagram of the 4-bus system for test case 2

In this case study, both near-end and far-end 3ϕ fault locations are considered. The network consists of 4 buses, 4 branches and 8 DOCRs, as shown in Fig.(5.4). In this system, both PS and TMS are considered continuous. The $CTRs$ for the relays R_1 to R_8, the listed P/B relay pairs, and the corresponding 3ϕ fault currents are given in Appendix D. Thus, the expression in Eq.(3.10) is used here as the objective function, where each term of the equation is calculated as (Deep, Birla, Maheshwari, Gupta, & Thakur, 2006; Deep & Bansal, 2009):

$$T_{pr,p}^{near} = TMS_p \left[\frac{\beta}{\left(\frac{a_p}{PS_p \times b_p} \right)^{\alpha} - 1} \right] \tag{5.2}$$

$$T_{pr,q}^{far} = TMS_q \left[\frac{\beta}{\left(\frac{c_p}{PS_q \times d_p} \right)^{\alpha} - 1} \right] \tag{5.3}$$

where $\{a, c\}$ and $\{b, d\}$ are the normalized fault currents and $CTRs$ of the primary relays for the near-end and far-end fault locations, respectively. Note that the constants of $T_{pr,p}^{far}$ in Eq.(5.3) are calculated in sequence of p instead of q. Also, these p and q notations are i and j in (Deep et al., 2006; Deep & Bansal, 2009); this replacement is essential to prevent any confusing with other systems, because in this thesis the notations i and j are assigned for primary and backup relays, respectively.

The number of selectivity constraints depends on the network, where only 9 selectivity constraints out of 16 are considered. This will result in

a total number of 33 constraints, which are given in Appendix D (Deep et al., 2006; Deep & Bansal, 2009; Thangaraj et al., 2010). To satisfy them, Eq.(3.12) is used, where Eq.(3.13) and Eq.(3.8) become:

$$T_{jk} = TMS_j \left[\frac{\beta}{\left(\frac{e_p}{PS_j \times f_p} \right)^{\alpha} - 1} \right] \tag{5.4}$$

$$T_{ik} = TMS_i \left[\frac{\beta}{\left(\frac{g_p}{PS_i \times h_p} \right)^{\alpha} - 1} \right] \tag{5.5}$$

where $\{e, g\}$ and $\{f, h\}$ are the normalized fault currents and $CTRs$ of the back-up and primary relays, respectively. Also, the side constraints T_{ik}^{min} and T_{ik}^{max} given in Eq.(3.15) are set at 0.05 and 1.0 s, respectively.

Table 5.9 shows the performance comparison between BBO and BBO-LP for this case study, while Table 5.10 shows that both algorithms provide feasible solutions without any violation on their selectivity constraints. Table 5.11 shows some obtained results in the literature (Deep et al., 2006; Deep & Bansal, 2009). From these tables, it is observed that the BBO-LP algorithm requires only 10 populations for converging to the near optimum solution, while the other algorithms require population size of 50. Also, BBO outperforms GA, and BBO-LP has the best mean. The performances of BBO and BBO-LP are evaluated using 1000 generations and 30 Monte-Carlo simulations.

Because PS is also continuous, BBO-LP may give more precise

results if PS is fine tuned by using a second LP stage or by employing non-simplified migration models of BBO. It has to be taken into account that the results in (Thangaraj et al., 2010), which are obtained by using modified differential evolution algorithms (MDEs), are slightly better, but the termination limit is set with 10000 generations (*i.e., 10 times higher than that of the other algorithms*).

Table (5.9)
Comparison between BBO and BBO-LP for the test case 2

Primary Relay	**Conventional BBO**				**Hybrid BBO-LP**			
	TMS	PS	T_{pr}^{near}	T_{pr}^{far}	TMS	PS	T_{pr}^{near}	T_{pr}^{far}
R_1	0.0537	1.2631	0.1032	0.4140	0.0500	1.2736	0.0964	0.4102
R_2	0.2257	1.2501	0.3007	0.1205	0.2123	1.4969	0.2939	0.1126
R_3	0.0500	1.2549	0.1543	0.3748	0.0500	1.2503	0.1540	0.3562
R_4	0.1698	1.2503	0.2601	0.1762	0.1516	1.4998	0.2427	0.1758
R_5	0.1439	1.2522	0.2351	0.1868	0.1264	1.4989	0.2164	0.1862
R_6	0.0501	1.2529	0.1584	0.3478	0.0500	1.2501	0.1580	0.3270
R_7	0.1494	1.2581	0.2608	0.1870	0.1339	1.4981	0.2451	0.1723
R_8	0.0540	1.2599	0.1450	0.4037	0.0500	1.2501	0.1337	0.3892
Pop. #	50				**10**			
Gen. #	**1000**				**1000**			
Best (s)	3.828242239541099				**3.669753524423449**			
Mean (s)	3.953278807337223				**3.671629037458293**			
StdDev (s)	6.3176184646692E-02				**7.541712535704323E-04**			

Table (5.10)
Feasibility comparison between BBO and BBO-LP for the test case 2

$R_{pr,i}$	$R_{pr,1}$	$R_{pr,1}$	$R_{pr,3}$	$R_{pr,3}$	$R_{pr,4}$	$R_{pr,6}$	$R_{pr,6}$	$R_{pr,8}$	$R_{pr,8}$
$R_{bc,i}$	$R_{bc,5}$	$R_{bc,5}$	$R_{bc,7}$	$R_{bc,7}$	$R_{bc,1}$	$R_{bc,2}$	$R_{bc,2}$	$R_{bc,4}$	$R_{bc,4}$
CTI (BBO)	0.313	0.406	0.311	0.356	0.315	0.347	0.301	0.393	0.306
CTI (BBO-LP)	0.300	0.400	0.300	0.350	0.300	0.349	0.300	0.398	0.300

Table (5.11)
Literature results for the test case 2 (Deep et al., 2006; Deep & Bansal, 2009)

Performance Criteria	**Algorithm Type**			
	PSO-W	**LXPSO-W**	**PSO-C**	**LXPSO-C**
Best (s)	3.670242	**3.669327**	3.677945	**3.669327**
Mean (s)	3.678875	3.675315	3.69403	**3.675038**
Previous Results:				
Performance Criteria	**Algorithm Type**			
	GA	**SOMA**	**RST**	**SOMGA**
Best (s)	3.85874	3.78922	3.705018	3.67453

5.2.3 Test Case 3: 8-Bus System with Normal Configuration

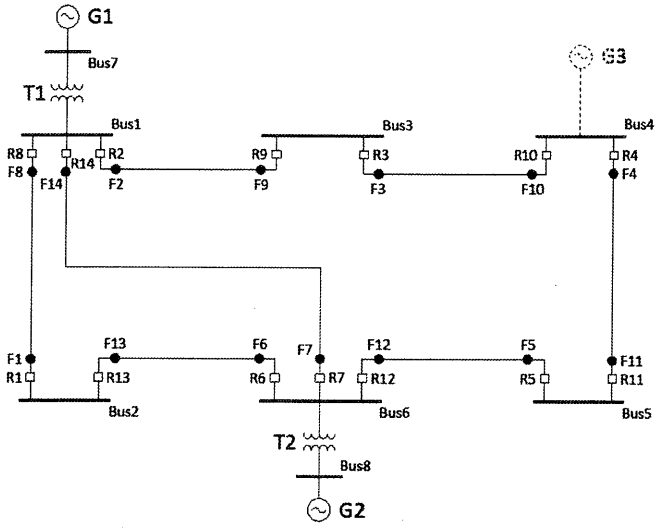


Figure (5.5)
Single-line diagram of the 8-bus system for test case 3

The network of this test case is shown in Fig.(5.5), where the near-end 3ϕ fault is considered. The $CTRs$ and the short-circuit current for each P/B pair of relays are given in Appendix D, and the available range of the discretized PS is {0.5,0.6,0.8,1.0,1.5,2,2.5} (Noghabi et al., 2009; Zeineldin, El-Saadany, & Salama, 2006).

The performance comparison between BBO and BBO-LP is shown in Table 5.12, and the comparison between BBO with GA, and BBO-LP with GA-LP are shown in Table 5.13.

As mentioned in Section 5.1.2, the highly constrained network with limited number of discretized PS cannot be solved easily without solving first the associated infeasibility condition. Based on that, both BBO and BBO-LP outperform GA and GA-LP, respectively, in all criteria (feasibility, fitness and CPU time); and they require less population size and

generations for converging to a feasible near optimal solution. Furthermore, the results can be enhanced even more by allowing some tolerance

Table (5.12)
Comparison between BBO and BBO-LP for the test case 3

Primary Relay	**Conventional BBO**			**Hybrid BBO-LP**		
	TMS	PS	T_{pr}	TMS	PS	T_{pr}
R_1	0.142386	2	0.512730	0.100000	2.5	0.408734
R_2	0.381589	1.5	0.927284	0.303288	2	0.823769
R_3	0.293264	2	0.832123	0.235793	2.5	0.739040
R_4	0.220805	2	0.733329	0.220428	1.5	0.640679
R_5	0.118339	2.5	0.589118	0.100000	2.5	0.497824
R_6	0.242713	2	0.651067	0.173397	2.5	0.511005
R_7	0.327045	2	0.797126	0.253206	2.5	0.672266
R_8	0.215597	2	0.578941	0.348051	0.5	0.596303
R_9	0.229333	2	0.767412	0.148658	2.5	0.559491
R_{10}	0.314997	1.5	0.905266	0.205193	2	0.672795
R_{11}	0.278153	2	0.933151	0.191977	2.5	0.724592
R_{12}	0.369401	2	1.005074	0.271461	2.5	0.812530
R_{13}	0.103632	2.5	0.444358	0.115402	2	0.433502
R_{14}	0.357359	2	0.872493	0.249280	2.5	0.663062
Best (s)	10.549472			**8.755591**		

Table (5.13)
Feasibility comparison with other paper result (Noghabi et al., 2009) for the test case 3

P/B Relay Pairs		**Violations** ($T_{bc,j} - T_{pr,i} - CTI \geqslant 0$)			
R_i	R_j	**GA**	**GA-LP**	**BBO**	**BBO-LP**
R_1	R_6	0.0021	0.0100	0.0613	0.0000
R_2	R_1	0.1801	0.3041	0.1282	0.2504
R_2	R_7	**-0.0682**	**-0.0315**	0.0390	0.0000
R_3	R_2	0.0830	0.0301	0.0077	0.0000
R_4	R_3	0.0282	0.0237	0.0003	0.0000
R_5	R_4	**-0.0056**	0.0062	0.0556	0.0000
R_6	R_5	**-0.0301**	**-0.1203**	0.2401	0.1956
R_6	R_{14}	0.0304	0.0360	0.4393	0.3015
R_7	R_5	**-0.1311**	**-0.1752**	0.0940	0.0343
R_7	R_{13}	0.4130	0.4244	0.3531	0.1403
R_8	R_7	0.1498	0.1356	0.3873	0.2275
R_8	R_9	**-0.0990**	**-0.0069**	0.3474	0.0668
R_9	R_{10}	0.0541	**-0.0036**	0.0523	0.0000
R_{10}	R_{11}	0.0651	0.0234	0.0032	0.0000
R_{11}	R_{12}	0.0397	0.0306	0.0061	0.0000
R_{12}	R_{13}	0.3823	0.4132	0.1451	0.0000
R_{12}	R_{14}	**-0.1013**	**-0.0301**	0.0853	0.0000
R_{13}	R_8	**-0.0153**	**-0.0199**	0.0655	0.0000
R_{14}	R_1	0.2862	0.3269	0.1830	0.4111
R_{14}	R_9	**-0.2110**	**-0.1513**	0.0539	0.0000
Pop. #		100	100	50	**20**
Gen. #		100000	30	10000	**30**
Best (s)		11.0010	10.9499	10.5495	**8.7556**
CPU (s)		36000	300	2065.02	**6.79**

on the selectivity constraints to increase its search space as in (Amraee, 2012)[1]. Also, the number of generations in (Amraee, 2012) is increased more than 5 times of (Noghabi et al., 2009), which leads to better solution quality.

5.2.4 Test Case 4: 9-Bus System with Normal Configuration

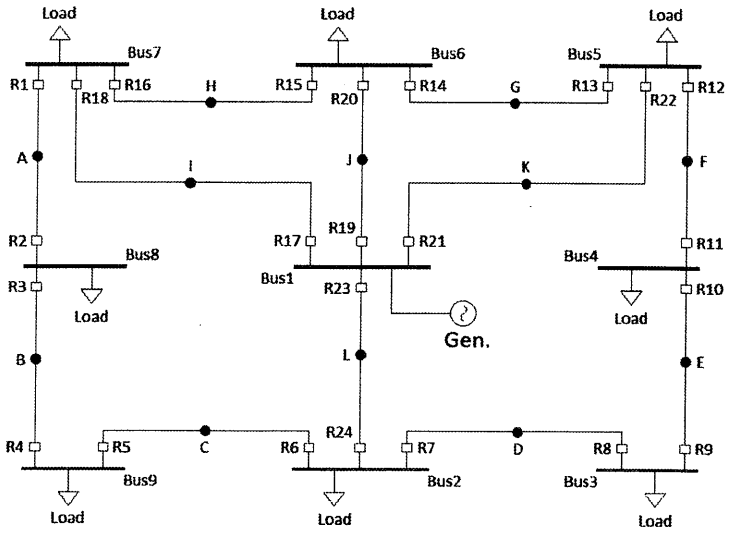


Figure (5.6)
Single-line diagram of the 9-bus system for test case 4

In this case study, 9-bus interconnected distribution system with one single end fed and equal impedances for all the lines is considered, as shown in Fig.(5.6). All the DOCRs are considered to be numerical, in which both PS and TMS are continuous. PS limits is calculated using Eqs.(3.18)-(3.19) with $OLF = 1.25$. In addition, relays $\{R_{17}, R_{19}, R_{21}, R_{23}\}$ are assumed with no backup and a 3ϕ fault at the midpoint of each line is considered. Also, the minimum operating time of each relay is taken as 0.2 s. The rest of the data of this test system can

[1]This means that the algorithm will tolerate the small violations on the selectivity constraints and accepting the infeasible solution as a feasible solution.

be found in Appendix D (Bedekar & Bhide, 2011). It can be obviously seen from Tables 5.14 and 5.15 that BBO and BBO-LP outperform GA and GA-NLP. Moreover, the proposed algorithms give optimized feasible solutions and require less number of populations. This can be graphically illustrated in Fig.(5.7) for 100 trials of simulation runs.

Although, BBO outperforms GA for that competition, it shows much less performance as compared to GA-NLP and BBO-LP. Because EAs are slow convergent techniques, BBO has the ability to give better solutions if the assigned number of generations is not restricted to 40. This can be observed in Fig.(5.8) and Table 5.16.

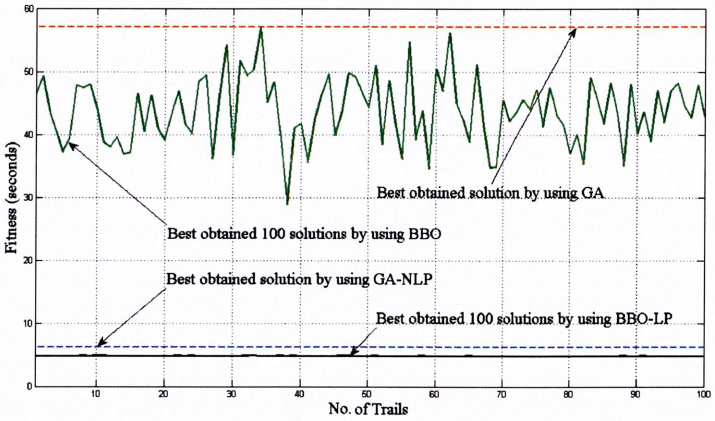


Figure (5.7)
The best obtained solutions through 100 trials for the test case 4

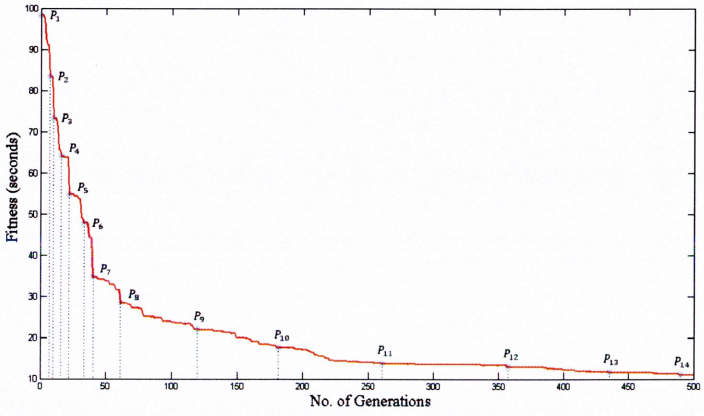


Figure (5.8)
Fitness function of the conventional BBO algorithm within 500 generations for the test case 4

Table (5.14)
Comparison between BBO and BBO-LP for the test case 4

Primary	**Conventional BBO**			**Hybrid BBO-LP**		
Relay	TMS	PS	T_{pr}	TMS	PS	T_{pr}
R_1	0.388350	1.040898	1.189412	0.067335	0.972447	0.200000
R_2	0.370953	0.821276	1.854003	0.048055	0.625116	0.200000
R_3	0.356891	0.694491	1.169716	0.049019	1.040737	0.200000
R_4	0.353097	0.930622	1.408622	0.043734	1.156009	0.200000
R_5	0.329012	0.651873	1.334616	0.041482	0.849980	0.200000
R_6	0.196642	1.150801	0.664606	0.051951	1.467938	0.200000
R_7	0.659594	0.679818	1.760774	0.061194	1.075367	0.200000
R_8	0.196997	0.748565	0.871243	0.044656	0.763080	0.200000
R_9	0.204749	0.528418	0.611494	0.050920	0.906199	0.200000
R_{10}	0.242394	0.983748	0.956479	0.042087	1.316613	0.200000
R_{11}	0.203619	0.484470	0.732427	0.039638	0.831929	0.200000
R_{12}	0.335936	0.591974	1.021274	0.055299	0.841893	0.200000
R_{13}	0.396154	0.691867	1.144718	0.065823	0.774899	0.200000
R_{14}	0.709068	0.507803	1.723927	0.060553	1.047077	0.200000
R_{15}	0.673042	0.443539	1.558759	0.049297	1.530138	0.200000
R_{16}	0.314204	1.299033	1.245330	0.065642	0.779604	0.200000
R_{17}	0.853200	1.609672	2.598978	0.064736	1.660039	0.200000
R_{18}	0.034601	1.834599	0.264669	0.030570	1.576170	0.200000
R_{19}	0.368156	1.045958	0.945288	0.069071	1.402983	0.200000
R_{20}	0.048160	2.213733	0.387168	0.042385	1.216552	0.200000
R_{21}	0.869399	1.618530	2.654954	0.075479	1.160021	0.200000
R_{22}	0.063310	1.304024	0.350627	0.039421	1.164883	0.200000
R_{23}	0.529085	1.767809	1.652733	0.070107	1.442597	0.200000
R_{24}	0.088371	1.442434	0.732952	0.027702	1.275045	0.200000
Best (s)		28.834770			**4.800000**	
Worst (s)		56.941878			**4.912645**	
Mean (s)		43.680701			**4.818836**	
StdDev (s)		5.246363			**0.019653**	

Table (5.15)
Feasibility comparison with other paper result (Bedekar & Bhide, 2011) for the test case 4

