# DESIGN AND APPLICATIONS OF SPIDER MONKEY OPTIMIZATION

**Ph.D. THESIS**

***by***

**KAVITA GUPTA**

**DEPARTMENT OF MATHEMATICS**
**INDIAN INSTITUTE OF TECHNOLOGY ROORKEE**
**ROORKEE – 247667 (INDIA)**
**AUGUST, 2017**

# DESIGN AND APPLICATIONS OF SPIDER MONKEY OPTIMIZATION

**A THESIS**

***Submitted in partial fulfilment of the requirements for the award of the degree***

***of***

**DOCTOR OF PHILOSOPHY**

***in***

**MATHEMATICS**

***by***

**KAVITA GUPTA**

**DEPARTMENT OF MATHEMATICS**
**INDIAN INSTITUTE OF TECHNOLOGY ROORKEE**
**ROORKEE – 247667 (INDIA)**
**AUGUST, 2017**

**INDIAN INSTITUTE OF TECHNOLOGY ROORKEE**
**ROORKEE**

## CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the thesis entitled **"DESIGN AND APPLICATIONS OF SPIDER MONKEY OPTIMIZATION"** in partial fulfilment of the requirements for the award of the Degree of Doctor of Philosophy and submitted in the **Department of Mathematics** of the **Indian Institute of Technology Roorkee, Roorkee** is an authentic record of my own work carried out during a period from January, 2013 to August, 2017 under the supervision of **Prof. Kusum Deep**, Professor, Department of Mathematics, Indian Institute of Technology Roorkee, Roorkee.

The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other Institute.

**(KAVITA GUPTA)**

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

(**Kusum Deep**)
**Supervisor**

The Ph.D. Viva-Voce Examination of Ms. Kavita Gupta, Research Scholar, has been held on ..........................................

**Chairman, SRC** **Signature of External Examiner**

This is to certify that the student has made all the corrections in the thesis.

**Signature of Supervisor** **Head of the Department**
**Date:**

# ABSTRACT

The contribution of this thesis is the proposal of four new SMO algorithms for solving continuous unconstrained and constrained optimization problems with a view to apply them in solving benchmark problems as well as real world optimization problems.

The applicability of SMO over non-linear continuous constrained optimization problems is investigated. A new version of SMO called Constrained Spider Monkey Optimization (CSMO) algorithm has been designed by using Deb's technique for handling constraints. The performance of proposed CSMO has been investigated over the constrained benchmark problems of IEEE CEC sessions 2006 and 2010. In order to assess the competitiveness of CSMO in solving constrained benchmark problems, it has been compared with three state-of-the-art algorithms namely ABC, DE and PSO on various performance metrics. The results have in presented numerically and graphically. The results have also been validated statistically by using a statistical test.

In order to further improve the performance of basic SMO, a new Tournament selection based SMO (TS-SMO) has been designed for solving non-linear continuous unconstrained optimization problems. The performance of proposed TS-SMO has been tested over a benchmark set of 46 benchmark problems and results are compared with basic SMO. For comparing the results, various performance metrics have been taken into account to justify the favourable effect of proposed modification. The results have been compared numerically, graphically and statistically.

One more modification of basic SMO named as Quadratic approximation based SMO (QASMO) has been designed for solving non-linear continuous unconstrained optimization problems. The performance of proposed QASMO has been tested over a benchmark set of 46 benchmark problems and results are compared with original SMO. For comparing the results, various performance metrics have been taken into account to justify the favourable effect of proposed modification. The results have been compared numerically, graphically and statistically.

Also, a new quadratic approximation based CSMO (QACSMO) has been designed. The performance of proposed QACSMO has been investigated over the constrained benchmark problems of IEEE CEC sessions 2006 and 2010 and the results have been compared with

CSMO on various performance metrics. The results have in presented numerically and graphically. The results have also been validated statistically by using a statistical test.

The main objective behind the development of these algorithms is to apply them over real life optimization problems; hence the applicability of proposed algorithms has been investigated over two real life optimization problems of Lennard-Jones problem and Portfolio Optimization problem.

Finally, the thesis is concluded with the overall conclusions, limitations and scope of the proposed algorithms. Also, the future scope and new directions for research in this area have been suggested.

# LIST OF PUBLICATIONS

## RESEARCH PAPERS PUBLISHED IN INTERNATIONAL REFEREED JOURNALS

1. **Gupta, K.**, Deep, K., & Bansal, J. C. (2016). Spider monkey optimization algorithm for constrained optimization problems. *Soft Computing* (**IF. 2.472**), Springer, 1-30.