P/B Relay Pairs		**Violations** ($T_{bc,j} - T_{pr,i} - CTI \geqslant 0$)			
R_i	R_j	**GA**	**GA-NLP**	**BBO**	**BBO-LP**
R_1	R_{15}	1.7093	0.5656	1.3990	0.4117
R_1	R_{17}	6.5676	1.1322	11.1303	0.6162
R_2	R_4	0.7741	0.4457	0.9792	0.1146
R_3	R_1	8.3133	0.8908	1.4298	0.0530
R_4	R_6	7.1918	0.8733	0.1973	0.3052
R_5	R_3	5.8342	0.5143	0.5667	0.0419
R_6	R_8	1.5067	0.2967	1.2698	0.0988
R_6	R_{23}	4.4240	1.2145	7.9129	0.3822
R_7	R_5	3.4809	0.2967	0.9680	0.1611
R_7	R_{23}	4.5769	1.2145	6.8167	0.3822
R_8	R_{10}	2.4953	0.5092	0.9641	0.1474
R_9	R_7	5.3983	0.8733	2.7417	0.1154
R_{10}	R_{12}	**-50.1313**	**-4.4442**	1.2241	0.2145
R_{11}	R_9	0.9909	0.4455	0.0962	0.0234
R_{12}	R_{14}	0.7690	0.4014	1.9812	0.1238
R_{12}	R_{21}	5.1232	0.9680	11.6861	0.2532
R_{13}	R_{11}	9.0987	1.1258	0.0771	0.2112
R_{13}	R_{21}	5.5214	1.2572	11.5626	0.2532
R_{14}	R_{16}	7.3664	0.0802	2.8072	0.0675
R_{14}	R_{19}	7.9644	0.8813	0.9703	0.4162
R_{15}	R_{13}	4.9253	0.3817	0.7514	0.0659
R_{15}	R_{19}	7.1290	0.8805	1.1355	0.4162
R_{16}	R_2	0.6299	28.4813	4.1156	0.0526
R_{16}	R_{17}	6.7923	1.2623	11.0744	0.6162
R_{18}	R_2	3.4009	28.4813	5.0963	0.0526
R_{18}	R_{15}	4.7051	0.6957	2.3238	0.4117
R_{20}	R_{13}	7.2145	0.4435	1.9230	0.0659
R_{20}	R_{16}	8.8201	0.1412	4.1440	0.0675
R_{22}	R_{11}	9.9250	1.1274	0.8711	0.2112
R_{22}	R_{14}	1.9936	0.6922	2.6518	0.1238
R_{24}	R_5	5.3334	0.4109	1.9958	0.1611
R_{24}	R_8	3.5120	0.4109	1.2014	0.0988
Pop. #		256	256	100	**50**
Gen. #		**40**	**40**	**40**	**40**
Best (s)		57.0406	6.1786	28.8348	**4.8000**

Table (5.16)
BBO performance within 500 generations for the test case 4

Point	1	2	3	4	5	6	7
Gen. #	1	7	10	16	22	34	41
Fitness (s)	98.37	83.31	73.12	64.03	54.79	47.84	34.53
Continue:							
Point	8	9	10	11	12	13	14
Gen. #	61	120	181	261	358	435	490
Fitness (s)	28.30	21.93	17.55	13.60	12.88	11.58	10.91

5.2.5 Test Case 5: 15-Bus System

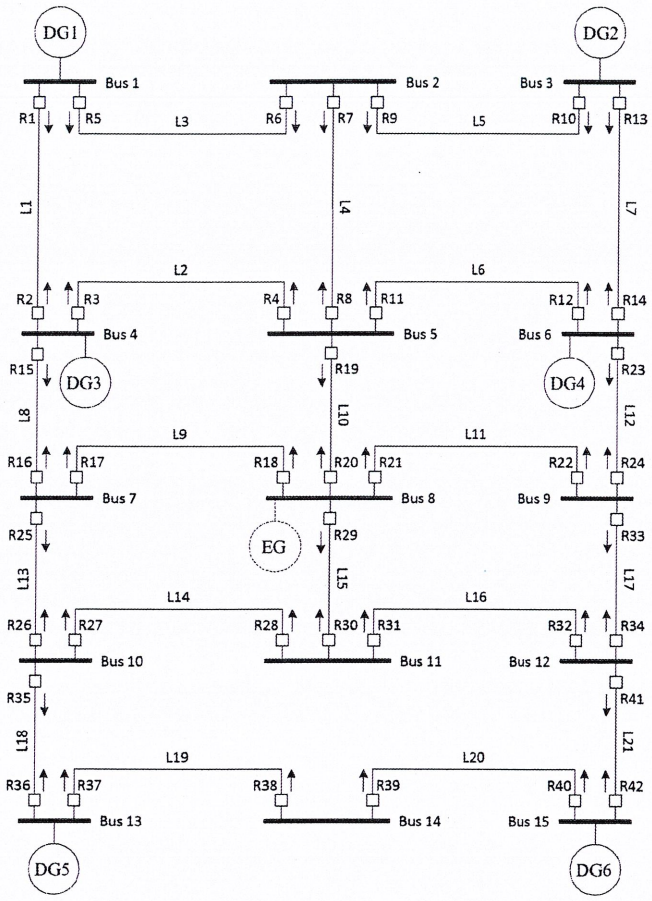


Figure (5.9)
Single-line diagram of the 15-bus system for test case 5

The case study in Fig.(5.9) is taken from (Amraee, 2012) as an example of highly distributed generation (DG) penetrated distribution networks. This system consists of 15 buses and 21 branches, and hence it has 42 DOCRs and 82 P/B relay pairs with 84 variables (*if IDMT-based DOCRs are used*). By referring to Eq.(5.1), the total constraints are 250, and addressed as: 82 inequality constraints for P/B selectivity criteria, 42 inequality constraints for minimum allowable operating time, 42 inequality constraints for maximum allowable operating time, 42 side constraints for TMS and 42 side constraints for PS; where TMS is greater

than 0.1, and PS is considered discrete in uniform steps of 0.5 A. The listed relays' $CTRs$, P/B relay pairs, currents for close-in 3ϕ faults, and the other information regarding this test system are available in Appendix D.

5.2.5.1 Scenario 5.1: IDMT-Based DOCRs with BBO and BBO-LP

Table 5.17 gives the overall comparison between Seeker Algorithm (Amraee, 2012), BBO and BBO-LP. As seen from that table, BBO-LP consumes more CPU time as compared to Seeker algorithm for same population size and generation number. The solutions of the BBO and BBO-LP algorithms are listed in Table 5.18. Although Seeker Algorithm provides slightly better result with less CPU time, it has some small violations, as tabulated in Table 5.19 in bold font. Because of feasibility checker (*i.e., pre-processing unit in* Fig.(5.2)), both BBO and BBO-LP provide feasible solutions, as shown in Table 5.20. In addition, one of the main strength of the BBO-LP algorithm is its convergence to good solutions even with small population size and generations, which would consequently lead to lower CPU time. Table 5.21 shows the processing speeds and solutions quality of BBO-LP at different population size and generation number. Therefore, because BBO-LP can solve the coordination problem within a very short time, it proves again that it is very convenient and suitable for the applications of online optimal coordination when networks with dynamic configurations are considered, see Fig.(3.19).

Table (5.17)
Overall comparison with other paper result for the test case 5 - scenario 1

Comparison Criteria	**Seeker** (Amraee, 2012)	**BBO**	**BBO-LP**
Pop. #	**120**	**120**	**120**
Gen. #	**385**	20000	385
Best (s)	**12.227**	14.794	12.609
CPU (s)	**406.3**	46826.3	568.8

Table (5.18)
Comparison between BBO and BBO-LP for the test case 5 - scenario 1

Primary Relay	**Conventional BBO**			**Hybrid BBO-LP**		
	TMS	PS	T_{pr}	TMS	PS	T_{pr}
R_1	0.149243	1.0	0.324573	0.100000	1.5	0.250998
R_2	0.102030	1.5	0.273325	0.102094	1.0	0.234973
R_3	0.118406	2.5	0.352364	0.108246	2.0	0.292964
R_4	0.104188	2.0	0.322547	0.115778	1.0	0.270994
R_5	0.170393	1.5	0.442249	0.100000	2.5	0.323869
R_6	0.133252	2.0	0.379311	0.107186	2.0	0.305111
R_7	0.153023	1.5	0.396686	0.100000	2.5	0.323384
R_8	0.160373	1.0	0.366409	0.100000	2.0	0.300008
R_9	0.105410	2.5	0.315822	0.101814	2.0	0.277267
R_{10}	0.101423	2.5	0.317389	0.146962	1.0	0.321178
R_{11}	0.116382	1.5	0.319100	0.100000	1.5	0.274184
R_{12}	0.102454	2.0	0.323703	0.100000	2.0	0.315949
R_{13}	0.102792	2.5	0.328989	0.100000	2.0	0.289189
R_{14}	0.101182	2.5	0.340466	0.120160	1.0	0.276365
R_{15}	0.100333	2.0	0.300521	0.104278	1.0	0.237948
R_{16}	0.104579	2.0	0.321471	0.101271	1.5	0.274888
R_{17}	0.111252	2.0	0.308695	0.110877	1.5	0.274659
R_{18}	0.100794	2.0	0.266731	0.105902	1.0	0.219318
R_{19}	0.109154	2.0	0.295001	0.142097	1.0	0.299223
R_{20}	0.106228	2.0	0.292155	0.100000	2.0	0.275025
R_{21}	0.102890	1.5	0.244670	0.100000	1.5	0.237797
R_{22}	0.100278	2.0	0.273770	0.100000	2.0	0.273012
R_{23}	0.100230	2.0	0.294773	0.100000	1.5	0.260960
R_{24}	0.101722	2.0	0.308241	0.107007	1.5	0.286783
R_{25}	0.159572	1.5	0.428185	0.104058	2.0	0.315757
R_{26}	0.108404	2.0	0.328229	0.100000	2.5	0.336710
R_{27}	0.175603	1.0	0.423887	0.103385	2.0	0.333255
R_{28}	0.141842	2.0	0.412055	0.108399	2.5	0.348672
R_{29}	0.108142	2.0	0.287267	0.106225	1.5	0.253014
R_{30}	0.113317	2.0	0.324833	0.100000	2.0	0.286659
R_{31}	0.116985	2.0	0.322027	0.100000	2.0	0.275272
R_{32}	0.169867	1.0	0.405830	0.100000	2.0	0.318000
R_{33}	0.135047	2.5	0.454223	0.119882	2.0	0.362628
R_{34}	0.159496	1.5	0.408711	0.127606	2.0	0.367717
R_{35}	0.126460	2.0	0.399774	0.108329	2.0	0.342456
R_{36}	0.103421	2.5	0.336720	0.100000	2.0	0.293717
R_{37}	0.123310	2.5	0.400413	0.123340	2.0	0.361401
R_{38}	0.141307	2.0	0.445759	0.106228	2.5	0.374324
R_{39}	0.132553	2.5	0.461804	0.105350	2.5	0.367032
R_{40}	0.170986	1.5	0.453613	0.123078	2.0	0.368717
R_{41}	0.190757	1.5	0.463868	0.108261	2.5	0.323699
R_{42}	0.100999	2.5	0.328253	0.109272	1.5	0.284420
Best (s)	14.794412			**12.609497**		

Table (5.19)
Feasibility comparison of MINLP and Seeker algorithms (Amraee, 2012) for the test case 5 - scenario 1

P/B Pair		Violations		P/B Pair		Violations	
R_i	R_j	**MINLP**	**Seeker**	R_i	R_j	**MINLP**	**Seeker**
R_1	R_6	**-0.001**	**-0.002**	R_{20}	R_{30}	0.046	0.035
R_2	R_4	0.114	0.003	R_{21}	R_{17}	0.110	0.041
R_2	R_{16}	0.069	0.054	R_{21}	R_{19}	0.001	0.001
R_3	R_1	0.382	0.001	R_{21}	R_{30}	0.000	**-0.001**
R_3	R_{16}	0.001	0.003	R_{22}	R_{23}	0.141	0.069
R_4	R_7	0.049	0.008	R_{22}	R_{34}	0.063	0.001
R_4	R_{12}	0.051	0.023	R_{23}	R_{11}	0.002	0.044
R_4	R_{20}	0.142	0.052	R_{23}	R_{13}	0.059	0.032
R_5	R_2	0.143	0.000	R_{24}	R_{21}	0.075	0.070
R_6	R_8	**-0.001**	0.003	R_{24}	R_{34}	**-0.001**	**-0.001**
R_6	R_{10}	0.081	0.000	R_{25}	R_{15}	0.001	**-0.003**
R_7	R_5	**-0.001**	0.001	R_{25}	R_{18}	0.007	**-0.001**
R_7	R_{10}	0.091	**-0.002**	R_{26}	R_{28}	0.048	0.025
R_8	R_3	0.067	**-0.002**	R_{26}	R_{36}	0.112	0.056
R_8	R_{12}	0.070	0.005	R_{27}	R_{25}	0.002	0.002
R_8	R_{20}	0.161	0.034	R_{27}	R_{36}	0.002	0.021
R_9	R_5	0.043	0.021	R_{28}	R_{29}	0.065	**-0.001**
R_9	R_8	0.052	0.022	R_{28}	R_{32}	0.055	**-0.002**
R_{10}	R_{14}	0.072	**-0.001**	R_{29}	R_{17}	0.166	0.076
R_{11}	R_3	0.068	0.011	R_{29}	R_{19}	0.056	0.036
R_{11}	R_7	0.070	0.003	R_{29}	R_{22}	0.031	0.000
R_{11}	R_{20}	0.163	0.047	R_{30}	R_{27}	**-0.001**	0.000
R_{12}	R_{13}	**-0.001**	**-0.001**	R_{30}	R_{32}	0.062	0.046
R_{12}	R_{24}	0.153	0.004	R_{31}	R_{27}	0.048	0.014
R_{13}	R_9	0.002	0.000	R_{31}	R_{29}	0.121	0.062
R_{14}	R_{11}	0.076	0.034	R_{32}	R_{33}	0.080	0.016
R_{14}	R_{24}	0.286	0.027	R_{32}	R_{42}	0.116	0.047
R_{15}	R_1	0.337	0.050	R_{33}	R_{21}	**-0.002**	0.002
R_{15}	R_4	0.001	0.000	R_{33}	R_{23}	**-0.001**	**-0.001**
R_{16}	R_{18}	0.115	0.040	R_{34}	R_{31}	**-0.001**	0.028
R_{16}	R_{26}	0.057	0.006	R_{34}	R_{42}	0.000	**-0.001**
R_{17}	R_{15}	0.173	0.032	R_{35}	R_{25}	0.063	0.011
R_{17}	R_{26}	0.121	0.000	R_{35}	R_{28}	0.000	0.000
R_{18}	R_{19}	0.084	0.066	R_{36}	R_{38}	**-0.002**	**-0.001**
R_{18}	R_{22}	0.059	0.030	R_{37}	R_{35}	0.002	**-0.001**
R_{18}	R_{30}	0.083	0.064	R_{38}	R_{40}	0.001	**-0.001**
R_{19}	R_3	**-0.002**	0.009	R_{39}	R_{37}	**-0.001**	**-0.001**
R_{19}	R_7	**-0.001**	0.001	R_{40}	R_{41}	**-0.001**	0.000
R_{19}	R_{12}	0.001	0.017	R_{41}	R_{31}	0.036	0.058
R_{20}	R_{17}	0.156	0.078	R_{41}	R_{33}	0.002	**-0.001**
R_{20}	R_{22}	0.022	0.002	R_{42}	R_{39}	0.001	0.003

Table (5.20)
Feasibility comparison of BBO and BBO-LP for the test case 5 - scenario 1

P/B Pair		Violations		P/B Pair		Violations	
R_i	R_j	**BBO**	**BBO-LP**	R_i	R_j	**BBO**	**BBO-LP**
R_1	R_6	0.036	0.000	R_{20}	R_{30}	0.048	0.001
R_2	R_4	0.168	0.003	R_{21}	R_{17}	0.138	0.037
R_2	R_{16}	0.167	0.058	R_{21}	R_{19}	0.073	0.015
R_3	R_1	0.061	0.052	R_{21}	R_{30}	0.095	0.039
R_3	R_{16}	0.088	0.000	R_{22}	R_{23}	0.504	0.220
R_4	R_7	0.055	0.057	R_{22}	R_{34}	0.049	0.014
R_4	R_{12}	0.114	0.150	R_{23}	R_{11}	0.075	0.028
R_4	R_{20}	0.186	0.196	R_{23}	R_{13}	0.042	0.120
R_5	R_2	0.110	0.000	R_{24}	R_{21}	0.193	0.195
R_6	R_8	0.012	0.086	R_{24}	R_{34}	0.015	0.000
R_6	R_{10}	0.115	0.018	R_{25}	R_{15}	0.365	0.000
R_7	R_5	0.069	0.029	R_{25}	R_{18}	0.339	0.000
R_7	R_{10}	0.098	0.000	R_{26}	R_{28}	0.081	0.006
R_8	R_3	0.078	0.000	R_{26}	R_{36}	0.174	0.020
R_8	R_{12}	0.070	0.121	R_{27}	R_{25}	0.058	0.009
R_8	R_{20}	0.142	0.167	R_{27}	R_{36}	0.079	0.023
R_9	R_5	0.149	0.075	R_{28}	R_{29}	0.102	0.000
R_9	R_8	0.075	0.114	R_{28}	R_{32}	0.052	0.101
R_{10}	R_{14}	0.529	0.000	R_{29}	R_{17}	0.095	0.022
R_{11}	R_3	0.125	0.026	R_{29}	R_{19}	0.030	0.000
R_{11}	R_7	0.059	0.054	R_{29}	R_{22}	0.011	0.044
R_{11}	R_{20}	0.190	0.193	R_{30}	R_{27}	0.032	0.000
R_{12}	R_{13}	0.013	0.065	R_{30}	R_{32}	0.139	0.163
R_{12}	R_{24}	0.092	0.000	R_{31}	R_{27}	0.035	0.011
R_{13}	R_9	0.072	0.000	R_{31}	R_{29}	0.192	0.073
R_{14}	R_{11}	0.029	0.013	R_{32}	R_{33}	0.083	0.006
R_{14}	R_{24}	0.075	0.040	R_{32}	R_{42}	0.251	0.050
R_{15}	R_1	0.113	0.107	R_{33}	R_{21}	0.047	0.119
R_{15}	R_4	0.141	0.000	R_{33}	R_{23}	0.323	0.130
R_{16}	R_{18}	0.446	0.041	R_{34}	R_{31}	0.057	0.001
R_{16}	R_{26}	0.043	0.152	R_{34}	R_{42}	0.248	0.000
R_{17}	R_{15}	0.484	0.041	R_{35}	R_{25}	0.082	0.000
R_{17}	R_{26}	0.055	0.152	R_{35}	R_{28}	0.010	0.000
R_{18}	R_{19}	0.051	0.034	R_{36}	R_{38}	0.033	0.000
R_{18}	R_{22}	0.031	0.078	R_{37}	R_{35}	0.055	0.000
R_{18}	R_{30}	0.073	0.057	R_{38}	R_{40}	0.020	0.000
R_{19}	R_3	0.149	0.001	R_{39}	R_{37}	0.006	0.000
R_{19}	R_7	0.083	0.028	R_{40}	R_{41}	0.064	0.000
R_{19}	R_{12}	0.141	0.122	R_{41}	R_{31}	0.002	0.045
R_{20}	R_{17}	0.090	0.000	R_{41}	R_{33}	0.025	0.000
R_{20}	R_{22}	0.006	0.022	R_{42}	R_{39}	0.081	0.000

Table (5.21)
Performance quality of BBO-LP under different parameters for the test case 5 - scenario 1

Primary	**Hybrid BBO-LP***			**Hybrid BBO-LP****		
R_i	TMS	PS	T_{pr}	TMS	PS	T_{pr}
R_1	0.100000	2.0	0.281575	0.100000	1.5	0.250998
R_2	0.100000	2.5	0.336820	0.100000	2.0	0.302873
R_3	0.132299	1.5	0.320467	0.100444	2.5	0.298911
R_4	0.109801	1.5	0.299925	0.100000	2.0	0.309583
R_5	0.100000	2.5	0.323869	0.100000	2.5	0.323869
R_6	0.100000	2.5	0.314535	0.126393	1.5	0.320351
R_7	0.113573	2.0	0.331541	0.103608	2.5	0.335050
R_8	0.100000	2.0	0.300008	0.110025	1.5	0.292258
R_9	0.128850	1.5	0.313856	0.130297	1.5	0.317379
R_{10}	0.117383	1.5	0.296281	0.100000	2.0	0.283339
R_{11}	0.100000	2.0	0.310900	0.100000	2.5	0.346731
R_{12}	0.100000	2.0	0.315949	0.103517	1.5	0.287908
R_{13}	0.141531	1.0	0.314280	0.100000	2.5	0.320053
R_{14}	0.100000	1.5	0.267676	0.100000	1.5	0.267676
R_{15}	0.100000	1.5	0.265247	0.100000	1.5	0.265247
R_{16}	0.142399	1.0	0.331495	0.133394	1.0	0.310532
R_{17}	0.110877	1.5	0.274659	0.111745	1.5	0.276809
R_{18}	0.100000	2.0	0.264630	0.100000	1.5	0.237370
R_{19}	0.100000	2.5	0.297126	0.101022	2.0	0.273023
R_{20}	0.100000	2.0	0.275025	0.104967	1.5	0.257957
R_{21}	0.118340	1.0	0.245462	0.100000	2.0	0.265157
R_{22}	0.100000	2.5	0.300439	0.100000	2.0	0.273012
R_{23}	0.100000	2.0	0.294098	0.100000	1.5	0.260960
R_{24}	0.107007	1.5	0.286783	0.100000	2.0	0.303022
R_{25}	0.108530	2.0	0.329325	0.100000	2.5	0.337521
R_{26}	0.102128	2.0	0.309228	0.119708	1.5	0.320598
R_{27}	0.124039	1.5	0.351155	0.110609	2.0	0.356543
R_{28}	0.131639	2.0	0.382413	0.181502	1.0	0.404465
R_{29}	0.100000	2.0	0.265638	0.117027	1.5	0.278742
R_{30}	0.100000	2.0	0.286659	0.145399	1.0	0.320667
R_{31}	0.101572	2.0	0.279601	0.123689	1.5	0.304211
R_{32}	0.114180	1.5	0.319388	0.100000	2.0	0.318000
R_{33}	0.111333	2.5	0.374463	0.106873	2.5	0.359462
R_{34}	0.131186	2.0	0.378033	0.115277	2.5	0.367508
R_{35}	0.131445	1.5	0.365766	0.107798	2.0	0.340778
R_{36}	0.100000	2.5	0.325583	0.100056	2.0	0.293881
R_{37}	0.110448	2.5	0.358650	0.110448	2.5	0.358650
R_{38}	0.130382	2.0	0.411294	0.106263	2.5	0.374449
R_{39}	0.144550	1.5	0.397850	0.144550	1.5	0.397850
R_{40}	0.110975	2.5	0.369289	0.123105	2.0	0.368797
R_{41}	0.151239	1.5	0.367771	0.126932	2.0	0.345026
R_{42}	0.100000	2.0	0.293246	0.100000	2.0	0.293246
Best (s)	13.327999			13.120464		
CPU (s)	22.201471			49.180755		
Parameters	*10 pop. and 100 gen.			**20 pop. and 150 gen.		

5.2.5.2 Scenario 5.2: Variable TCCC-Based DOCRs with BBO

All the previous test cases are taken based on IDMT TCCC. With this assumption, Eq.(3.3) will have only two variables. Therefore, solving the 15-bus system of Fig.(5.9) requires dealing with a constrained problem of 84 variables. By considering different characteristic curves, the total variables of the 15-bus system problem will increase to 210 variables, because α, β and γ of Eq.(3.3) will change depending on the TCCC.

Actually, the solution quality and algorithm speed are two conflicting things that need to be compromised. When different TCCCs are available, the algorithm will have the ability to search for better solutions. On the other hand, this approach will increase the number of variables by 250%, and make the algorithm very slow. For example, the conventional BBO takes more than 50 minutes when the population size is 120 and number of generations is 385 for the case when different TCCCs are considered. The issue of algorithm processing speed can be solved by hybridizing the BBO with linear programming.

To match between North American standards (i.e., IEEE and CO-family[2]) and European standards (i.e., IEC, UK and AREVA) it is essen-

[2]Westinghouse electromechanical OCRs brand, and it is acquired by ABB since 1989. CO means Circuit Opening, and the OCRs listed under this series are: CO-2, CO-5, CO-6, CO-7, CO-8, CO-9 and CO-11. Based on (AREVA, 2011; Gers & Holmes, 2004), the TCCCs constants (α, β and γ) of just CO-2 and CO-8 OCRs are provided.

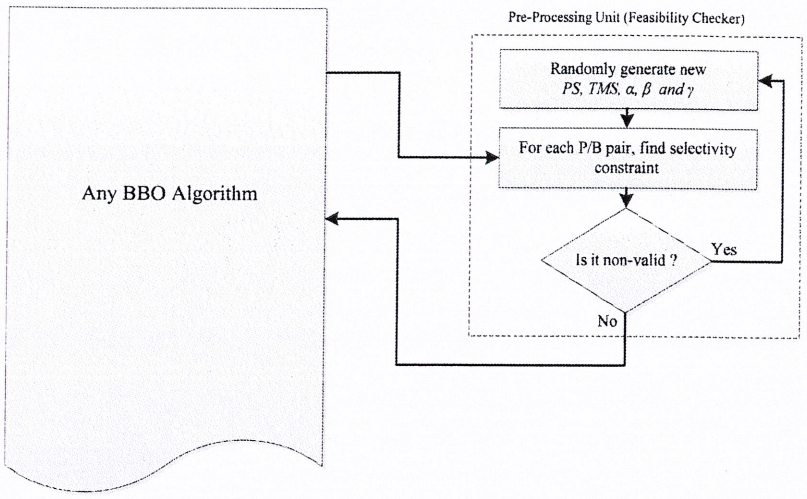


Figure (5.10)
Proposed modified flowchart of the BBO algorithm

tial to modify the original IEEE equation from Eq.(5.6) to be as Eq.(3.3) when TDS is divided by 7 (AREVA, 2011; Gers & Holmes, 2004).

$$T = \frac{TDS}{7} \times \left[\frac{\beta}{PSM^{\alpha} - 1} + \gamma \right] \tag{5.6}$$

In North American standards, TDS starts from 0.5 to 11; while European standards has TMS in a range between 0.1 and 1.1 (ALSTOM, 2002).

The pre-processing unit which will be used to detect and correct any violated selectivity constraint as covered in Sub-Section 5.1.2 has to be modified as in Fig.(5.10). Thus, the while loop will require less effort to detect new feasible set of $\{TMS, PS, \alpha, \beta, \gamma\}$.

The last important note is about how to incorporate and formulate the new modification within the BBO algorithm. This can be done by modifying Eq.(5.1), so the total number of n SIV will be divided into

five groups as:

- SIV_1 - SIV_{42}: TMS_1 - TMS_{42}
- SIV_{43} - SIV_{84}: PS_1 - PS_{42}
- SIV_{85} - SIV_{126}: α_1 - α_{42}
- SIV_{127} - SIV_{168}: β_1 - β_{42}
- SIV_{169} - SIV_{210}: γ_1 - γ_{42}

There are many types of TCCCs available today based on the standard values of α, β and γ. The listed popular standard TCCCs in Table 3.3 are used in this research work.

5.2.7.2.1 Performance Evaluation and Discussions

To evaluate the performance of this high dimensional case study, there are six possibilities that can be listed as follows:

- **Considering All TCCC Standards:**

  Table 5.22 shows the optimal coordination based on IEEE, IEC, AREVA, UK and CO-family standard values.
- **Considering Only North American TCCC Standards:**

  Table 5.23 shows the optimal coordination based on IEEE and CO-family standard values.
- **Considering Only European TCCC Standards:**

  Table 5.24 shows the optimal coordination based on IEC, AREVA and UK standard values.
- **Considering Only IEEE and IEC TCCC Standards:**

  Table 5.25 shows the optimal coordination based on both IEEE and IEC standard values.

- **Considering Only IEEE TCCC Standards:**

  Table 5.26 shows the optimal coordination based on IEEE standard values.

- **Considering Only IEC TCCC Standards:**

  Table 5.27 shows the optimal coordination based on IEC standard values.

As can be seen from Table 5.22 to Table 5.27, the obtained result with using only IEC TCCC values gives both the best solution and CPU time. All the results can be presented graphically as shown in loglog plot of Fig.(5.11).

5.2.7.2.2 Performance Comparison with IDMT-based DOCRs

To see the advantages of using DOCRs with different TCCCs, the next lines will give a brief comparison between the results that are obtained in the previous two tables (*i.e.,* Tables 5.22 *and* 5.23) and the results that are obtained in Table 5.17). It is obviously noticed from Table 5.28 that the solutions obtained by different TCCCs are better than that obtained by only IDMT TCCC. In contrast, the dimension of the coordination problem will increase by 250%, which consumes more CPU time than BBO-LP. However, it has been observed that the obtained solutions by using different TCCCs are better even with very small number of population size and generations, and this approach will need very small CPU time (less than 60 seconds). Moreover, the previous processing speed issue could be solved if the BBO algorithm is hybridized with LP stage and/or ISS model. This could be one of the future works.

Table (5.22)
Optimal coordination of the test case 5 with all standards - scenario 2

Primary Relay	DOCRs Settings		TCCCs Values			TCCCs Standard Names
	TMS	PS	α	β	γ	
R_1	0.182885	1.5	2	80	0	IEC Extremely Inverse
R_2	0.232583	2.5	2	5.95	0.18	CO8 Inverse
R_3	0.623956	2.5	2	28.2	0.1217	IEEE Extremely Inverse
R_4	0.147104	2.5	2	28.2	0.1217	IEEE Extremely Inverse
R_5	0.540057	2.5	0.04	0.05	0	AREVA Short-Time Inverse
R_6	0.553597	1.5	2	28.2	0.1217	IEEE Extremely Inverse
R_7	0.339854	1.5	2	80	0	IEC Extremely Inverse
R_8	0.748261	1.5	2	5.95	0.18	CO8 Inverse
R_9	0.503833	2.5	2	5.95	0.18	CO8 Inverse
R_{10}	0.319966	1.5	2	28.2	0.1217	IEEE Extremely Inverse
R_{11}	0.490257	2.0	2	5.95	0.18	CO8 Inverse
R_{12}	0.518064	2.5	2	5.95	0.18	CO8 Inverse
R_{13}	0.243115	1.5	2	80	0	IEC Extremely Inverse
R_{14}	0.845047	1.0	2	28.2	0.1217	IEEE Extremely Inverse
R_{15}	0.797605	2.5	2	28.2	0.1217	IEEE Extremely Inverse
R_{16}	0.231673	2.5	2	28.2	0.1217	IEEE Extremely Inverse
R_{17}	0.331655	2.5	2	28.2	0.1217	IEEE Extremely Inverse
R_{18}	0.488519	1.5	2	28.2	0.1217	IEEE Extremely Inverse
R_{19}	0.693861	2.0	0.04	0.05	0	AREVA Short-Time Inverse
R_{20}	0.219173	1.5	2	80	0	IEC Extremely Inverse
R_{21}	0.160622	2.0	2	28.2	0.1217	IEEE Extremely Inverse
R_{22}	0.906792	1.0	2	28.2	0.1217	IEEE Extremely Inverse
R_{23}	0.182915	1.5	1	13.5	0	IEC Very Inverse
R_{24}	0.601498	1.0	2	28.2	0.1217	IEEE Extremely Inverse
R_{25}	1.047118	0.5	0.02	0.02394	0.01694	CO2 Short-Time Inverse
R_{26}	0.803415	1.0	2	80	0	IEC Extremely Inverse
R_{27}	0.481713	2.0	2	28.2	0.1217	IEEE Extremely Inverse
R_{28}	1.008869	2.5	2	5.95	0.18	CO8 Inverse
R_{29}	0.426727	2.5	2	5.95	0.18	CO8 Inverse
R_{30}	0.479480	1.0	2	80	0	IEC Extremely Inverse
R_{31}	0.786416	1.5	2	80	0	IEC Extremely Inverse
R_{32}	0.806506	2.0	2	5.95	0.18	CO8 Inverse
R_{33}	0.421003	2.0	2	28.2	0.1217	IEEE Extremely Inverse
R_{34}	0.861915	1.5	2	80	0	IEC Extremely Inverse
R_{35}	0.465239	1.0	2	19.61	0.491	IEEE Very Inverse
R_{36}	0.254528	2.0	2	19.61	0.491	IEEE Very Inverse
R_{37}	0.265497	1.5	2	80	0	IEC Extremely Inverse
R_{38}	0.931085	2.5	2	5.95	0.18	CO8 Inverse
R_{39}	0.773385	0.5	1	13.5	0	IEC Very Inverse
R_{40}	0.550922	1.5	2	28.2	0.1217	IEEE Extremely Inverse
R_{41}	0.384480	1.5	2	80	0	IEC Extremely Inverse
R_{42}	0.445843	2.5	0.04	0.05	0	AREVA Short-Time Inverse
Best (s)				7.621750		

Table (5.23)
Optimal coordination of the test case 5 with North American standards - scenario 2

Primary Relay	DOCRs Settings		TCCCs Values			TCCCs Standard Names
	TMS	PS	α	β	γ	
R_1	0.823833	2.0	2	5.95	0.18	CO8 Inverse
R_2	0.700426	1.0	2	28.2	0.1217	IEEE Extremely Inverse
R_3	0.162518	2.5	2	28.2	0.1217	IEEE Extremely Inverse
R_4	0.590500	1.0	2	28.2	0.1217	IEEE Extremely Inverse
R_5	0.878431	1.0	2	28.2	0.1217	IEEE Extremely Inverse
R_6	0.390683	2.0	2	28.2	0.1217	IEEE Extremely Inverse
R_7	0.403984	2.0	2	28.2	0.1217	IEEE Extremely Inverse
R_8	0.399445	1.5	2	28.2	0.1217	IEEE Extremely Inverse
R_9	0.508553	2.5	2	5.95	0.18	CO8 Inverse
R_{10}	0.911574	1.5	2	5.95	0.18	CO8 Inverse
R_{11}	0.254173	2.0	2	28.2	0.1217	IEEE Extremely Inverse
R_{12}	0.260103	2.5	2	5.95	0.18	CO8 Inverse
R_{13}	0.170810	2.5	2	28.2	0.1217	IEEE Extremely Inverse
R_{14}	0.463905	1.0	2	28.2	0.1217	IEEE Extremely Inverse
R_{15}	0.190103	2.5	2	5.95	0.18	CO8 Inverse
R_{16}	0.617486	1.0	2	28.2	0.1217	IEEE Extremely Inverse
R_{17}	0.386412	1.5	2	28.2	0.1217	IEEE Extremely Inverse
R_{18}	0.337522	1.5	2	5.95	0.18	CO8 Inverse
R_{19}	0.460931	1.5	2	28.2	0.1217	IEEE Extremely Inverse
R_{20}	0.713340	2.0	2	5.95	0.18	CO8 Inverse
R_{21}	0.656518	1.0	2	28.2	0.1217	IEEE Extremely Inverse
R_{22}	0.299992	1.5	2	19.61	0.491	IEEE Very Inverse
R_{23}	0.755177	1.5	2	5.95	0.18	CO8 Inverse
R_{24}	0.555525	1.5	2	28.2	0.1217	IEEE Extremely Inverse
R_{25}	0.158384	2.5	2	19.61	0.491	IEEE Very Inverse
R_{26}	0.712782	1.0	2	28.2	0.1217	IEEE Extremely Inverse
R_{27}	0.209772	2.5	2	28.2	0.1217	IEEE Extremely Inverse
R_{28}	0.645718	1.5	2	28.2	0.1217	IEEE Extremely Inverse
R_{29}	0.434565	1.5	2	28.2	0.1217	IEEE Extremely Inverse
R_{30}	0.692119	1.5	2	28.2	0.1217	IEEE Extremely Inverse
R_{31}	0.497312	2.0	2	28.2	0.1217	IEEE Extremely Inverse
R_{32}	0.155486	2.0	2	19.61	0.491	IEEE Very Inverse
R_{33}	0.450443	2.5	2	28.2	0.1217	IEEE Extremely Inverse
R_{34}	0.343872	2.5	2	28.2	0.1217	IEEE Extremely Inverse
R_{35}	0.954882	1.5	2	5.95	0.18	CO8 Inverse
R_{36}	0.384600	1.5	2	28.2	0.1217	IEEE Extremely Inverse
R_{37}	0.505554	1.5	2	28.2	0.1217	IEEE Extremely Inverse
R_{38}	0.286288	2.0	2	28.2	0.1217	IEEE Extremely Inverse
R_{39}	1.065169	1.0	2	28.2	0.1217	IEEE Extremely Inverse
R_{40}	0.617989	1.5	2	28.2	0.1217	IEEE Extremely Inverse
R_{41}	1.076917	1.5	2	28.2	0.1217	IEEE Extremely Inverse
R_{42}	0.419170	1.5	2	28.2	0.1217	IEEE Extremely Inverse
Best (s)				5.940783		

Table (5.24)
Optimal coordination of the test case 5 with European standards - scenario 2

Primary Relay	**DOCRs Settings**		**TCCCs Values**			**TCCCs Standard Names**
	TMS	PS	α	β	γ	
R_1	1.088388	0.5	2	80	0	IEC Extremely Inverse
R_2	0.712755	0.5	2	80	0	IEC Extremely Inverse
R_3	0.461135	1.5	2	80	0	IEC Extremely Inverse
R_4	0.750810	0.5	2	80	0	IEC Extremely Inverse
R_5	0.220041	2.0	1	13.5	0	IEC Very Inverse
R_6	0.631214	1.0	2	80	0	IEC Extremely Inverse
R_7	0.717340	0.5	1	13.5	0	IEC Very Inverse
R_8	0.997140	0.5	2	80	0	IEC Extremely Inverse
R_9	0.198899	1.5	2	80	0	IEC Extremely Inverse
R_{10}	0.504602	0.5	0.02	0.14	0	IEC Standard Inverse (IDMT)
R_{11}	0.272637	1.5	2	80	0	IEC Extremely Inverse
R_{12}	0.388771	1.0	2	80	0	IEC Extremely Inverse
R_{13}	0.832765	1.0	2	80	0	IEC Extremely Inverse
R_{14}	0.346046	2.0	2	80	0	IEC Extremely Inverse
R_{15}	0.125127	1.5	2	80	0	IEC Extremely Inverse
R_{16}	0.545718	0.5	1	13.5	0	IEC Very Inverse
R_{17}	0.189873	1.5	2	80	0	IEC Extremely Inverse
R_{18}	0.146818	2.5	2	80	0	IEC Extremely Inverse
R_{19}	0.210295	2.0	2	80	0	IEC Extremely Inverse
R_{20}	0.199592	2.0	2	80	0	IEC Extremely Inverse
R_{21}	0.231435	2.0	2	80	0	IEC Extremely Inverse
R_{22}	0.358182	1.0	2	80	0	IEC Extremely Inverse
R_{23}	0.636686	1.0	2	80	0	IEC Extremely Inverse
R_{24}	1.071958	0.5	2	80	0	IEC Extremely Inverse
R_{25}	0.374754	1.5	2	80	0	IEC Extremely Inverse
R_{26}	0.107117	2.5	0.02	0.14	0	IEC Standard Inverse (IDMT)
R_{27}	0.173467	2.5	1	13.5	0	IEC Very Inverse
R_{28}	0.386250	1.5	2	80	0	IEC Extremely Inverse
R_{29}	0.435328	1.0	2	80	0	IEC Extremely Inverse
R_{30}	0.629177	0.5	1	13.5	0	IEC Very Inverse
R_{31}	0.424960	2.5	0.04	0.05	0	AREVA Short-Time Inverse
R_{32}	0.732994	1.0	2	80	0	IEC Extremely Inverse
R_{33}	0.266705	1.5	2	80	0	IEC Extremely Inverse
R_{34}	0.312851	2.0	2	80	0	IEC Extremely Inverse
R_{35}	0.972214	0.5	2	80	0	IEC Extremely Inverse
R_{36}	0.856748	1.0	2	80	0	IEC Extremely Inverse
R_{37}	0.723006	1.0	2	80	0	IEC Extremely Inverse
R_{38}	0.203024	2.5	2	80	0	IEC Extremely Inverse
R_{39}	0.663764	1.5	2	80	0	IEC Extremely Inverse
R_{40}	0.175275	2.5	2	80	0	IEC Extremely Inverse
R_{41}	0.680511	0.5	1	13.5	0	IEC Very Inverse
R_{42}	0.271410	1.0	2	80	0	IEC Extremely Inverse
Best (s)	7.310656					

Table (5.25)
Optimal coordination of the test case 5 with IEEE and IEC standards - scenario 2