2. **Gupta, K**., Deep, K., & Bansal, J. C. (2017). Improving the local search ability of spider monkey optimization algorithm using quadratic approximation for unconstrained optimization. *Computational Intelligence*. (**IF. 0.964)**, *33*(2), 210-240.

## RESEARCH PAPERS IN INTERNATIONAL CONFERENCES

1. **Gupta, K.**, Deep, K., Tournament Selection Based Probability Scheme in Spider Monkey Optimization Algorithm: In Harmony Search Algorithm, Springer, Berlin, Heidelberg, vol. 382, (2016), pp. 239–250.

## RESEARCH PAPERS NOT INCLUDED IN THESIS

1. **Gupta, K.,** Deep, K., Investigation of Suitable Perturbation Rate Scheme for Spider Monkey Optimization Algorithm: In Proceedings of Fifth International Conference on Soft Computing for Problem Solving, Springer, Singapore, vol. 437, (2016), pp.839-850.

## ACKNOWLEDGEMENTS

It is a pleasure to convey my gratitude to few of the many sources of inspiration for me and to all the people who helped me in successful realization of this thesis.

First and foremost I thank the incessant source of divine blessings, the almighty GOD, who always motivates me to move forward with his omens and love.

I express my deep sense of gratitude to my supervisor Prof. Kusum Deep, Professor, Department of Mathematics, Indian Institute of Technology Roorkee. I feel privileged to express my sincere regards to my supervisor for her valuable guidance, support and constant encouragement throughout the course of my research work. Her truly scientific intuition and broad vision inspired and enriched my growth as a student and researcher. The critical comments rendered by her during the discussions are deeply acknowledged. I humbly acknowledge a lifetime's gratitude to her.

I express my gratitude to Prof. V.K. Katiyar, Head of the Department. Special thanks to my SRC members: Prof. S.P. Yadav, Prof. N. Sukavanam and Dr. Durga Toshniwal for spending their valuable time during the discussions over the seminars. I express heartfelt thanks to Dr. Jagdish Chand Bansal , Associate Professor, SAU, New Delhi for his valuable suggestions in my research work. I am indebted to the Department of Mathematics, IIT Roorkee, to all its faculty and staff for their academic support and encouragement.

I am thankful to my seniors Dr. Amreek Singh, Dr. Pinkey Chauhan, Dr. Anupam Yadav, Dr. Amarjeet Singh and Dr. Garima Singh. I am also thankful to my colleagues Dr. Vanita Garg, Assif Assad, Shail Dinkar and Shubham Gupta for providing healthy and progressive research environment. I am also thankful to the members of Institute Computer Centre for their cooperation and support.

I consider myself truly blessed as I have always been in a good company of friends. Their love, inspiration, support and cooperation is beyond the scope of any acknowledgement, yet I would like to express my heartfelt gratitude to my friends Meenu Goyal, Pushpa Chaudhary, Urvashi Arora, Priyanka Sangal,  Kiran Behera and Renu Chaudhary.

Now I would like to thank some remarkable people who had, are having and will continue to have a tremendous influence on my life. I feel a deep sense of gratitude for my parents, Mr.

Sushil Kumar Gupta and Mrs. Santosh Gupta, who raised me with their unconditional love, mental support that I needed in every step and every sphere of my life. My parents have been the biggest source of inspiration in my life and have inculcated in me the values that really matters in life. My sincere and heartfelt thanks to my younger brother Akash Gupta for his love and continuous support during tough times.  My family has been the source of endless love and support that always helped me remain optimistic during ups and downs in life. I would like to give my special thanks to my uncles, aunts and cousins for their love and blessings. I also want to express my gratitude to my grandfather Sh. Musaddi Lal and grandmother Smt. Shanti Devi for their blessings.

My heartfelt thanks to all those who have supported me for successful realization of this thesis. My sincere apologies, if I forgot someone but I am grateful for their support. Finally, I would like to thank all the readers of this work, since any piece of academia is useful if it is read and understood by others so that it can become bridge for further research. The financial assistance from MHRD, Govt. of India is also gratefully acknowledged.

With profound gratitude, love and devotion, I dedicate this thesis to my family.