Primary Relay	DOCRs Settings		TCCCs Values			TCCCs Standard Names
	TMS	PS	α	β	γ	
R_1	0.276527	1.0	2	19.61	0.491	IEEE Very Inverse
R_2	0.449561	1.0	2	28.2	0.1217	IEEE Extremely Inverse
R_3	0.218243	2.0	2	80	0	IEC Extremely Inverse
R_4	0.647605	0.5	2	80	0	IEC Extremely Inverse
R_5	0.490741	2.0	2	28.2	0.1217	IEEE Extremely Inverse
R_6	0.543259	0.5	1	13.5	0	IEC Very Inverse
R_7	0.354686	2.0	2	80	0	IEC Extremely Inverse
R_8	0.327172	2.5	2	28.2	0.1217	IEEE Extremely Inverse
R_9	0.309761	2.5	2	28.2	0.1217	IEEE Extremely Inverse
R_{10}	0.459766	1.0	2	80	0	IEC Extremely Inverse
R_{11}	0.212769	1.5	2	28.2	0.1217	IEEE Extremely Inverse
R_{12}	0.410155	0.5	1	13.5	0	IEC Very Inverse
R_{13}	0.175136	1.5	2	80	0	IEC Extremely Inverse
R_{14}	0.208983	2.0	2	28.2	0.1217	IEEE Extremely Inverse
R_{15}	0.241927	2.0	2	28.2	0.1217	IEEE Extremely Inverse
R_{16}	0.252681	1.0	2	80	0	IEC Extremely Inverse
R_{17}	0.337084	2.5	2	19.61	0.491	IEEE Very Inverse
R_{18}	0.679174	1.0	2	28.2	0.1217	IEEE Extremely Inverse
R_{19}	0.782462	1.5	2	80	0	IEC Extremely Inverse
R_{20}	0.156913	2.0	2	28.2	0.1217	IEEE Extremely Inverse
R_{21}	0.361397	1.0	2	80	0	IEC Extremely Inverse
R_{22}	0.556410	1.0	2	80	0	IEC Extremely Inverse
R_{23}	0.369936	1.0	2	80	0	IEC Extremely Inverse
R_{24}	0.326438	0.5	1	13.5	0	IEC Very Inverse
R_{25}	0.488303	1.5	2	28.2	0.1217	IEEE Extremely Inverse
R_{26}	0.578651	1.0	2	80	0	IEC Extremely Inverse
R_{27}	0.320892	1.5	2	80	0	IEC Extremely Inverse
R_{28}	1.029839	1.0	2	80	0	IEC Extremely Inverse
R_{29}	0.732214	0.5	2	80	0	IEC Extremely Inverse
R_{30}	0.345105	1.5	2	80	0	IEC Extremely Inverse
R_{31}	0.968844	1.0	2	80	0	IEC Extremely Inverse
R_{32}	0.603803	0.5	1	13.5	0	IEC Very Inverse
R_{33}	0.807071	1.0	2	80	0	IEC Extremely Inverse
R_{34}	0.112575	1.5	0.02	0.14	0	IEC Standard Inverse (IDMT)
R_{35}	0.242782	1.5	2	80	0	IEC Extremely Inverse
R_{36}	0.422003	1.5	2	28.2	0.1217	IEEE Extremely Inverse
R_{37}	0.489966	2.0	2	28.2	0.1217	IEEE Extremely Inverse
R_{38}	0.301230	2.0	2	28.2	0.1217	IEEE Extremely Inverse
R_{39}	0.318956	1.5	2	80	0	IEC Extremely Inverse
R_{40}	0.225685	2.0	2	28.2	0.1217	IEEE Extremely Inverse
R_{41}	0.988232	1.0	2	80	0	IEC Extremely Inverse
R_{42}	0.224627	2.5	2	28.2	0.1217	IEEE Extremely Inverse
Best (s)				5.682841		

Table (5.26)
Optimal coordination of the test case 5 with IEEE standards - scenario 2

Primary Relay	DOCRs Settings		TCCCs Values			TCCCs Standard Names
	TMS	PS	α	β	γ	
R_1	0.379171	1.5	2	28.2	0.1217	IEEE Extremely Inverse
R_2	0.179213	1.5	2	28.2	0.1217	IEEE Extremely Inverse
R_3	0.224438	2.5	2	28.2	0.1217	IEEE Extremely Inverse
R_4	0.256803	1.5	2	28.2	0.1217	IEEE Extremely Inverse
R_5	0.261104	2.5	2	28.2	0.1217	IEEE Extremely Inverse
R_6	0.572082	1.5	2	28.2	0.1217	IEEE Extremely Inverse
R_7	0.574391	1.5	2	28.2	0.1217	IEEE Extremely Inverse
R_8	0.379976	1.5	2	28.2	0.1217	IEEE Extremely Inverse
R_9	0.263731	2.5	2	28.2	0.1217	IEEE Extremely Inverse
R_{10}	0.144754	2.0	2	28.2	0.1217	IEEE Extremely Inverse
R_{11}	0.198166	2.5	2	28.2	0.1217	IEEE Extremely Inverse
R_{12}	0.170450	2.0	2	28.2	0.1217	IEEE Extremely Inverse
R_{13}	0.272516	2.0	2	28.2	0.1217	IEEE Extremely Inverse
R_{14}	0.246320	1.5	2	28.2	0.1217	IEEE Extremely Inverse
R_{15}	0.187613	1.5	2	28.2	0.1217	IEEE Extremely Inverse
R_{16}	0.403661	1.5	2	28.2	0.1217	IEEE Extremely Inverse
R_{17}	0.519361	2.0	2	28.2	0.1217	IEEE Extremely Inverse
R_{18}	0.219411	2.0	2	28.2	0.1217	IEEE Extremely Inverse
R_{19}	0.479610	2.5	2	28.2	0.1217	IEEE Extremely Inverse
R_{20}	0.266397	2.5	2	28.2	0.1217	IEEE Extremely Inverse
R_{21}	0.338299	1.0	2	28.2	0.1217	IEEE Extremely Inverse
R_{22}	0.265689	2.0	2	28.2	0.1217	IEEE Extremely Inverse
R_{23}	0.195880	1.5	2	28.2	0.1217	IEEE Extremely Inverse
R_{24}	0.319259	2.5	2	19.61	0.491	IEEE Very Inverse
R_{25}	0.759574	1.5	2	28.2	0.1217	IEEE Extremely Inverse
R_{26}	0.789357	1.5	2	28.2	0.1217	IEEE Extremely Inverse
R_{27}	0.962536	1.5	2	28.2	0.1217	IEEE Extremely Inverse
R_{28}	0.504375	2.0	2	28.2	0.1217	IEEE Extremely Inverse
R_{29}	0.582609	1.5	2	28.2	0.1217	IEEE Extremely Inverse
R_{30}	0.314517	2.0	2	28.2	0.1217	IEEE Extremely Inverse
R_{31}	0.279691	2.0	2	28.2	0.1217	IEEE Extremely Inverse
R_{32}	0.243595	2.0	2	28.2	0.1217	IEEE Extremely Inverse
R_{33}	0.287231	2.0	2	28.2	0.1217	IEEE Extremely Inverse
R_{34}	0.424491	2.5	2	28.2	0.1217	IEEE Extremely Inverse
R_{35}	0.276828	2.0	2	28.2	0.1217	IEEE Extremely Inverse
R_{36}	0.163917	2.5	2	28.2	0.1217	IEEE Extremely Inverse
R_{37}	0.620242	1.5	2	28.2	0.1217	IEEE Extremely Inverse
R_{38}	0.398090	2.0	2	28.2	0.1217	IEEE Extremely Inverse
R_{39}	0.602126	0.5	2	19.61	0.491	IEEE Very Inverse
R_{40}	0.975034	1.5	2	28.2	0.1217	IEEE Extremely Inverse
R_{41}	0.469359	2.5	2	28.2	0.1217	IEEE Extremely Inverse
R_{42}	0.301767	1.5	2	28.2	0.1217	IEEE Extremely Inverse
Best (s)	5.832762					

Table (5.27)
Optimal coordination of the test case 5 with IEC standards - scenario 2

Primary	**DOCRs Settings**		**TCCCs Values**			**TCCCs Standard**
R_i	TMS	PS	α	β	γ	**Names**
R_1	0.561482	1.0	2	80	0	IEC Extremely Inverse
R_2	0.136387	1.0	2	80	0	IEC Extremely Inverse
R_3	0.544697	1.0	1	13.5	0	IEC Very Inverse
R_4	0.202612	1.5	2	80	0	IEC Extremely Inverse
R_5	0.535077	1.0	2	80	0	IEC Extremely Inverse
R_6	0.358583	1.5	2	80	0	IEC Extremely Inverse
R_7	0.634255	1.0	2	80	0	IEC Extremely Inverse
R_8	0.421144	1.0	2	80	0	IEC Extremely Inverse
R_9	0.365516	1.0	2	80	0	IEC Extremely Inverse
R_{10}	0.650417	1.0	2	80	0	IEC Extremely Inverse
R_{11}	0.169308	1.5	2	80	0	IEC Extremely Inverse
R_{12}	1.051114	0.5	2	80	0	IEC Extremely Inverse
R_{13}	0.387035	1.0	2	80	0	IEC Extremely Inverse
R_{14}	0.750933	0.5	2	80	0	IEC Extremely Inverse
R_{15}	0.337634	1.0	1	13.5	0	IEC Very Inverse
R_{16}	0.427700	1.0	2	80	0	IEC Extremely Inverse
R_{17}	0.233013	2.0	1	13.5	0	IEC Very Inverse
R_{18}	0.200405	1.0	2	80	0	IEC Extremely Inverse
R_{19}	0.478526	1.5	2	80	0	IEC Extremely Inverse
R_{20}	0.550367	0.5	2	80	0	IEC Extremely Inverse
R_{21}	0.115349	1.5	2	80	0	IEC Extremely Inverse
R_{22}	0.150889	1.5	2	80	0	IEC Extremely Inverse
R_{23}	0.135673	1.0	2	80	0	IEC Extremely Inverse
R_{24}	0.462898	0.5	1	13.5	0	IEC Very Inverse
R_{25}	0.615369	1.0	2	80	0	IEC Extremely Inverse
R_{26}	0.158366	2.0	2	80	0	IEC Extremely Inverse
R_{27}	0.388314	1.0	2	80	0	IEC Extremely Inverse
R_{28}	0.248596	1.5	2	80	0	IEC Extremely Inverse
R_{29}	0.720679	1.0	2	80	0	IEC Extremely Inverse
R_{30}	0.247832	2.0	2	80	0	IEC Extremely Inverse
R_{31}	0.633638	1.0	2	80	0	IEC Extremely Inverse
R_{32}	0.732885	0.5	2	80	0	IEC Extremely Inverse
R_{33}	0.569733	1.5	2	80	0	IEC Extremely Inverse
R_{34}	0.467033	1.5	2	80	0	IEC Extremely Inverse
R_{35}	0.258365	1.5	2	80	0	IEC Extremely Inverse
R_{36}	0.116042	1.5	2	80	0	IEC Extremely Inverse
R_{37}	0.186179	2.5	2	80	0	IEC Extremely Inverse
R_{38}	0.487529	1.0	2	80	0	IEC Extremely Inverse
R_{39}	0.334533	1.0	1	13.5	0	IEC Very Inverse
R_{40}	0.584387	0.5	1	13.5	0	IEC Very Inverse
R_{41}	0.118966	1.5	0.02	0.14	0	IEC Standard Inverse (IDMT)
R_{42}	0.591539	1.0	2	80	0	IEC Extremely Inverse
Best (s)				**5.259732**		

Table (5.28)
Performance comparison between DOCRs with different TCCCs and DOCRs with only IDMT model - test case 5

Comparison Criteria	**Only IDMT**			**Variable TCCC**					
	Seeker (Amraee, 2012)	**BBO**	**BBO-LP**	**BBO**[1]	**BBO**[2]	**BBO**[3]	**BBO**[4]	**BBO**[5]	**BBO**[6]
Population Size	120	120	120	120	120	120	120	120	120
TCCC Standards	**Only IEC IDMT**			[1]**All**	[2]**N. American**	[3]**European**	[4]**IEEE/IEC**	[5]**IEEE**	[6]**IEC**
Generation No.	385	20000	385	385	385	385	385	385	385
Variables No.	84	84	84	210	210	210	210	210	210
Min. OBJ (s)	12.227	14.794	12.609	7.622	5.941	7.311	5.683	5.833	**5.260**
CPU Time (s)	**406.3**	46826.3	568.8	3498.5	3546.1	3422.1	3265.3	3315.0	3177.3

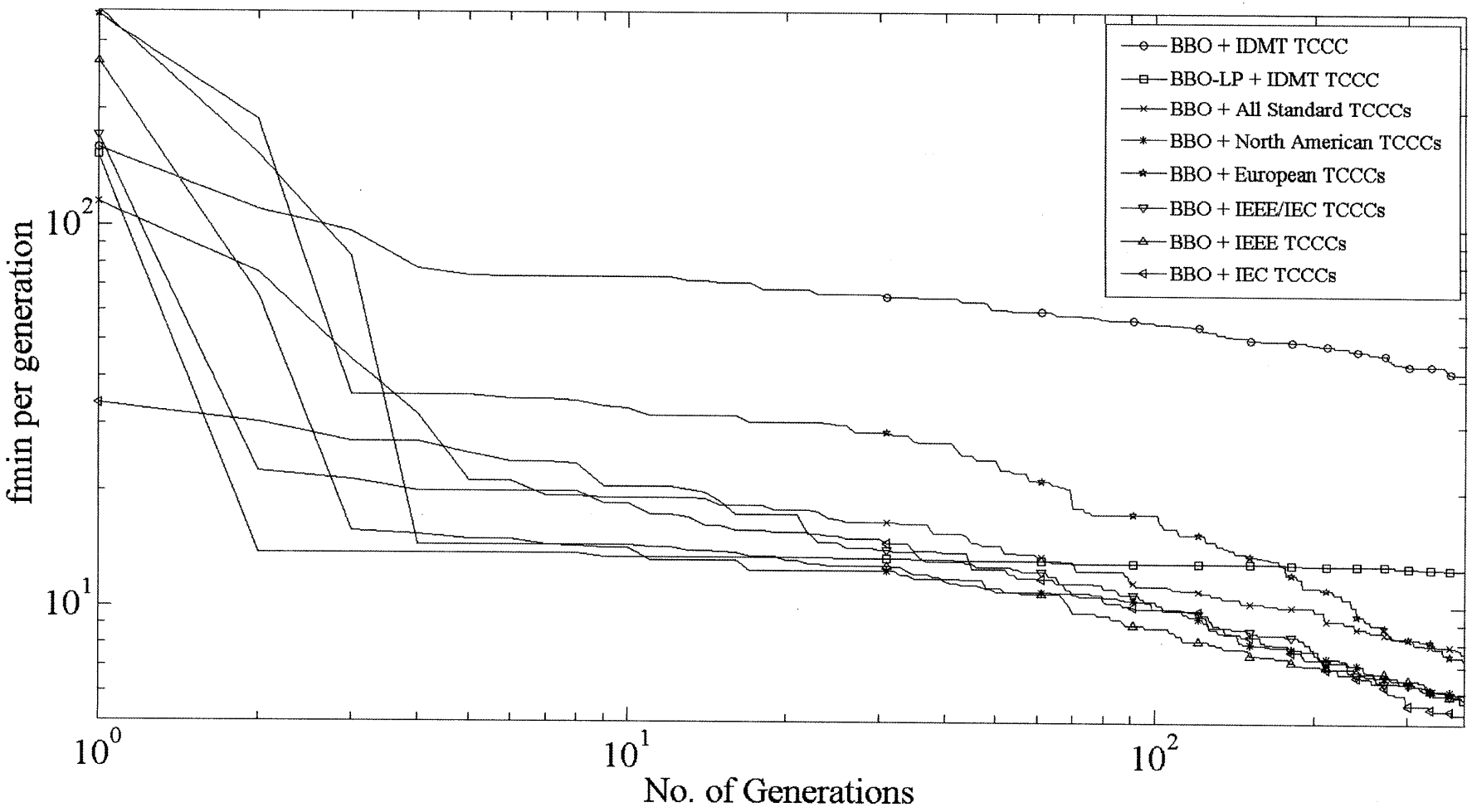


Figure (5.11)
Log-Log curves of fitness function with different TCCCs - test case 5

Chapter 6

Conclusion and Future Works

6.1 Suggestions

6.2 Scope of Future Works

6.3 Applications and Soft Tools

Chapter 6
Conclusion and Future Works

This thesis covers two main parts: the performance of different BBO algorithms and the optimal coordination of DOCRs using some selected modified/hybridized BBO algorithms.

By using large number of test functions, this thesis has found that the first presented BBO algorithm, which is called PMB-BBO, is the slowest algorithm; but it gives the best performance as the complexity, side constraints and/or dimensions of a given problem increases and as the number of islands or population size decreases.

The slow processing speed of PMB-BBO as well as its performance can be enhanced by applying some essential modifications that solve the associated weakness of the orignal BBO forms without using any complicated migration rates. It shows that the simplified linear model still can give good results. The new modified PMB-BBO is called BBO-EM, and it is used as a basis for all the studies of this thesis.

One of the most superior BBO models is presented in this study based on BBO-EM. This algorithm is called ISS-BBO, which utilizes the collected data from unerased islands to iterate the search space in decaying intervals. Based on the experimental results, ISS-BBO outperforms the original PMB-BBO and the other two proposed models (BSU and BMU). Moreover, ISS-BBO can perform better than the most advanced

BBO algorithms available in the literature that are equipped with some of the most complicated migration rates (Sin-BBO and GenSin-BBO). Furthermore, ISS-BBO is faster than all the BBOs available in the literature, that are based on the original BBO code by 31.6% of CPU time.

The iterative search space process can be replaced with an adaptive search space process (AVB-BBO) by just changing the conditions of the "if-statements" in the ISS-BBO algorithm, where the values of the new lower and upper bounds can be adaptively increased or decreased (zooming-in or zooming-out). Although, this approach can raise the level of algorithm's immunity against the local minimum traps, it shows less performance than ISS-BBO.

BBO is successfully applied to solve the optimal coordination problem of DOCRs in different case studies. The proposed further modifications on the BBO algorithm have enhanced its performance and satisfied all the design criteria (feasibility, optimality, simplicity and processing speed). The proposed hybrid BBO-LP achieved better performance and convergence speed as compared with other proposed algorithms in the literature. Based on the obtained results, BBO-LP can be effectively used to solve the adaptive optimal coordination for dynamic network topologies, where both the optimality and the realtime processing speed are very important factors in the field of power protection.

Also, for some advanced case studies, ISS-BBO has proved that the quality and speed performance of the BBO algorithm can be enhanced markedly.

Furthermore, if the coordination problem is considered with TCCCs instead of using only IDMT TCCC. The solution quality can be improved more, but the processing speed of the algorithm will decrease steeply.

6.1 Suggestions

The main objective of this thesis is to construct a strong foundation on each one of the above main two parts:

In the first part, it shows that the performance of the BBO algorithm can be enhanced even with the most simple migration model. This study uses only the basic BBO form, where λ_s and μ_s are linear and $I = E$. It is highly recommended to test the existing BBO models available in the literature by considering the code of BBO-EM. Moreover, ISS model is one of the most superior BBO algorithms, but its performance could be improved more if different migration and mutation rates are used. In addition, the strengths of Oppositional and Blended models (OBBO and BBBO) can be merged with the ISS model.

In the second part, the hybridization between BBO algorithm and LP algorithm can satisfies the optimality within very few processing time. The accuracy and precision can be improved if the complicated BBO models are used. Also, for case studies where the PS vector is available in a continuous form, a second stage of linear-programming (LP) can be used in the overall algorithm to fine tune PS, or even non-linear programming (NLP) can be used in the position of both LP sub-algorithms;

but, these add-ons may affect the simplicity and processing time.

6.2 Scope of Future Works

This thesis creates the main lines for the future works. Actually, it opens many of opportunities that have to be covered in the future, such as:

- Integrating different BBO algorithms into one superior BBO algorithm, like integrating the models of BBBO, OƎBO, GenSin-BBO and ISS-BBO into one final algorithm.
- Hybridizing BBO with other traditional and non-traditional optimization algorithms to increase the total advantages and minimize the total disadvantages of the proposed new model.
- Since BBO-LP has achieved different performance criteria within very short processing time, this algorithm will be used to solve the adaptive optimal coordination for dynamic network topologies.
- This study considers only DOCRs, while the practical transmission, sub-transmission and distribution systems could have a mixture of fuses, distance relays, directional and non directional overcurrent relays. Thus, one of the future missions is to focus on these complicated scenarios.
- Considering other design constraints, like CT saturation, cables thermal limits, etc.
- Because ISS-BBO performs very well with the 3-bus system, this algorithm will be further evaluated with other large systems.

- When different TCCCs are considered, better solutions can be extracted. But, the main faced problem is its slow processing time. Because the modern electrical networks will depend mainly on the numerical relays, so it is essential to solve this speed issue either by hybridizing the BBO algorithm with LP/NLP or by replacing it with other advanced BBO algorithms that need less population size and generation number. Also, because the numerical relays provide almost continuous tap settings PS, it is interesting to solve the problem based on continuous vectors of TMS and PS.
- The design criteria of any protection system has many conflicting criteria, as covered in Section 3.5. This motivated us to solve the optimal coordination problem based on multi-objective techniques. This part of the future works is almost completed and shows some encouraging results.