(KAVITA GUPTA)

# TABLE OF CONTENTS

**ABSTRACT** i

**LIST OF PUBLICATIONS** iii

**ACKNOWLEDGEMENTS** v

**TABLE OF CONTENTS** vii

**LIST OF FIGURES** xiii

**LIST OF TABLES** xv

**CHAPTER 1: INTRODUCTION** 1

1.1 WHAT IS OPTIMIZATION 1

1.2 DEFINITION OF AN OPTIMIZATION PROBLEM 2

1.3 LOCAL AND GLOBAL OPTIMAL SOLUTION 3

1.4 METHODS FOR GLOBAL OPTIMIZATION 4

1.5 METAHEURISTIC FOR GLOBAL OPTIMIZATION PROBLEMS 5

1.6 APPLICATIONS 5

1.7 WHY FURTHER RESEARCH IS REQUIRED? 6

1.7.1 MAIN ISSUES WITH NATURE INSPIRED OPTIMIZATION ALGORITHMS 6

1.7.2 NO FREE LUNCH THEOREM 7

1.8 MOTIVATION AND OBJECTIVES OF THE THESIS 8

1.9 MAJOR CONTRIBUTIONS OF THE THESIS 9

1.10 ORGANIZATION OF THE THESIS 9

**CHAPTER 2: SPIDER MONKEY OPTIMIZATION: CONCEPT, DEVELOPMENT AND LITERATURE REVIEW** 13

2.1 SWARM INTELLIGENT OPTIMIZATION ALGORITHMS 13

2.2 SPIDER MONKEY OPTIMIZATION 15

2.2.1 SOCIAL AND FORAGING BEHAVIOUR OF SPIDER MONKEYS 15

2.2.2 BASIC DEFINITIONS 17

2.2.3 ALGORITHMIC STRUCTURE OF SMO 17

2.3 LITERATURE REVIEW ON SMO 26

2.3.1 MODIFICATIONS AND COMPARISONS 26

2.3.2 HYBRIDIZATION 29

2.3.3 APPLICATIONS 29

2.4 CONCLUSIONS 30

**CHAPTER 3: NOVEL SPIDER MONKEY OPTIMIZATION ALGORITHM FOR CONSTRAINED OPTIMIZATION PROBLEMS** 33

3.1 CONSTRAINT HANDLING TECHNIQUES 34

3.2 CONSTRAINED SPIDER MONKEY OPTIMIZATION (CSMO) 34

3.2.1 DEB'S TECHNIQUE FOR CONSTRAINT HANDLING 35

3.2.2 THE PROPOSED CONSTRAINED SPIDER MONKEY OPTIMIZATION (CSMO) ALGORITHM 36

3.3 EXPERIMENTAL SETUP 40

3.3.1 BENCHMARK PROBLEMS AND EVALUATION CRITERIA 40

3.3.2 STATE-OF-THE-ART ALGORITHMS USED FOR COMPARISON OF RESULTS 42

3.3.3 SETTING OF CONTROL PARAMETERS 43

3.4 DISCUSSION OF EXPERIMENTAL RESULTS 43

3.4.1 DISCUSSION OF RESULTS FOR CEC2006 BENCHMARK PROBLEMS 44

3.4.2 DISCUSSION OF RESULTS FOR CEC2010 BENCHMARK PROBLEMS FOR 10 DIMENSIONS 46

3.4.3 DISCUSSION OF RESULTS FOR CEC2010 BENCHMARK PROBLEMS FOR 30 DIMENSIONS 47

3.5 CONCLUSIONS 48

**CHAPTER 4: TOURNAMENT SELECTION BASED SPIDER MONKEY OPTIMIZATION ALGORITHM FOR UNCONSTRAINED OPTIMIZATION** 99

4.1 MOTIVATION 99

4.2 THE PROPOSED TOURNAMENT SELECTION BASED SPIDER MONKEY OPTIMIZATION 100

4.2.1 TOURNAMENT SELECTION BASED PROBABILITY SCHEME 100

4.2.2 IMPLEMENTATION STRATEGY OF TOURNAMENT SELECTION BASED PROBABILITY SCHEME IN SMO 100

4.3 EXPERIMENTAL SETUP 101

4.3.1 BENCHMARK PROBLEMS 101

4.3.2 SETTING OF CONTROL PARAMETERS AND TERMINATION CRITERIA 102

4.3.3 PERFORMANCE EVALUATION CRITERIA 103

4.4 EXPERIMENTAL RESULTS AND DISCUSSION 105

4.4.1 DISCUSSION OF RESULTS FOR SCALABLE PROBLEMS 106

4.4.2 DISCUSSION OF RESULTS FOR NON-SCALABLE PROBLEMS 107

4.5 CONCLUSIONS 108

**CHAPTER 5: QUADRATIC APPROXIMATION BASED SPIDER MONKEY OPTIMIZATION ALGORITHM FOR UNCONSTRAINED OPTIMIZATION** 123

5.1 MOTIVATION 123

5.2 THE PROPOSED QUADRATIC APPROXIMATION BASED SPIDER MONKEY OPTIMIZATION 124