6.3 Applications and Soft Tools

The contents of this thesis are created by using different softwares as follows:

- MathWorks MATLAB 2011a
- LaTeX: MikTex 2.9, WinEdt 8, JabRef 2.9.2 and LaTable 0.7.2
- Microsoft Office 2010: Word, Excel and Visio
- Adobe Photoshop CS5 Extended (version 12.0)
- Autodesk 3ds Max 2014 (version 16.0)
- ETAP 7

References

Adorio, E. P. (2005, Jan.). *MVF - Multivariate Test Functions Library in C for Unconstrained Global Optimization.* Quezon City, Metro Manila, Philippines. Retrieved from `http://geocities.ws/eadorio/mvf.pdf`

Albrecht, R. E., Nisja, M. J., Feero, W. E., Rockefeller, G. D., & Wagner, C. L. (1964, April). Digital Computer Protective Device Coordination Program, Part I: General Program Description. *Power Apparatus and Systems, IEEE Transactions on, PAS-83*, 402-410.

Ali, M. M., Khompatraporn, C., & Zabinsky, Z. B. (2005, Apr.). A Numerical Evaluation of Several Stochastic Algorithms on Selected Continuous Global Optimization Test Problems. *Journal of Global Optimization, 31*(4), 635-672.

ALSTOM. (2002). *Network protection & automation guide.* Levallois-Perret, France: Author.

Amraee, T. (2012, July). Coordination of Directional Overcurrent Relays Using Seeker Algorithm. *Power Delivery, IEEE Transactions on, 27*(3), 1415-1422. doi: 10.1109/TPWRD.2012.2190107

Anthony, M. A. (1995). *Electric power system protection and coordination: A design handbook for overcurrent protection.* New York: McGraw-Hill Inc.

AREVA. (2011). *Directional / non Directional Overcurrent Protection.* Schneider Electric. Retrieved from `http://mt.schneider-electric.be/OP_MAIN/Micom/P12xy_TDS_EN_H76.pdf` ([Accessed March. 2, 2014])

Arfken, G. B., Weber, H. J., & Harris, F. E. (2012). *Mathematical methods for physicists: A comprehensive guide* (7th, Ed.). Waltham, Massachusetts: Academic Press.

Aria, M. (2013, Jan.). Educational Simulator for Teaching of Particle Swarm Optimization in LabView. *TELEKONTRAN, 1*(1), 1-15.

Bandyopadhyay, M. (2006). *Electrical power systems: Theory and practice.* New Delhi, India: Prentice-Hall of India Pvt. Ltd.

Bartz-Beielstein, T. (2006). *Experimental research in evolutionary computation: The new experimentalism.* Berlin, Germany: Springer.

Bedekar, P., & Bhide, S. (2011, Jan.). Optimum Coordination of Directional Overcurrent Relays Using the Hybrid GA-NLP Approach. *Power Delivery, IEEE Transactions on, 26*(1), 109-119. doi: 10.1109/TPWRD.2010.2080289

Belegundu, A. D., & Chandrupatla, T. R. (2011). *Optimization concepts and applications in engineering* (2nd. ed.). New York: Cambridge University Press.

Bhattacharya, A., & Chattopadhyay, P. (2009, Nov.). Biogeography-Based Optimization and its Application to Nonconvex Economic Emission Load Dispatch Problems. In *Advances in power system control, operation and management (apscom 2009), 8th international conference on* (p. 1-6). doi: 10.1049/cp.2009.1826

Birla, D., Maheshwari, R., & Gupta, H. (2006, July). A New Nonlinear Directional Overcurrent Relay Coordination Technique, and Banes and Boons of Near-End Faults Based Approach. *Power Delivery, IEEE Transactions on, 21*(3), 1176-1182. doi: 10.1109/TPWRD.2005.861325

Blackburn, J. L., & Domin, T. J. (2006). *Protective relaying: Principles and applications* (3rd ed.). Boca Raton, FL: CRC Press.

Breiman, L., & Cutler, A. (1993, Jan.). A Deterministic Algorithm for Global Optimization. *Mathematical Programming, 58*(1-3), 179-199.

Chao, C.-W., Fand, S.-C., & Liao, C.-J. (2012, Feb.). A Tropical Cyclone-Based Method for Global Optimization. *Journal of Industrial and Management Optimization, 8*(1), 103-115.

Chelouah, R., & Siarry, P. (2000, Jun.). Tabu Search Applied to Global Optimization. *European Journal of Operational Research, 123*(2), 256-270.

Chen, R.-B., Wang, W., & Tsai, F. (2006). A Basis-Based Response Surface Method for Computer Experiment Optimization. *Techincal report, Institute of Statistics and Department of Applied Math., National University of Kaohsiung.*

Chong, E. K. P., & Żak, S. H. (2001). *An introduction to optimization* (2nd ed.). New York: Wiley-Interscience.

Chou, W.-C. (2013, Mar.). *Multiple Optima Problems in Higher Dimensions.* Taipei,

Taiwan. Retrieved from http://www.math.ntu.edu.tw/~wwang/cola_lab/test_problems/multiple_opt/multiopt_prob/ ([Accessed April 25, 2013])

Cody, M. L. (2006). *Plants on islands: Diversity and dynamics on a continental archipelago*. Berkeley, Calif: University of California Press.

Costa, M. F. P., Fernandes, F. P., & da G. P. Fernandes, E. M. (2010, Sept.). A Deterministic-Stochastic Method for Nonconvex MINLP Problems. In *2nd international conference on engineering optimization* (p. 1-10).

Cox, C. B., & Moore, P. D. (1993). *Biogeography: An ecological and evolutionary approach*. Oxford England Cambridge, Mass., USA: Blackwell Scientific Publications.

Damborg, M., Ramaswami, R., Venkata, S., & Postforoosh, J. M. (1984, Jan.). Computer Aided Transmission Protection System Design Part I: Alcorithms. *Power Apparatus and Systems, IEEE Transactions on, PAS-103*(1), 51-59. doi: 10.1109/TPAS.1984.318576

Deb, K. (2010). *Optimization for engineering design: Algorithms and examples* (11th ed.). Prentice-Hall of India.

Deep, K., & Bansal, J. (2009, Dec). Optimization of Directional Overcurrent Relay Times Using Laplace Crossover Particle Swarm Optimization (LXPSO). In *Nature biologically inspired computing, 2009. nabic 2009. world congress on* (p. 288-293). doi: 10.1109/NABIC.2009.5393722

Deep, K., Birla, D., Maheshwari, R. P., Gupta, H. O., & Thakur, M. (2006, March). A Population Based Heuristic Algorithm for Optimal Relay Operating Times. *World Journal of Modeling and Simulation*, *2*(3), 167–176.

Desmond, R. (2007). *Sir joseph dalton hooker : Traveller and plant collector* (1st ed.). Woodbridge: Antique Collectors' Club with Royal Botanic Gardens, Kew.

Dwarakanath, M. H., & Nowitz, L. (1980). An Application of Linear Graph Theory for Coordination of Directional Overcurrent Relays. In *Proc. of elect. power problems - the mathematical challenge, siam meeting* (p. 104-114).

Eiben, A., & Smith, J. (2003). *Introduction to evolutionary computing*. Berlin, Heidel-

berg: Springer Berlin Heidelberg.

Elmore, W. A. (1994). *Protective relaying: Theory and applications.* New York: ABB - Marcel Dekker Inc.

Elrafie, H. B., & Irving, M. R. (1993, Aug.). Linear Programming for Directional Overcurrent Relay Coordination in Interconnected Power Systems with Constraint Relaxation. *Electric Power Systems Research, 27*(3), 209 - 216. Retrieved from http://www.sciencedirect.com/science/article/pii/037877969390047I doi: http://dx.doi.org/10.1016/0378-7796(93)90047-I

Emami, Z., Farahi, M. H., & Ansari, E. (2012, Feb.). Global Optimization with Electro Weak. *The Open Industrial and Manufacturing Engineering Journal, 5*, 1-7. Retrieved from http://benthamscience.com/open/toimej/articles/V005/1TOIMEJ.pdf

Ergezer, M., Simon, D., & Du, D. (2009, Oct.). Oppositional Biogeography-Based Optimization. In *Systems, man and cybernetics, 2009. smc 2009. ieee international conference on* (p. 1009-1014). doi: 10.1109/ICSMC.2009.5346043

Ezzeddine, M., & Kaczmarek, R. (2011a, Oct.). A New Algorithm to Avoid the Infeasibility of Linear Programming Techniques Used to Solve the Over-Current Relays Coordination Problem. *Electric Power Components and Systems, 39*(14), 1596-1607. Retrieved from http://www.tandfonline.com/doi/abs/10.1080/15325008.2011.596500 doi: 10.1080/15325008.2011.596500

Ezzeddine, M., & Kaczmarek, R. (2011b, July). A Novel Method for Optimal Coordination of Directional Overcurrent Relays Considering Their Available Discrete Settings and Several Operation Characteristics. *Electric Power Systems Research, 81*(7), 1475 - 1481. Retrieved from http://www.sciencedirect.com/science/article/pii/S0378779611000630 doi: http://dx.doi.org/10.1016/j.epsr.2011.02.014

Fletcher, R. (1987). *Practical methods of optimization* (2nd ed.). Chichester, New York: John Wiley & Sons, Inc.

Floreano, D., & Mattiussi, C. (2008). *Bio-inspired artificial intelligence: Theories,*

methods, and technologies (intelligent robotics and autonomous agents series). Cambridge, Massachusetts: The MIT Press.

Forst, W., & Hoffmann, D. (2010). *Optimization– theory and practice.* New York: Springer.

Gavana, A. (2013, Feb.). *Test Functions Index.* Retrieved from http://infinity77.net/global_optimization/test_functions.html ([Accessed April 01, 2013])

Gendreau, M., & Potvin, J.-Y. (2010). *Handbook of metaheuristics.* Springer.

Gers, J. M., & Holmes, E. J. (2004). *Protection of electricity distribution networks* (2nd ed.). London, UK: Institution Of Engineering And Technology.

Glover, J. D., Sarma, M. S., & Overbye, T. J. (2010). *Power system analysis and design with cd-rom - si version* (4th ed.). Stamford, CT: Cengage Learning.

Gong, W., Cai, Z., & Ling, C. (2010). DE/BBO: A Hybrid Differential Evolution with Biogeography-Based Optimization for Global Numerical Optimization. *Soft Computing, 15*(4), 645-665. Retrieved from http://dx.doi.org/10.1007/s00500-010-0591-1 doi: 10.1007/s00500-010-0591-1

Gong, W., Cai, Z., Ling, C. X., & Li, H. (2010, Jul.). A Real-Coded Biogeography-Based Optimization with Mutation. *Applied Mathematics and Computation, 216*(9), 2749 - 2758. Retrieved from http://www.sciencedirect.com/science/article/pii/S0096300310003887 doi: http://dx.doi.org/10.1016/j.amc.2010.03.123

Grainger, J. J., & Stevenson, W. D. (1994). *Power system analysis.* New York: McGraw-Hill Education.

Guo, W., Chen, M., Wang, L., Ge, S. S., & Wu, Q. (2013). *Design of Migration Operators for Biogeography-Based Optimization and Markov Analysis.* (Manuscript submitted for publication [Online]. Available: http://embeddedlab.csuohio.edu/BBO/BBO_Papers/GuoMigration.pdf)

Guo, W., Wang, L., & Wu, Q. (2014). An Analysis of the Migration Rates for Biogeography-Based Optimization. *Information Sciences, 254*(0), 111 - 140. Retrieved from http://www.sciencedirect.com/science/

article/pii/S002002551300529X doi: http://dx.doi.org/10.1016/j.ins.2013.07.018

Hadj-Alouane, A. B., & Bean, J. C. (1997). A Genetic Algorithm for the Multiple-Choice Integer Program. *Operations Research, 45*(1), 92–101.

Hewitson, L., Brown, M., & Balakrishnan, R. (2004). *Practical power system protection.* Oxford Burlington, MA: Newnes.

Hodgkiss, J. W. (1995). Overcurrent Protection. In T. E. T. Association (Ed.), *Power System Protection, vol.2: Systems and Methods* (p. 1 - 65). Cambridge, UK: IEE-Cambride University Press.

Hui, W. (2006). *Protection Coordination in Power System with Distributed Generations* (Doctoral dissertation, The Hong Kong Polytechnic University, Hung Hom, kowloon). Retrieved from http://repository.lib.polyu.edu.hk/jspui/bitstream/10397/3485/2/b20940440_ir.pdf ([Accessed Nov. 19, 2013])

Husain, A. (2007). *Electrical power sytems* (5th ed.). New Delhi, India: CBS Publishers & Distributors.

IEC. (1989). Single Input Energizing Quality Measuring Relays With Dependent or Independent. *IEC Publication 255-3.*

IEEE. (1999, Jul.). IEEE Standard Inverse-Time Characteristic Equations for Overcurrent Relays. *IEEE Standard C37.112-1996, 14*(3), 868-872. doi: 10.1109/61.772326

Igelnik, B., & Simon, D. (2011, Sept.). The Eigenvalues of a Tridiagonal Matrix in Biogeography. *Applied Mathematics and Computation, 218*(1), 195 - 201. Retrieved from http://www.sciencedirect.com/science/article/pii/S0096300311007247 doi: http://dx.doi.org/10.1016/j.amc.2011.05.054

Jenkins, L., Khincha, H. P., Shivakumar, S., & Dash, P. (1992, Jan.). An Application of Functional Dependencies to the Topological Analysis of Protection Schemes. *Power Delivery, IEEE Transactions on, 7*(1), 77-83. doi: 10.1109/61.108892

Johal, N. K., Singh, S., & Kundra, H. (2010, Sep.). A Hybrid FPAB/BBO Algorithm for Satellite Image Classification. *International Journal of Computer Ap-*

plications, *6*(5), 31-36. Retrieved from http://dx.doi.org/10.5120/1074-1403

Johns, A. T., & Salman, S. K. (1995). *Digital protection for power systems*. London, UK: P. Peregrinus on behalf of the Institution of Electrical Engineers.

Johnson, R. A. (2000). *Miller and freund's probability and statistics for engineers* (6th ed.). Upper Saddle River, NJ: Prentice Hall Inc.

Jones, R. (1980). *Biogeography: Structure, process, pattern, and change within the biosphere*. Hulton Educational Publications Limited.

Jr., D. R. M. (2010). *The Uses of the Slime Mold Lifecycle as a Model for Numerical Optimization* (Doctoral dissertation, Oklahoma State University, Oklahoma City, OK). Retrieved from http://www.cs.okstate.edu/~monismi/dissertation/MonismithDissertation.pdf ([Accessed Apr. 30, 2013])

Kajee-Bagdadi, Z. (2007). *Differential Evolution Algorithms for Constrained Global Optimization* (Master's thesis, University of the Witwatersrand, Johannesburg, South Africa). Retrieved from http://wiredspace.wits.ac.za/bitstream/handle/10539/4733/Thesis.pdf ([Accessed Aug. 15, 2013])

Karegar, H. K., Abyaneh, H. A., Ohis, V., & Meshkin, M. (2005, Aug.). Pre-Processing of the Optimal Coordination of Overcurrent Relays. *Electric Power Systems Research*, *75*(2-3), 134-141. Retrieved from http://www.sciencedirect.com/science/article/pii/S037877960500129X doi: http://dx.doi.org/10.1016/j.epsr.2005.02.005

Kearfott, B. (1979). An Efficient Degree-Computation Method for a Generalized Method of Bisection. *Numerische Mathematik*, *32*(2), 109-127.

Kiseleva, E., & Stepanchuk, T. (2003, Feb.). On the Efficiency of a Global Non-Differentiable Optimization Algorithm Based on the Method of Optimal Set Partitioning. *Journal of Global Optimization*, *25*(2), 209-235.

Knable, A. H. (1961). A Computer Approach to Setting Directional Over-Current Relays on Industrial Network Type Systems. In *Ieee cp-61-933*.

Knable, A. H. (1969). A Standardized Approach to Relay Coordination. In *Ieee winter power meeting, 69 cp 58.*

Kothari, D. P., & Dhillon, J. S. (2011). *Power system optimization* (2nd ed.). New Delhi, India: PHI Learning Private Limited.

Kundra, H., & Sood, M. (2010, Sep.). Cross-Country Path Finding Using Hybrid Approach of PSO and BBO. *International Journal of Computer Applications*, *7*(6), 15-19. Retrieved from http://dx.doi.org/10.5120/1167-1370

Kundu, D., & Mitra, A. (1998, May). Estimating the Parameters of the Linear Compartment Model. *Journal of Statistical Planning and Inference*, *70*(2), 317-334.

Lee, J. B., & Lee, B. C. (1996). A Global Optimization Algorithm Based on the New Filled Function Method and the Genetic Algorithm. *Engineering Optimization*, *27*(1), 1-20.

Lee, K. Y., & El-Sharkawi, M. A. (Eds.). (2008). *Modern heuristic optimization techniques: Theory and applications to power systems (ieee press series on power engineering).* Piscataway, N.J. Hoboken, N.J: IEEE Press Wiley-Interscience.

Leondes, C. T. (1998). *Optimization techniques, volume 2 (neural network systems techniques and applications) (pt. 2).* San Diego: Academic Press.

Lincoln, R. J., Boxshall, G. A., & Clark, P. F. (1982). *A dictionary of ecology, evolution and systematics* (1st ed.). Cambridge University Press.

Lohokare, M., Pattnaik, S., Panigrahi, B., & Das, S. (2013, May). Accelerated Biogeography-Based Optimization with Neighborhood Search for Optimization. *Applied Soft Computing*, *13*(5), 2318 - 2342. Retrieved from http://www.sciencedirect.com/science/article/pii/S1568494613000392 doi: http://dx.doi.org/10.1016/j.asoc.2013.01.020

Lomolino, M. V., Riddle, B. R., & Brown, J. H. (2009). *Biogeography* (3rd. ed.). Sunderland, Mass: Sinauer Associates, Incorporated.

Losos, J. B., & Ricklefs, R. E. (2010). *The theory of island biogeography revisited.* Princeton: Princeton University Press.

Luke, S. (2009). *Essentials of metaheuristics.* Available for free at http://cs.gmu.edu/~sean/book/metaheuristics/: Lulu.

Ma, H. (2010, Sept.). An Analysis of the Equilibrium of Migration Models for Biogeography-Based Optimization. *Information Sciences, 180*(18), 3444 - 3464. Retrieved from http://www.sciencedirect.com/science/article/pii/S0020025510002446 doi: http://dx.doi.org/10.1016/j.ins.2010.05.035

Ma, H., Fei, M., Ding, Z., & Jin, J. (2012, June). Biogeography-Based Optimization with Ensemble of Migration Models for Global Numerical Optimization. In *Evolutionary computation (cec), 2012 ieee congress on* (p. 1-8). doi: 10.1109/CEC.2012.6252930

Ma, H., Ni, S., & Sun, M. (2009, Dec.). Equilibrium Species Counts and Migration Model Tradeoffs for Biogeography-Based Optimization. In *Decision and control, 2009 held jointly with the 2009 28th chinese control conference. cdc/ccc 2009. proceedings of the 48th ieee conference on* (p. 3306-3310). doi: 10.1109/CDC.2009.5400004

Ma, H., & Simon, D. (2011a). Analysis of Migration Models of Biogeography-Based Optimization Using Markov Theory. *Engineering Applications of Artificial Intelligence, 24*(Jun.), 1052 - 1060. Retrieved from http://www.sciencedirect.com/science/article/pii/S0952197611000789 doi: http://dx.doi.org/10.1016/j.engappai.2011.04.012

Ma, H., & Simon, D. (2011b). Blended Biogeography-Based Optimization for Constrained Optimization. *Engineering Applications of Artificial Intelligence, 24*(March), 517 - 525. Retrieved from http://www.sciencedirect.com/science/article/pii/S0952197610001600 doi: http://dx.doi.org/10.1016/j.engappai.2010.08.005

Ma, H., Simon, D., Fei, M., & Xie, Z. (2013, Jan.). Variations of Biogeography-Based Optimization and Markov Analysis. *Information Sciences, 220*(0), 492 - 506. Retrieved from http://www.sciencedirect.com/science/article/pii/S0020025512004719 (Online Fuzzy Machine Learning and Data Mining) doi: http://dx.doi.org/10.1016/j.ins.2012.07.007

MacArthur, R. H. (1972). *Geographical ecology: Patterns in the distribution of species.* New York: Harper & Row Publishers Inc.

MacArthur, R. H., & Connell, J. H. (1966). *The biology of populations.* New York: John Wiley & Sons Inc.

MacArthur, R. H., & Wilson, E. O. (1963, Dec.). An Equilibrium Theory of Insular Zoogeography. *Evolution, 17*(4), 373–387. Retrieved from http://dx.doi.org/10.2307/2407089 doi: 10.2307/2407089

MacArthur, R. H., & Wilson, E. O. (1967). *The theory of island biogeography.* Princeton University Press.

MacDonald, G. M. (2003). *Biogeography: Introduction to space, time, and life.* New York: John Wiley & Sons Inc.

Madsen, K. (2008, Jun.). *Test Problems for Global Optimization.* Retrieved from http://www2.imm.dtu.dk/~kajm/Test_ex_forms/test_ex.html ([Accessed April 09, 2013])

Maiti, P., & Maiti, P. K. (2011). *Biodiversity: Perception, peril and preservation.* New Delhi: Prentice-Hall of India Pvt.Ltd.

Marriott, K., & Stuckey, P. (1998). *Programming with constraints: An introduction.* The MIT Press.

Mason, C. R. (1956). *The art and science of protective relaying.* New York: Wiley, Chapman & Hall.

MathWorks. (2013). *Global Optimization Toolbox.* Retrieved from http://www.mathworks.com/products/global-optimization/index.html ([Accessed April 25, 2013])

McCleer, P. J. (1987). *The theory and practice of overcurrent protection.* Jackson, Mich: Mechanical Products Inc.

McMahon, J. A. (1987). Disturbed Lands and Ecological Theory: An Essay about a Mutualistic Association. In J. Fagerberg, D. Mowery, & R. Nelson (Eds.), *Restoration ecology, a synthetic approach to ecological research* (p. 221 - 237). Cambridge, UK: Cambride University Press.

Michalewicz, Z. (1995). A survey of Constraint Handling Techniques in Evolutionary

Computation Methods. In *Proceedings of the 4th annual conference on evolutionary programming* (pp. 135–155). MIT Press.

Mishra, S. (2006a, Oct.). *Repulsive Particle Swarm Method on Some Difficult Test Problems of Global Optimization.* Shillong, India. Retrieved from http://mpra.ub.uni-muenchen.de/1742/1/MPRA_paper_1742.pdf

Mishra, S. (2006b, Aug.). *Some New Test Functions for Global Optimization and Performance of Repulsive Particle Swarm Method.* Shillong, India. Retrieved from http://mpra.ub.uni-muenchen.de/2718/1/MPRA_paper_2718.pdf

Mishra, S. (2012, Aug.). *Global Optimization of Some Difficult Benchmark Functions by Cuckoo-Host Co-Evolution Meta-Heuristics.* Shillong, India. Retrieved from http://mpra.ub.uni-muenchen.de/40666/1/MPRA_paper_40666.pdf

Mohammed, Z., & Talaq, J. (2012, March). Unit Commitment by Biogeography Based Optimization Method. In *Electrotechnical conference (melecon), 2012 16th ieee mediterranean* (p. 551-554). doi: 10.1109/MELCON.2012.6196494

Molga, M., & Smutnicki, C. (2005, Apr.). *Test Functions for Optimization Needs.* Retrieved from http://www.zsd.ict.pwr.wroc.pl/files/docs/functions.pdf ([Accessed March 28, 2013])

Momoh, J. A. (2001). *Electric power system applications of optimization (power engineering (willis)).* New York: Marcel Dekker, Inc.

Moravej, Z., Jazaeri, M., & Gholamzadeh, M. (2012, Apr.). Optimal Coordination of Distance and Over-Current Relays in Series Compensated Systems Based on MAPSO. *Energy Conversion and Management, 56*, 140–151. doi: 10.1016/j.enconman.2011.11.024

Moré, J. J., Garbow, B. S., & Hillstrom, K. E. (1981). Testing Unconstrained Optimization Software. *ACM Transactions on Mathematical Software, 7*(1), 17–41. Retrieved from http://doi.acm.org/10.1145/355934.355936 doi: 10.1145/355934.355936

Müller, K., & Brown, L. D. (1979). Location of Saddle Points and Minimum Energy

Paths by a Constrained Simplex Optimization Procedure. *Theoretica chimica acta, 53*(1), 75-93.

Myers, A. A., & Giller, P. S. (Eds.). (1990). *Analytical biogeography: An integrated approach to the study of animal and plant distributions* (1st ed.). London, UK: Chapman and Hall.

Nierenberg, W. A. (Ed.). (1995). *Encyclopedia of environmental biology, three-volume set: 1-3* (1st ed.). Academic Press.

Nocedal, J., & Wright, S. (2006). *Numerical optimization* (2nd ed.). New York: Springer.

Noghabi, A., Sadeh, J., & Mashhadi, H. (2009, Oct.). Considering Different Network Topologies in Optimal Overcurrent Relay Coordination Using a Hybrid GA. *Power Delivery, IEEE Transactions on, 24*(4), 1857-1863. doi: 10.1109/TPWRD.2009.2029057

of Technology, L. U. (2007, May). *The Function Testbed.* Retrieved from `http://www.it.lut.fi/ip/evo/functions/functions.html` ([Accessed April 10, 2013])

Onbaşoğlu, E., & ʻOzdamar, L. (2001, Jan.). Parallel Simulated Annealing Algorithms in Global Optimization. *Journal of Global Optimization, 19*(1), 27-50.

Padiyar, K. R. (2008). *Power system dynamics: Stability and control* (2nd ed.). Hyderabad, India: BSP, BS Publications.

Paithankar, Y., & Bhide, S. (2003). *Fundamentals of power system protection.* New Delhi: Prentice-Hall of India Pvt. Ltd.

Panchal, V., Goel, S., & Bhatnagar, M. (2009, Dec.). Biogeography Based Land Cover Feature Extraction. In *Nature biologically inspired computing, 2009. nabic 2009. world congress on* (p. 1588-1591). doi: 10.1109/NABIC.2009.5393653

Pattnaik, S., Lohokare, M., & Devi, S. (2010, Dec.). Enhanced Biogeography-Based Optimization Using Modified Clear Duplicate Operator. In *Nature and biologically inspired computing (nabic), 2010 second world congress on* (p. 715-720). doi: 10.1109/NABIC.2010.5716322

Price, K. V., Storn, R. M., & Lampinen, J. A. (2005). *Differential evolution: A practical*

approach to global optimization (natural computing series). Berlin, Germany: Springer Science+Business Media.

Qin, A. K., Huang, V. L., & Suganthan, P. (2009, April). Differential Evolution Algorithm With Strategy Adaptation for Global Numerical Optimization. *Evolutionary Computation, IEEE Transactions on, 13*(2), 398-417. doi: 10.1109/TEVC.2008.927706

Qing, A. (2009). *Differential evolution: Fundamentals and applications in electrical engineering* (1st ed.). Singapore Hoboken, NJ Piscataway, NJ: Wiley-IEEE Press. doi: 10.1002/9780470823941

Radke, G. E. (1963). A Method for Calculating Time-Overcurrent Relay Settings by Digital Computer. *Power Apparatus and Systems, IEEE Transactions on, PAS-82*(Special Supplement), 189-205.

Rahnamayan, S., Tizhoosh, H., & Salama, M. (2006, Jul.). Opposition-Based Differential Evolution for Optimization of Noisy Problems. In *Evolutionary computation, 2006. cec 2006. ieee congress on* (p. 1865-1872). doi: 10.1109/CEC.2006.1688534

Rahnamayan, S., Tizhoosh, H. R., & Salama, M. M. A. (2007, May). A Novel Population Initialization Method for Accelerating Evolutionary Algorithms. *Computers & Mathematics with Applications, 53*(10), 1605-1614.

Ramadas, G. C., & Fernandes, E. M. (2011, Jun.). Solving Nonlinear Equations by a Tabu Search Strategy. In *11th international conference on computational and mathematical methods in science and engineering, cmmse2011* (Vol. 4, p. 1578-1589).

Rao, S. S. (2008). *Switchgear protection and power systems: Theory, practice & solved problems* (13th ed. ed.). Delhi, India: Khanna Publishers.

Rao, S. S. (2009). *Engineering optimization: Theory and practice* (4th ed.). Hoboken, New Jersey: John Wiley & Sons.

Rebizant, W., Szafran, J., & Wiszniewski, A. (2011). *Digital signal processing in power system protection and control*. London, UK: Springer.

Saadat, H. (1999). *Power system analysis*. Singapore: WCB/McGraw-Hill.

Santos, C. A. G., Freire, P. K. M. M., & Mishra, S. K. (2012, Dec.). Cuckoo Search via Lévy Flights for Optimization of a Physically-Based Runoff-Erosion Model. *Journal of Urban and Environmental Engineering*, *6*(2), 123-131.

Shermer, M. (2011). *In darwin's shadow: The life and science of alfred russel wallace: A biographical study on the psychology of history.* Oxford New York: Oxford University Press, Incorporated.

Simon, D. (2008a, Dec.). Biogeography-Based Optimization. *Evolutionary Computation, IEEE Transactions on*, *12*(6), 702-713. doi: 10.1109/TEVC.2008.919004

Simon, D. (2008b, Aug.). *The Matlab Code of Biogeography-Based Optimization.* Retrieved from `http://academic.csuohio.edu/simond/bbo/` ([Accessed Feb. 1, 2013])

Simon, D. (2011, June). A Probabilistic Analysis of a Simplified Biogeography-Based Optimization Algorithm. *Evolutionary Computation*, *19*(2), 167–188. Retrieved from `http://dx.doi.org/10.1162/EVCO_a_00018` doi: 10.1162/EVCO_a_00018

Simon, D. (2013). *Evolutionary optimization algorithms : Biologically-inspired and population-based approaches to computer intelligence.* Hoboken, New Jersey: John Wiley & Sons Inc.

Simon, D., Ergezer, M., & Du, D. (2009, Oct.). Population Distributions in Biogeography-Based Optimization Algorithms with Elitism. In *Systems, man and cybernetics, 2009. smc 2009. ieee international conference on* (p. 991-996). doi: 10.1109/ICSMC.2009.5346058

Singh, M., Panigrahi, B., & Abhyankar, A. (2013). Optimal Coordination of Directional Over-Current Relays Using Teaching Learning-Based Optimization (TLBO) Algorithm. *International Journal of Electrical Power & Energy Systems*, *50*, 33 - 41. doi: http://dx.doi.org/10.1016/j.ijepes.2013.02.011

Sivananaithaperumal, S., Amali, S. M. J., Baskar, S., & Suganthan, P. (2011, Sept.). Constrained Self-Adaptive Differential Evolution Based Design of Robust Optimal Fixed Structure Controller. *Engineering Applications of Artificial Intelligence*, *24*(6), 1084 - 1093. Retrieved from `http://www.sciencedirect`

.com/science/article/pii/S0952197611000856 doi: http://dx.doi.org/10.1016/j.engappai.2011.05.003

Sood, M., & Kaur, M. (2012, Feb.). Shortest Path Finding in Country Using Hybrid Approach of BBO and BCO. *International Journal of Computer Applications*, *40*(6), 9-13. Retrieved from http://dx.doi.org/10.5120/4958-7215

Storn, R., & Price, K. (1997, Dec.). Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*, *11*(4), 341-359.

Subotic, M., Tuba, M., & Stanarevic, N. (2011, Mar.). Different Approaches in Parallelization of the Artificial Bee Colony Algorithm. *International Journal of Mathematical Models and Methods in Applied Sciences*, *5*(4), 755-762.

Sulloway, F. (1982, March). Darwin and his Finches: The Evolution of a Legend. *Journal of the History of Biology*, *15*(1), 1-53. Retrieved from http://dx.doi.org/10.1007/BF00132004 doi: 10.1007/BF00132004

Svirezhev, Y. M., & Logofet, D. O. (1983). *Stability of biological communities.* Moscow, Soviet Union: Mir Publishers.

Talbi, E.-G. (2009). *Metaheuristics: From design to implementation.* Hoboken, New Jersey: John Wiley & Sons, Inc.

Thangaraj, R., Pant, M., & Deep, K. (2010, Aug.). Optimal Coordination of Over-Current Relays Using Modified Differential Evolution Algorithms. *Engineering Applications of Artificial Intelligence*, *23*(5), 820–829. Retrieved from http://www.sciencedirect.com/science/article/pii/S0952197610000424 (Advances in metaheuristics for hard optimization: new trends and case studies) doi: http://dx.doi.org/10.1016/j.engappai.2010.01.024

Trabia, M. B., & Lu, X. B. (2001, Jun.). A Fuzzy Adaptive Simplex Search Optimization Algorithm. *Journal of Mechanical Design*, *123*(2), 1-10.

Tsien, H. Y. (1964, Oct.). An Automatic Digital Computer Program for Setting Transmission Line Directional Overcurrent Relays. *Power Apparatus and Systems,*

IEEE Transactions on, PAS-83(10), 1048-1053.

Urdaneta, A., Nadira, R., & Perez Jimenez, L. (1988, Jul.). Optimal Coordination of Directional Overcurrent Relays in Interconnected Power Systems. *Power Delivery, IEEE Transactions on, 3*(3), 903-911. doi: 10.1109/61.193867

Urdaneta, A., Perez, L., & Restrepo, H. (1997, Oct.). Optimal Coordination of Directional Overcurrent Relays Considering Dynamic Changes in the Network Topology. *Power Delivery, IEEE Transactions on, 12*(4), 1458-1464. doi: 10.1109/61.634161

Venkataraman, P. (2009). *Applied optimization: with matlab programming* (2nd ed.). Hoboken, N.J: John Wiley & Sons.

Vrahatis, M., Sotiropoulos, D., & Triantafyllou, E. (1997). Global Optimization for Imprecise Problems. In I. Bomze, T. Csendes, R. Horst, & P. Pardalos (Eds.), *Developments in global optimization* (Vol. 18, p. 37-54). Springer US. Retrieved from `http://dx.doi.org/10.1007/978-1-4757-2600-8_3` doi: 10.1007/978-1-4757-2600-8_3

Wan, W., & Birch, J. B. (2011, Dec.). Using a Modified Genetic Algorithm to Find Feasible Regions of a Desirability Function. *Quality and Reliability Engineering International, 27*(8), 1173-1182.

Wood, H. C., Sidhu, T. S., Sachdev, M. S., & Nagpal, M. (1989, Sept.). A General Purpose Hardware for Microprocessor Based Relays. In *International conference on power system protection '89* (p. 43-59). 1989. Retrieved from `http://www.eng.uwo.ca/people/tsidhu/Documents/General%20purpose%20hardware.pdf`

Yao, X., Liu, Y., & Lin, G. (1999, Jul.). Evolutionary Programming Made Faster. *Evolutionary Computation, IEEE Transactions on, 3*(2), 82-102. doi: 10.1109/4235.771163

Yeniay, Ö. (2005). Penalty Function Methods for Constrained Optimization with Genetic Algorithms. *Mathematical and Computational Applications, 10*, 45–56.

Yu, X., & Gen, M. (2010). *Introduction to evolutionary algorithms*. London, UK: Springer Science+Business Media.

Zeineldin, H., El-Saadany, E., & Salama, M. (2006). Optimal Coordination of Overcurrent Relays Using a Modified Particle Swarm Optimization. *Electric Power Systems Research*, *76*(11), 988 - 995. Retrieved from http://www.sciencedirect.com/science/article/pii/S0378779605002701 doi: http://dx.doi.org/10.1016/j.epsr.2005.12.001

Zhao, X., & Gao, X.-S. (2007, Apr.). Affinity Genetic Algorithm. *Journal of Heuristics*, *13*(2), 133-150.

Zitko, V. (1984, Sept.). *Simplex Optimization* (Tech. Rep. No. 1308). Fisheries and Oceans. Biological Station (St. Andrews, N.B.), Canada: Canadian Technical Report of Fisheries and Aquatic Sciences. Retrieved from http://www.dfo-mpo.gc.ca/Library/20701.pdf

List of Appendices

Appendix A

This Appendix contains the derivation of the species colonization curve on an island:

$$\begin{aligned} \frac{dS(t)}{dt} &= \lambda_s - \mu_s = I - \left(\frac{I}{S_{max}}\right) S(t) - \left(\frac{E}{S_{max}}\right) S(t) \\ S_{max}\frac{dS(t)}{dt} &= I\left[S_{max} - S(t)\right] - ES(t) \\ S_{max}\frac{dS(t)}{dt} &+ (I+E)S(t) = IS_{max} \end{aligned}$$

By applying laplace transform:

$$\begin{aligned} \mathscr{L}\left\{S_{max}\frac{dS(t)}{dt}\right\} &+ \mathscr{L}\left\{(I+E)S(t)\right\} = \mathscr{L}\left\{IS_{max}\right\} \\ S_{max}[sS(s) - S(0)] &+ (I+E)S(s) = \frac{IS_{max}}{s} \end{aligned}$$

Because at the beginning $(t = 0)$, $S(t) = 0 \rightarrow S(0) = 0$:

$$\begin{aligned} s\left[S_{max}s + (I+E)\right]S(s) &= IS_{max} \\ S(s) = \frac{IS_{max}}{s\left[S_{max}s + (I+E)\right]} &= \frac{\left(\frac{IS_{max}}{I+E}\right)}{s\left[\left(\frac{S_{max}}{I+E}\right)s + 1\right]} = \frac{1}{s\left[\left(\frac{S_{max}}{I+E}\right)s + 1\right]}\hat{S} \end{aligned}$$

By applying inverse laplace transform:

$$\mathscr{L}^{-}\left\{S(s)\right\} = \mathscr{L}^{-}\left\{\frac{1}{s\left[\left(\frac{S_{max}}{I+E}\right)s + 1\right]}\hat{S}\right\} \Rightarrow S(t) = \hat{S}\left[1 - e^{-\left(\frac{I+E}{S_{max}}\right)t}\right]$$

Appendix B

This Appendix contains the first proof of the iterative variable bounds algorithms: Substituting Eq.(4.31) in Eq.(4.28) and Eq.(4.29) yields:

$$\begin{aligned} SIV_{min,s}^{new} &= SIV_{min,s}^{old} + SF\left(SIV_{s,1} - SIV_{min,s}^{old}\right) \\ SIV_{max,s}^{new} &= SIV_{max,s}^{old} - SF\left(SIV_{max,s}^{old} - SIV_{s,1}\right) \end{aligned}$$

If $SF = 100\% = 1.0$:

$$\begin{aligned} SIV_{min,s}^{new} &= SIV_{min,s}^{old} + SIV_{s,1} - SIV_{min,s}^{old} = SIV_{s,1} \\ SIV_{max,s}^{new} &= SIV_{max,s}^{old} - SIV_{max,s}^{old} + SIV_{s,1} = SIV_{s,1} \\ \Delta SIV_{s}^{new} &= SIV_{max,s}^{new} - SIV_{min,s}^{new} = SIV_{s,1} - SIV_{s,1} = 0 \end{aligned}$$

If $SF = 0\% = 0.0$:

$$\begin{aligned} SIV_{min,s}^{new} &= SIV_{min,s}^{old} \\ SIV_{max,s}^{new} &= SIV_{max,s}^{old} \\ \Delta SIV_{s}^{new} &= \Delta SIV_{s}^{old} \end{aligned}$$

If $SF = 98\% = 0.98$:

$$\begin{aligned} SIV_{min,s}^{new} &= SIV_{min,s}^{old} + SF\left(SIV_{s,1}\right) - SF\left(SIV_{min,s}^{old}\right) \\ SIV_{max,s}^{new} &= SIV_{max,s}^{old} - SF\left(SIV_{max,s}^{old}\right) + SF\left(SIV_{s,1}\right) \\ \Delta SIV_{s}^{new} &= (1 - SF)\,\Delta SIV_{s}^{old} = 0.02\Delta SIV_{s}^{old} \end{aligned}$$

Appendix C

This Appendix contains the second proof of the iterative variable bounds algorithms:

$$\begin{aligned}
R &= \frac{G_1}{G_T}, \text{ or } G_1 = RG_T \\
G_l &= G_1 e^{-\beta(l-1)} \\
G_T &= \sum_{l=1}^{L} G_l = G_1 + \sum_{l=2}^{L} G_l = RG_T + \sum_{l=2}^{L} G_l \\
\sum_{l=2}^{L} G_l &= \sum_{l=2}^{L} G_1 e^{-\beta(l-1)} = (1-R)G_T \\
\sum_{l=2}^{L} e^{-\beta(l-1)} &= \sum_{l-1=1}^{L} e^{-\beta(l-1)} = \frac{(1-R)G_T}{RG_T} = \frac{1-R}{R}
\end{aligned}$$

Let $x = l - 1$, then:

$$\sum_{x=1}^{L} e^{-\beta x} = \sum_{x=1}^{L} \left(e^{-\beta}\right)^x = \frac{1-R}{R} = \sigma$$

The above summation can be represented as a geometric series. If $r = e^{-\beta}$, then:

$$\begin{aligned}
S_L &= r + r^2 + r^3 + \cdots + r^{L-1} + r^L \\
rS_L &= r^2 + r^3 + r^4 + \cdots + r^L + r^{L+1} \\
(1-r)S_L &= r - r^{L+1} \Rightarrow S_L = \frac{r - r^{L+1}}{1-r} = \frac{e^{-\beta} - \left(e^{-\beta}\right)^{L+1}}{1 - e^{-\beta}}
\end{aligned}$$

$\because$ R is always $+ve$ value and less than one, $0 < R < 1$

$\therefore$ As L becomes large, the term $\left(e^{-\beta}\right)^{L+1}$ will approach 0

$\therefore$ The geometric series will converge to:

$$S_L = \frac{e^{-\beta}}{1 - e^{-\beta}}$$

Now, to find the value of β:

$$\frac{e^{-\beta}}{1 - e^{-\beta}} = \sigma \Rightarrow e^{-\beta} = \sigma - \sigma e^{-\beta} \Rightarrow (1+\sigma)e^{-\beta} = \sigma$$

$$\frac{e^{-\beta}}{1-e^{-\beta}} = \sigma \Rightarrow e^{-\beta} = \sigma - \sigma e^{-\beta} \Rightarrow (1+\sigma)e^{-\beta} = \sigma$$

$$e^{-\beta} = \frac{\sigma}{1+\sigma} = \frac{\frac{1-R}{R}}{1+\frac{1-R}{R}} = \frac{1-R}{R+1-R} = 1-R$$

$$\therefore \beta = -\ln(1-R)$$

$$\text{If } G_1 = \frac{1}{2}G_T \text{ , i.e., } R = 50\% \text{ , then :}$$

$$\beta = -\ln(1-0.5) = \mathbf{0.693147180559945}$$

Appendix D

System 1: 3-Bus System

This system has 3ϕ fault at the midpoint of each line, as shown in Fig.(D1) (Urdaneta et al., 1988; Amraee, 2012).

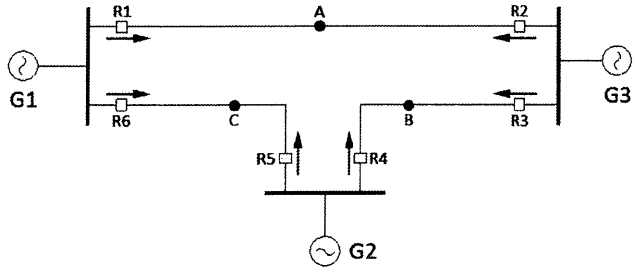


Figure D1: Single-line diagram of the 3-bus system

Table D1: CTR for the relays of the 3-bus system

Relay	R_1	R_2	R_3	R_4	R_5	R_6
CTR	300:5	200:5	200:5	300:5	200:5	400:5

Table D2: 3ϕ faults of the 3-bus system

Primary	**Fault Current (A)**		**Backup**	**Fault Current (A)**	
R_i	**Normal**	**Transient**	R_j	**Normal**	**Transient**
R_1	1978.90	2075.00	R_5	175.00	400.70
R_2	1525.70	1621.70	R_4	545.00	700.64
R_3	1683.90	1779.60	R_1	617.22	760.17
R_4	1815.40	1911.50	R_6	466.17	622.65
R_5	1499.66	1588.50	R_3	384.00	558.13
R_6	1766.30	1855.40	R_2	145.34	380.70

Table D3: Generator data of the 3-bus system

Gen.	S_n (MVA)	V_p (kV)	x (%)
G1	100	69	20
G2	25	69	12
G3	50	69	18

Table D4: Line data of the 3-bus system

Nodes	R (Ω)	X (Ω)	Length (km)
1-2	5.5	22.85	50
2-3	4.4	18.00	40
1-3	7.6	27.00	60

System 2: 4-Bus System

This system has 3ϕ fault at both the near-end and the far-end locations of each line, as shown in Fig.(D2) (Deep et al., 2006; Deep & Bansal, 2009).

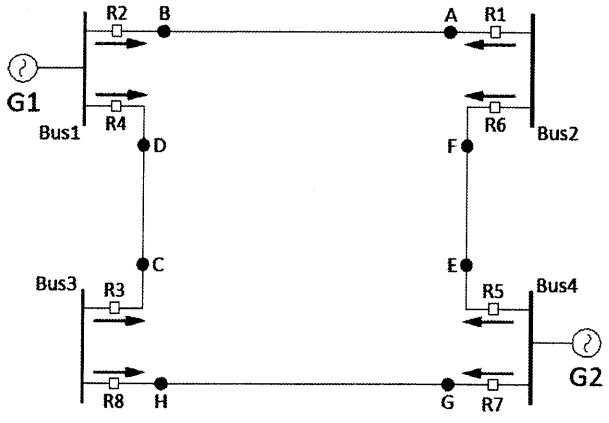


Figure D2: Single-line diagram of the 4-bus system

Table D5: a_p, b_p, c_p and d_p of the 4-bus system

$T_{pr,p}^{near}$			$T_{pr,p}^{far}$		
p	a_p	b_p	q	c_p	d_p
1	20.32	0.48	2	23.75	0.48
2	88.85	0.48	1	12.48	0.48
3	13.61	1.1789	4	31.92	1.1789
4	116.81	1.1789	3	10.38	1.1789
5	116.7	1.5259	6	12.07	1.5259
6	16.67	1.5259	5	31.92	1.5259
7	71.7	1.2018	8	11.00	1.2018
8	19.27	1.2018	7	18.91	1.2018

Table D6: e_p, f_p, g_p and h_p of the 4-bus system

T_{jk}			T_{ik}		
j	e_p	f_p	i	g_p	h_p
5	20.32	1.5259	1	20.32	0.48
5	12.48	1.5259	1	12.48	0.48
7	13.61	1.2018	3	13.61	1.1789
7	10.38	1.2018	3	10.38	1.1789
1	1.16	0.48	4	116.81	1.1789
2	12.07	0.48	6	12.07	1.1789
2	16.67	0.48	6	16.67	1.5259
4	11	1.1789	8	11	1.2018
4	19.27	1.1789	8	19.27	1.2018

System 3: 8-Bus System

This system has 3ϕ fault at the near-end location of each line, as shown in Fig.(D3). Bus 4 is connected to an external grid that is modeled by 400 MVA short-circuit capacity (Noghabi et al., 2009; Zeineldin et al., 2006).

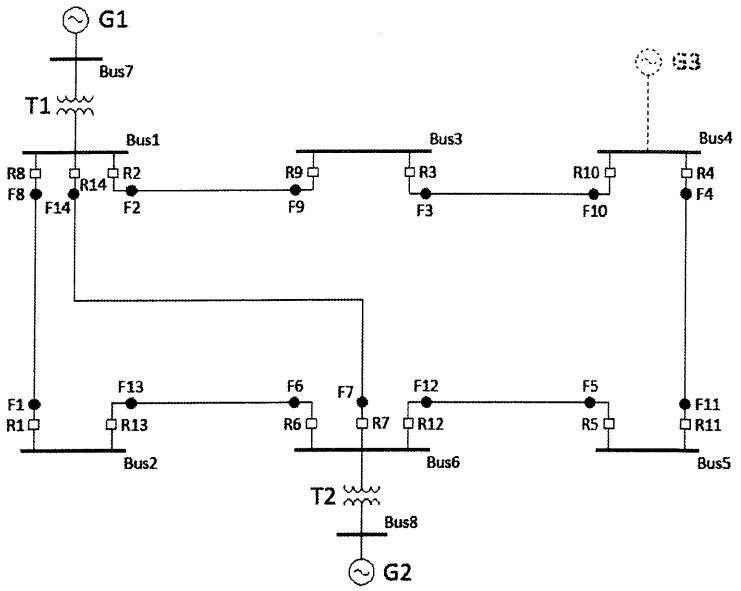


Figure D3: Single-line diagram of the 8-bus system

Table D7: Generator data of the 8-bus system

Gen.	S_n **(MVA)**	V_p **(kV)**	x (%)
G1	150	10	15
G2	150	10	15

Table D8: Transformer data of the 8-bus system

Trans.	S_n **(MVA)**	V_p **(kV)**	V_s **(kV)**	x (%)
T1	150	10	150	4
T2	150	10	150	4

Table D9: Line data of the 8-bus system

Nodes	R **(Ω/km)**	X **(Ω/km)**	Y **(S/km)**	**Length (km)**
1-2	0.004	0.05	0.0	100
1-3	0.0057	0.0714	0.0	70
3-4	0.005	0.0563	0.0	80
4-5	0.005	0.045	0.0	100
5-6	0.0045	0.0409	0.0	110
2-6	0.0044	0.05	0.0	90
1-6	0.005	0.05	0.0	100

Table D10: 3ϕ faults of the 8-bus system

Primary		**Fault Current**	**Backup**	**Fault Current**
R_i	CTR	**(A)**	R_j	**(A)**
R_1	1200:5	3232	R_6	3232
R_2	1200:5	5924	R_1	996
R_2	1200:5	5924	R_7	1890
R_3	800:5	3556	R_2	3556
R_4	1200:5	3783	R_3	2244
R_5	1200:5	2401	R_4	2401
R_6	1200:5	6109	R_5	1197
R_6	1200:5	6109	R_{14}	1874
R_7	800:5	5223	R_5	1197
R_7	800:5	5223	R_{13}	987
R_8	1200:5	6093	R_7	1890
R_8	1200:5	6093	R_9	1165
R_9	800:5	2484	R_{10}	2484
R_{10}	1200:5	3883	R_{11}	2344
R_{11}	1200:5	3707	R_{12}	3707
R_{12}	1200:5	5899	R_{13}	987
R_{12}	1200:5	5899	R_{14}	1874
R_{13}	1200:5	2991	R_8	2991
R_{14}	800:5	5199	R_1	996
R_{14}	800:5	5199	R_9	1165

System 4: 9-Bus System

This system has 3ϕ fault at the midpoint of each line, as shown in Fig.(D4), with no backup protection for relays $\{R_{17}, R_{19}, R_{21}, R_{23}\}$. Bus 1 is supplied by a power source of 100 MVA, 33 kV with a source impedance of (0+j0.1) p.u. All the lines impedances are equal to (0+j0.2) p.u. Also, all the DOCRs have same CTR of $500:1$ (Bedekar & Bhide, 2011).

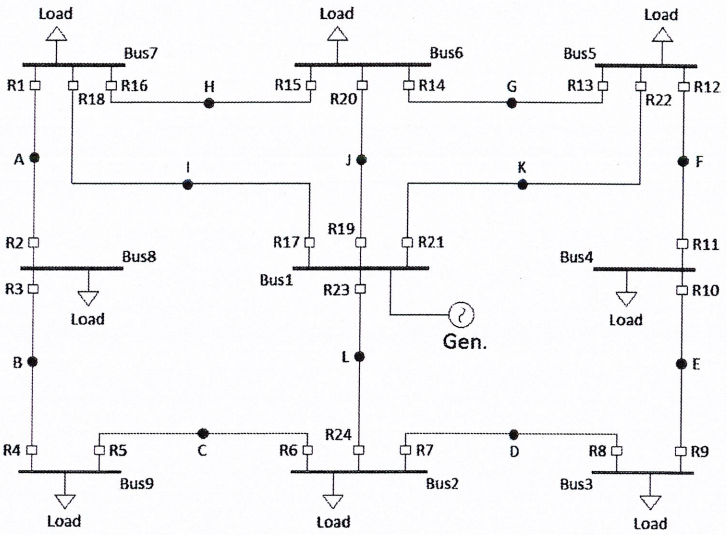


Figure D4: Single-line diagram of the 9-bus system

Table D11: Maximum load currents and 3ϕ fault currents of the 9-bus system

Fault Point	**Primary "R_i"**	**Max. Load "$I_{L_{max}}$" (A)**	**Relay Action Mode**		**Backup(s) "R_j"**
			$I_{pr,i}$ **(A)**	$I_{bc,i}$ **(A)**	
A	R_1	121.74	4863.6	1361.6	R_{15}, R_{17}
	R_2	212.74	1634.4	653.6	R_4
B	R_3	21.74	2811.4	1124.4	R_1
	R_4	21.74	2610.5	1044.2	R_6
C	R_5	78.26	1778.0	711.2	R_3
	R_6	78.26	4378.5	1226.0	R_8, R_{23}
D	R_7	78.26	4378.5	1226.0	R_5, R_{23}
	R_8	78.26	1778.0	711.2	R_{10}
E	R_9	21.74	2610.5	1044.2	R_7
	R_{10}	21.74	2811.4	1124.4	R_{12}
F	R_{11}	121.74	1634.4	653.6	R_9
	R_{12}	121.74	2811.4	787.2	R_{14}, R_{21}
G	R_{13}	30.44	3684.5	1031.7	R_{11}, R_{21}
	R_{14}	30.44	4172.5	1168.3	R_{16}, R_{19}
H	R_{15}	30.44	4172.5	1168.3	R_{13}, R_{19}
	R_{16}	30.44	3684.5	1031.7	R_2, R_{17}
I	R_{17}	441.3	7611.2	1293.9	—
	R_{18}	441.3	2271.7	1953.7	R_2, R_{15}
J	R_{19}	410.87	7435.8	1264.1	—
	R_{20}	410.87	2624.2	2256.8	R_{13}, R_{16}
K	R_{21}	441.3	7611.2	1293.9	—
	R_{22}	441.3	2271.7	1953.7	R_{11}, R_{14}
L	R_{23}	506.52	7914.7	1345.5	—
	R_{24}	506.52	1665.5	1432.3	R_5, R_8

Table D12: Load currents of the 9-bus system

Bus. No.	Bus1	Bus2	Bus3	Bus4	Bus5	Bus6	Bus7	Bus8	Bus9
Load Current (A)	—	350	100	100	350	350	350	100	100

System 5: 15-Bus System

This system has 3ϕ close-in faults on all the lines, as shown in Fig.(D5). All the generators have the same ratings of 15 MVA, 20 kV and a synchronous reactance of $x = 15\%$. Also, all the lines have the same impedance of $Z = 0.19 + j0.46\ \Omega/km$. Bus 8 is connected to an external grid that is modeled by 200 MVA short-circuit capacity. The fault analysis is done based on IEC standard (Amraee, 2012).

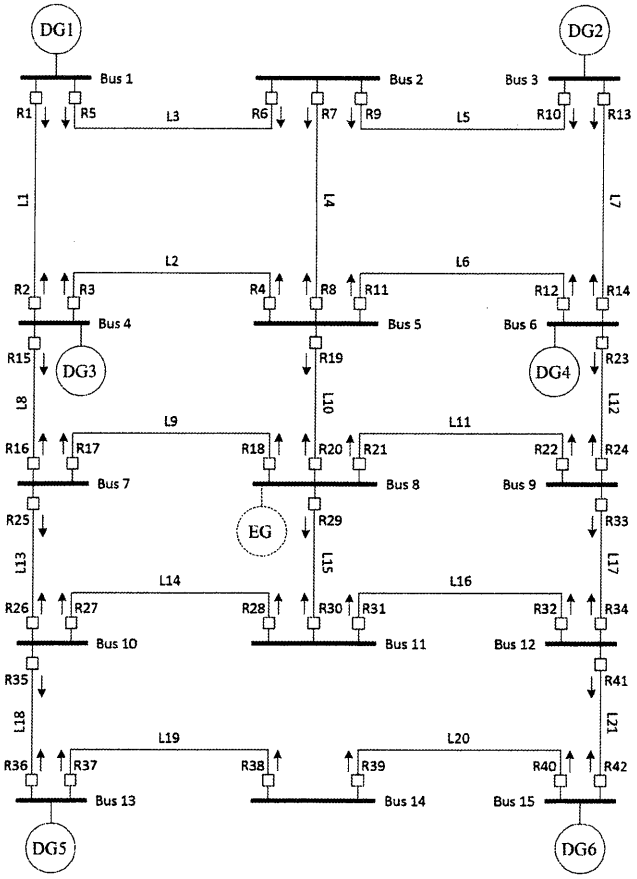


Figure D5: Single-line diagram of the 15-bus system

Table D13: Relays $CTRs$ of the 15-bus system

Relay No.	CTR
18-20-21-29	1600:5
2-4-8-11-12-14-15-23	1200:5
1-3-5-10-13-19-36-37-40-42	800:5
6-7-9-16-24-25-26-27-28-31-32-33-35	600:5
17-22-30-34-38-39-41	400:5

Table D14: 3ϕ faults of the 15-bus system

Primary	I(A)	Backup	I(A)	Primary	I(A)	Backup	I(A)
R_1	3621	R_6	1233	R_{20}	7662	R_{30}	681
R_2	4597	R_4	1477	R_{21}	8384	R_{17}	599
R_2	4597	R_{16}	743	R_{21}	8384	R_{19}	1372
R_3	3984	R_1	853	R_{21}	8384	R_{30}	681
R_3	3984	R_{16}	743	R_{22}	1950	R_{23}	979
R_4	4382	R_7	1111	R_{22}	1950	R_{34}	970
R_4	4382	R_{12}	1463	R_{23}	4910	R_{11}	1475
R_4	4382	R_{20}	1808	R_{23}	4910	R_{13}	1503
R_5	3319	R_2	922	R_{24}	2296	R_{21}	1326
R_6	2647	R_8	1548	R_{24}	2296	R_{34}	970
R_6	2647	R_{10}	1100	R_{25}	2289	R_{15}	969
R_7	2497	R_5	1397	R_{25}	2289	R_{18}	1320
R_7	2497	R_{10}	1100	R_{26}	2300	R_{28}	1192
R_8	4695	R_3	1424	R_{26}	2300	R_{36}	1109
R_8	4695	R_{12}	1463	R_{27}	2011	R_{25}	903
R_8	4695	R_{20}	1808	R_{27}	2011	R_{36}	1109
R_9	2943	R_5	1397	R_{28}	2525	R_{29}	1828
R_9	2943	R_8	1548	R_{28}	2525	R_{32}	697
R_{10}	3568	R_{14}	1175	R_{29}	8346	R_{17}	599
R_{11}	4342	R_3	1424	R_{29}	8346	R_{19}	1372
R_{11}	4342	R_7	1111	R_{29}	8346	R_{22}	642
R_{11}	4342	R_{20}	1808	R_{30}	1736	R_{27}	1039
R_{12}	4195	R_{13}	1503	R_{30}	1736	R_{32}	697
R_{12}	4195	R_{24}	753	R_{31}	2867	R_{27}	1039
R_{13}	3402	R_9	1009	R_{31}	2867	R_{29}	1828
R_{14}	4606	R_{11}	1475	R_{32}	2069	R_{33}	1162
R_{14}	4606	R_{24}	753	R_{32}	2069	R_{42}	907
R_{15}	4712	R_1	853	R_{33}	2305	R_{21}	1326
R_{15}	4712	R_4	1477	R_{33}	2305	R_{23}	979
R_{16}	2225	R_{18}	1320	R_{34}	1715	R_{31}	809
R_{16}	2225	R_{26}	905	R_{34}	1715	R_{42}	907
R_{17}	1875	R_{15}	969	R_{35}	2095	R_{25}	903
R_{17}	1875	R_{26}	905	R_{35}	2095	R_{28}	1192
R_{18}	8426	R_{19}	1372	R_{36}	3283	R_{38}	882
R_{18}	8426	R_{22}	642	R_{37}	3301	R_{35}	910
R_{18}	8426	R_{30}	681	R_{38}	1403	R_{40}	1403
R_{19}	3998	R_3	1424	R_{39}	1434	R_{37}	1434
R_{19}	3998	R_7	1111	R_{40}	3140	R_{41}	745
R_{19}	3998	R_{12}	1463	R_{41}	1971	R_{31}	809
R_{20}	7662	R_{17}	599	R_{41}	1971	R_{33}	1162
R_{20}	7662	R_{22}	642	R_{42}	3295	R_{39}	896

Table D15: The listed lengths of the lines

Line No.	Length
L_2, L_4, L_6, L_{11}, L_{14}, L_{15}, L_{17}, L_{19} and L_{20}	10 km
L_5, L_7, L_9, L_{10}, L_{13} and L_{18}	15 km
L_1, L_8, L_{12}, L_{16} and L_{21}	20 km

Appendix E

This appendix contains the completed list of all the involved test functions that are collected from various sources where some of them are hard to be found while the others have uncompleted information and need to be verified before been used here, because they are available in the literature with rounded global solutions. For more information regarding any test function, please refer to (Jr., 2010; Yao, Liu, & Lin, 1999; Chao, Fand, & Liao, 2012; Kiseleva & Stepanchuk, 2003; Ali, Khompatraporn, & Zabinsky, 2005; Chen, Wang, & Tsai, 2006; Subotic, Tuba, & Stanarevic, 2011; Rao, 2009; Santos, Freire, & Mishra, 2012; Mishra, 2006b; Gavana, 2013; Zhao & Gao, 2007; Adorio, 2005; Chelouah & Siarry, 2000; Emami, Farahi, & Ansari, 2012; Breiman & Cutler, 1993; Molga & Smutnicki, 2005; Rahnamayan, Tizhoosh, & Salama, 2007; Trabia & Lu, 2001; J. B. Lee & Lee, 1996; Costa, Fernandes, & da G. P. Fernandes, 2010; Madsen, 2008; Vrahatis, Sotiropoulos, & Triantafyllou, 1997; Kearfott, 1979; Onbaşoğlu & Özdamar, 2001; Rahnamayan, Tizhoosh, & Salama, 2006; Müller & Brown, 1979; Chou, 2013; MathWorks, 2013; Ramadas & Fernandes, 2011; Moré, Garbow, & Hillstrom, 1981; Aria, 2013; Mishra, 2006a; Zitko, 1984; Storn & Price, 1997; Belegundu & Chandrupatla, 2011; Kundu & Mitra, 1998; Price, Storn, & Lampinen, 2005; Mishra, 2012; of Technology, 2007; Wan & Birch, 2011; Qin, Huang, & Suganthan, 2009; Qing, 2009).

Table C1: Some information about the involved benchmark functions in this book

f No.	Function Name	Dimension (n)	Variables Bounds	Global Optimal
f1-01	Mineshaft F1	1	$0 \leqslant x \leqslant 10$	1.380487165157852
f1-02	Mineshaft F2	1	$-10 \leqslant x \leqslant 10$	-1.416353520337699
f1-03	Shekel's F1	1	$0 \leqslant x \leqslant 10$	-10.153198755084881776356839400251
f1-04	Shekel's F2	1	$0 \leqslant x \leqslant 10$	-10.402822044707775329070518200751
f1-05	Shekel's F3	1	$0 \leqslant x \leqslant 10$	-10.536290299294717105427357601002
f1-06	Strongin-Zilinskas-Shaltyanis	1	$3 \leqslant x \leqslant 7.5$	-1.601307546494396
f1-07	Suharev	1	$0 \leqslant x \leqslant 1$	-1
f1-08	Zilinskas F2	1	$0 \leqslant x \leqslant 1$	-1.125
f2-01	Aluffi-Pentini	2	$-10 \leqslant x_i \leqslant 10$	-0.352386073800034
f2-02	Banana Shape	2	$-1.5 \leqslant x_1 \leqslant 1.5$ $-2.5 \leqslant x_2 \leqslant 0.5$	-25
f2-03	Beale	2	$-4.5 \leqslant x_i \leqslant 4.5$	0
f2-04	Becker-Lago	2	$-10 \leqslant x_i \leqslant 10$	0
f2-05	Bird	2	$-2\pi \leqslant x_i \leqslant 2\pi$	-106.7645367198034
f2-06	Bohachevsky F1	2	$-50 \leqslant x_i \leqslant 50$	0
f2-07	Bohachevsky F2	2	$-50 \leqslant x_i \leqslant 50$	0
f2-08	Bohachevsky F3	2	$-50 \leqslant x_i \leqslant 50$	0
f2-09	Booth	2	$-10 \leqslant x_i \leqslant 10$	0
f2-10	Branin RCOS	2 2	$-5 \leqslant x_1 \leqslant 10$ $0 \leqslant x_2 \leqslant 15$	0.39788735772973816

f No.	Function Name	Dimension (n)	Variables Bounds	Global Optimal
$f2$-11	Bukin F4	2	-15 $\leqslant x_i \leqslant$ 5	0
$f2$-12	Bukin F6	2	-15 $\leqslant x_i \leqslant$ 5	0
$f2$-13	Carrom Table	2	-10 $\leqslant x_i \leqslant$ 10	-24.15681551650653
$f2$-14	Chichinadze	2	-30 $\leqslant x_i \leqslant$ 30	-42.94438701899098
$f2$-15	Complex	2	-2 $\leqslant x_i \leqslant$ 2	0
$f2$-16	Cosine Mixture	2	-1 $\leqslant x_i \leqslant$ 1	0.2
$f2$-17	Cross In Tray	2	-15 $\leqslant x_i \leqslant$ 15	-2.062611870822739
$f2$-18	Cross Leg Table	2	-10 $\leqslant x_i \leqslant$ 10	-1
$f2$-19	Crowned Cross	2	-10 $\leqslant x_i \leqslant$ 10	0.0001
$f2$-20	Davis	2	-100 $\leqslant x_i \leqslant$ 100	0
$f2$-21	Decanomial	2	-10 $\leqslant x_i \leqslant$ 10	0
$f2$-22	Dekkers-Aarts	2	-20 $\leqslant x_i \leqslant$ 20	-24777
$f2$-23	Drop Wave	2	-5.12 $\leqslant x_i \leqslant$ 5.12	-1
$f2$-24	Easom	2	-10 $\leqslant x_i \leqslant$ 10	-1
$f2$-25	Egg Holder	2	-512 $\leqslant x_i \leqslant$ 512	-959.640662711 (for n=2)
$f2$-26	EXP2	2	0 $\leqslant x_i \leqslant$ 20	0
$f2$-27	Freudenstein-Roth	2	-10 $\leqslant x_i \leqslant$ 10	0
$f2$-28	Giunta	2	-1 $\leqslant x_i \leqslant$ 1	0.06447042053690566 (for n=2)
$f2$-29	Goldstein-Price	2	-2 $\leqslant x_i \leqslant$ 2	3
$f2$-30	Himmelblau	2	-6 $\leqslant x_i \leqslant$ 6	0
$f2$-31	Holder Table	2	-10 $\leqslant x_i \leqslant$ 10	-19.20850256788675
$f2$-32	Hosaki	2	0 $\leqslant x_i \leqslant$ 10	-2.345811576101292
$f2$-33	Kearfott	2	-3 $\leqslant x_i \leqslant$ 4	0
$f2$-34	Inverted Cosine Wave	2	-5 $\leqslant x_i \leqslant$ 5	-n+1
$f2$-35	Levy F3 (Shubert or Hansen)	2	-10 $\leqslant x_i \leqslant$ 10	-176.5417931365915
$f2$-36	Levy F5	2	-10 $\leqslant x_i \leqslant$ 10	-176.1375
$f2$-37	Matyas	2	-10 $\leqslant x_i \leqslant$ 10	0
$f2$-38	McCormick	2	-1.5 $\leqslant x_1 \leqslant$ 4 -3 $\leqslant x_2 \leqslant$ 4	-1.913222954981037
$f2$-39	Michalewicz	2	0 $\leqslant x_i \leqslant \pi$	-1.801303228593281 (for n=2)
$f2$-40	Muller-Brown Surface	2	-1.5 $\leqslant x_1 \leqslant$ 1 -0.5 $\leqslant x_2 \leqslant$ 2.5	-146.6995172099539
$f2$-41	Parsopoulos	2	-5 $\leqslant x_i \leqslant$ 5	0
$f2$-42	Peaks	2	-4 $\leqslant x_i \leqslant$ 4	-6.551133332622496
$f2$-43	Pen Holder	2	-11 $\leqslant x_i \leqslant$ 11	-0.9635348327265058
$f2$-44	Powell's Badly Scaled	2	-10 $\leqslant x_i \leqslant$ 10	0
$f2$-45	Sawtoothxy	2	-20 $\leqslant x_i \leqslant$ 20	0
$f2$-46	Schaffer's F1	2	-100 $\leqslant x_i \leqslant$ 100	0
$f2$-47	Schaffer's F2	2	-100 $\leqslant x_i \leqslant$ 100	0
$f2$-48	Shekel's Foxholes	2	-65.536 $\leqslant x_i \leqslant$ 65.536	0.998003837794449325873406851315
$f2$-49	Sinusoidal Problem	2	0 $\leqslant x_i \leqslant 180^o$	-3.5
$f2$-50	Stenger	2	-1 $\leqslant x_i \leqslant$ 4	0
$f2$-51	Storn	2	-4 $\leqslant x_i \leqslant$ 4	-18.0587
$f2$-52	Stretched V	2	-10 $\leqslant x_i \leqslant$ 10	0 (for n=2)
$f2$-53	Test Tube Holder	2	-10 $\leqslant x_i \leqslant$ 10	-10.872299901558
$f2$-54	Treccani	2	-5 $\leqslant x_i \leqslant$ 5	0
$f2$-55	Trefethen F4	2 2	-6.5 $\leqslant x_1 \leqslant$ 6.5 -4.5 $\leqslant x_2 \leqslant$ 4.5	-3.3068686474
$f2$-56	Tripod	2	-100 $\leqslant x_i \leqslant$ 100	0
$f2$-57	Zakharov	2	-5 $\leqslant x_i \leqslant$ 10	0
$f2$-58	Zettl	2	-1 $\leqslant x_i \leqslant$ 5	-0.003791237220468656
$f2$-59	3-Hump Camel-Back	2	-5 $\leqslant x_i \leqslant$ 5	0
$f2$-60	6-Hump Camel-Back	2	-5 $\leqslant x_i \leqslant$ 5	-1.031628453489877
$f3$-01	Box-Betts	3	0.9 $\leqslant x_1, x_3 \leqslant$ 1.2 9 $\leqslant x_2 \leqslant$ 11.2	0
$f3$-02	Hartman's F1	3	0 $\leqslant x_i \leqslant$ 1	-3.86278214782076
$f3$-03	Helical Valley	3	-100 $\leqslant x_i \leqslant$ 100	0
$f3$-04	Levy F8	3	-10 $\leqslant x_i \leqslant$ 10	0
$f3$-05	Meyer and Roth	3	-20 $\leqslant x_i \leqslant$ 20	0.4×10^{-4}
$f3$-06	Perm F1	3	-$n \leqslant x_i \leqslant n$+1	0
$f4$-01	Corana (or Ingber)	4	-100 $\leqslant x_i \leqslant$ 100	0
$f4$-02	Kowalik	4	-5 $\leqslant x_i \leqslant$ 5	0.00030748598780560421684043449710 09
$f4$-03	Miele and Cantrell	4	-1 $\leqslant x_i \leqslant$ 1	0
$f4$-04	Powell's Quartic	4	-10 $\leqslant x_i \leqslant$ 10	0
$f4$-05	Neumaier F2	4	0 $\leqslant x_i \leqslant n$	0
$f4$-06	Wood (or Colville)	4	-10 $\leqslant x_i \leqslant$ 10	0
$f5$-01	AMGM	5	0 $\leqslant x_i \leqslant$ 10	0
$f5$-02	Osborne F1	5	0 $\leqslant x_1, x_2, x_4, x_5 \leqslant$ 3 -3 $\leqslant x_3 \leqslant$ 0	5.46×10^{-5}
$f5$-03	SODP	5	-1 $\leqslant x_i \leqslant$ 1	0

f No.	Function Name	Dimension (n)	Variables Bounds	Global Optimal
f5-04	Styblinski-Tang	5	-5 $\leqslant x_i \leqslant$ 5	-39.16616570377142n
f6-01	Hartman's F2	6	0 $\leqslant x_i \leqslant$ 1	-3.32236801141551
f6-02	Perm F2	6	-1 $\leqslant x_i \leqslant$ 1	0
f9-01	ANNs XOR	9	-1 $\leqslant x_i \leqslant$ 1	0.959759
f9-02	Price's Transistor	9	-10 $\leqslant x_i \leqslant$ 10	0
f9-03	Storn's Chebyshev	9	-$2^n \leqslant x_i \leqslant 2^n$	0
f10-01	Epistatic Michalewicz	10	0 $\leqslant x_i \leqslant \pi$	-9.660152
f10-02	Katsuura	10	-1000 $\leqslant x_i \leqslant$ 1000	1 (for n=10)
f10-03	Odd Square	10	-15 $\leqslant x_i \leqslant$ 15	-1.143833 (for n=10)
f10-04	Paviani	10	-2.001 $\leqslant x_i \leqslant$ 9.999	-45.77848
f15-01	Dixon-Price	15	-10 $\leqslant x_i \leqslant$ 10	0
f15-02	Neumaier F3 (or Trid)	15	-$n^2 \leqslant x_i \leqslant n^2$	-665 (for n=15)
f15-03	Normalized Rana's Function + Diagonal Wrap	15	-520 $\leqslant x_i \leqslant$ 520	-512.753162426239100568636786193
f17-01	Bent Cigar	17	-100 $\leqslant x_i \leqslant$ 100	0
f17-02	Deflected Corrugated Spring	17	0 $\leqslant x_i \leqslant$ 10	-1
f17-03	Infinity (or Csendes)	17	-1 $\leqslant x_i \leqslant$ 1	0
f20-01	Alpine	20	-10 $\leqslant x_i \leqslant$ 10	0
f20-02	Quintic	20	-10 $\leqslant x_i \leqslant$ 10	0
f20-03	Pathological	20	-100 $\leqslant x_i \leqslant$ 100	0
f30-01	Ackley	30	-32 $\leqslant x_i \leqslant$ 32	0
f30-02	Generalized Griewank	30	-600 $\leqslant x_i \leqslant$ 600	0
f30-03	Generalized Penalized F1	30	-50 $\leqslant x_i \leqslant$ 50	0
f30-04	Generalized Penalized F2	30	-50 $\leqslant x_i \leqslant$ 50	0
f30-05	Generalized Rastrigin	30	-5.12 $\leqslant x_i \leqslant$ 5.12	0
f30-06	Generalized Rosenbrock	30	-30 $\leqslant x_i \leqslant$ 30	0
f30-07	Generalized Schwefel F2.26	30	-500 $\leqslant x_i \leqslant$ 500	-418.98288727243379980791360139$8n$
f30-08	Mishra F1	30	0 $\leqslant x_i \leqslant$ 1	2
f30-09	Mishra F2	30	0 $\leqslant x_i \leqslant$ 1	2
f30-10	Quartic (or De Jong's F4)	30	-1.28 $\leqslant x_i \leqslant$ 1.28	0
f30-11	Schwefel F1.2	30	-100 $\leqslant x_i \leqslant$ 100	0
f30-12	Schwefel F2.21	30	-100 $\leqslant x_i \leqslant$ 100	0
f30-13	Schwefel F2.22	30	-10 $\leqslant x_i \leqslant$ 10	0
f30-14	Sphere (Square Sum, Harmonic or De Jong's F1)	30	-100 $\leqslant x_i \leqslant$ 100	0
f30-15	Step	30	-100 $\leqslant x_i \leqslant$ 100	0
f60-01	Hyper-Ellipsoid	60	-1 $\leqslant x_i \leqslant$ 1	0
f60-02	Qing	60	-500 $\leqslant x_i \leqslant$ 500	0
f60-03	Salomon	60	-100 $\leqslant x_i \leqslant$ 100	0

جامعة البحرين
المكتبة
LIBRARY
University of Bahrain

ملخص الدراسة

التنسيق الأمثل للمرحلات الكهربائية القائم عملها على التيار الاتجاهي الزائد هو قضية حساسة في هندسة القدرة الكهربائية. هذه المشكلة العملية تعتبر من مشاكل التحسين غير الخطية والمقيدة للغاية. زمن التشغيل لكل مرّحِل كهربائي يعتمد على متغيرين مستقلين يسميان بإعداد القابس وإعداد الوقت المضاعف. مجموع كل أوقات التشغيل للمرحلات الكهربائية القائم عملها على التيار الاتجاهي الزائد، عندما تعمل كمرحلات أولية، هو هدف دالة خوارزمية التحسين لمثل هذه المشكلة . كلما كبرت هذه الشبكات الكهربائية كلما زاد التعقيد، حيث أن عدد المرحلات سيزداد، ولهذا يكون إيجاد الحل الأمثل صعباً جداً. منذ عام 1988م، العديد من التقنيات العددية تم توظيفها. وبالرغم من ذلك، لا تزال هناك العديد من المناطق المفتوحة التي لم تتم تغطيتها بعد في مراجع هذا الموضوع.

هذه الأطروحة لها مساهمات متعددة؛ فهي أولاً تستخدم خوارزمية تطورية جديدة من النوع القائم على السكان وتسمى خوارزمية الحل الأمثل القائمة على الجغرافية البيولوجية، وثانياً تقوم بتقييم أداء عشرة أنواع من تقنيات مناولة القيد ، وثالثاً تقترح خوارزمية فرعية جديدة لتجنب حالة عدم الجدوى بسبب القيود الانتقائية ، ورابعاً تقدم تهجين جديد ما بين خوارزمية الجغرافية البيولوجية وخوارزمية البرمجة الخطية القياسية. تم تقييم أداء هذه الخوارزميات المفترضة بلستخدام عدة نظم اختباريه مختلفة، والنتائج المتحصلة تم مقارنتها مع أداء طرق تحسين أخرى تم عرضها مسبقاً في مراجع هذا الموضوع. وقد بينت النتائج أن خوارزمية المهجنة مع التحسينات المقترحة قد تفوقت على العديد من الخوارزميات المقترحة في المراجع سواء في الجدوى، الجودة، أو زمن المعالجة السريع جداً.

من ناحية أخرى، تقدم هذه الأطروحة دراسة موسعة عن أداء النماذج الأصلية لخوارزمية الجغرافية البيولوجية بواسطة عمل مقارنة بينهم من خلال اختبار أداءها بلستخدام 23 دالة معيارية ذات أبعاد وتعقيدات مختلفة. علاوة على ذلك، تحاول هذه الدراسة أيضاً حل جذور المشكلة الموجودة في النماذج الأصلية لخوارزمية الجغرافية البيولوجية من خلال معالجة ضعف آلية الاستكشاف لديها بدلاً من استخدام طرق معقدة لحل الآثار المترتبة عليها . تعديلين تم اقتراحهما؛ تعديل عملية ا لاختيار الاحتمالي وتحجيم مدى عملية إزالة الخصائص المتكررة . هذه التعديلات الأساسية على خوارزمية الجغرافية البيولوجية تم اختبارها على نطاق واسع وذلك باستخدام 120 دالة اختبارية مع أبعاد وتعقيدات مختلفة ، ومن ثم تم استخدامها كبنية أساس لابتكار أربعة نماذج متقدمة لخوارزمية الجغرافية البيولوجية والتي تعتمد في عملها على التحديث في فضاء بحثها، وهي تدعى على التوالي المحدث الفردي للحدود، المحدث المتعدد للحدود، فضاء البحث المتكرر ، وحدود المتغيرات المتكيفة. أخيراً، تم حل التنسيق الأمثل لشبكة متكونة من 3 قضبان توزيع باستخدام خوارزمية الجغرافية البيولوجية المعززة بفضاء البحث المتكرر.

أيضاً، تم عمل دراسة موسعة على الأنواع المختلفة للمنحنيات المميزة للعلاقة ما بين الوقت والتيار على شبكة متكونة من 15 قضيب توزيع.

جامعة البحرين
المكتبة
LIBRARY
University of Bahrain

جامعة البحرين

كلية الهندسة

قسم الهندسة الكهربائية والإلكترونية

الإعدادات المثلى للمرحلات الكهربائية باستخدام تقنية الحل الأمثل القائم على الجغرافية البيولوجية

أطروحة مقدمة كجزء من متطلبات الحصول على درجة الماجستير العلوم
في الهندسة الكهربائية

إعـــداد

علي رضا عبدعلي محمد الرومي

20012006

إشـــراف

الدكتور جواد حسن طلاق
أستاذ مشارك

الدكتور فاضل عباس البصري
أستاذ مساعد

جامعة البحرين

مملكة البحرين

ديسمبر، 2014