5.2.1 WORKING OF QA OPERATOR 124

5.2.2 IMPLEMENTATION STRATEGY OF QA IN SMO 124

5.3 EXPERIMENTAL SETUP 125

5.4 DISCUSSION OF EXPERIMENTAL RESULTS 126

5.4.1 DISCUSSION OF RESULTS FOR SCALABLE PROBLEMS 127

5.4.2 DISCUSSION OF RESULTS FOR NON-SCALABLE PROBLEMS 130

5.5 COMPARISON AMONG SMO, TS-SMO AND QASMO 131

5.6 CONCLUSIONS 132

**CHAPTER 6: QUADRATIC APPROXIMATION BASED CONSTRAINED SPIDER MONKEY OPTIMIZATION ALGORITHM** 155

6.1 THE PROPOSED QUADRATIC APPROXIMATION BASED CONSTRAINED SPIDER MONKEY OPTIMIZATION 155

6.2 EXPERIMENTAL SETUP 158

6.3 DISCUSSION OF EXPERIMENTAL RESULTS 158

6.3.1 DISCUSSION OF RESULTS FOR CEC2006 BENCHMARK PROBLEMS 159

6.3.2 DISCUSSION OF RESULTS FOR CEC2010 BENCHMARK PROBLEMS FOR 10 DIMENSIONS AND 30 DIMENSIONS 160

6.4 CONCLUSIONS 160

**CHAPTER 7: APPLICATION OF SPIDER MONKEY OPTIMIZATION TO SOLVE LENNARD-JONES PROBLEM** 181

7.1 LENNARD-JONES (L-J) PROBLEM 182

7.2 EXPERIMENTAL SETUP 182

7.3 DISCUSSION OF EXPERIMENTAL RESULTS 183

7.4 CONCLUSIONS 184

**CHAPTER 8: APPLICATION OF CONSTRAINED SPIDER MONKEY OPTIMIZATION TO SOLVE PORTFOLIO OPTIMIZATION PROBLEM** 191

8.1 PORTFOLIO OPTIMIZATION PROBLEM 191

8.2 MEAN-VARIANCE MODEL 192

8.3 EXPERIMENTAL SETUP 193

8.3.1 PARAMETER SETTING AND TERMINATION CRITERIA 193

8.3.2 OPTIMIZATION MODEL AND INPUT DATA 194

8.4 DISCUSSION OF EXPERIMENTAL RESULTS 195

8.5 CONCLUSIONS 196

**CHAPTER 9: CONCLUSIONS AND FUTURE SCOPE** 205

9.1 CONCLUSIONS 205

9.2 FUTURE RESEARCH 207

9.2.1 ON CSMO 207

9.2.2 ON TS-SMO 207

9.2.3 ON QASMO 207

9.2.4 ON QACSMO 208

**BIBLIOGRAPHY** 209

**APPENDICES** 229

**Appendix I: LIST OF CONSTRAINED BENCHMARK PROBLEMS FROM IEEE CEC 2006** 231

**Appendix II: LIST OF CONSTRAINED BENCHMARK PROBLEMS FROM IEEE CEC 2010** 245

**Appendix III: LIST OF UNCONSTRAINED BENCHMARK PROBLEMS** 251

**Appendix IV: PERFORMANCE INDEX (PI)** 261

# LIST OF FIGURES

| Figure No. | Particulars | Page No. |
|---|---|---|
| 1.1 | Demonstration of Local Maxima and Global Maxima | 4 |
| 3.1 | Convergence graphs of problems g01-g08 | 49 |
| 3.2 | Convergence graphs of problems g09-g16 | 50 |
| 3.3 | Convergence graphs of problems g17-g24 | 51 |
| 3.4 | Convergence graphs of problems C01-C08 (10 dimensions) | 52 |
| 3.5 | Convergence graphs of problems C09-C16 (10 dimensions) | 53 |
| 3.6 | Convergence graphs of problems C017-C18 (10 dimensions) | 54 |
| 3.7 | Convergence graphs of problems C01-C08 (30 dimensions) | 55 |
| 3.8 | Convergence graphs of problems C09-C16 (30 dimensions) | 56 |
| 3.9 | Convergence plots of problems C17-C18 (30 dimensions) | 57 |
| 4.1 | Performance Index graph for Scalable problems for success rate versus average number of function evaluations for successful runs | 109 |
| 4.2 | Convergence graphs for Problem No.1-Problem No.8 | 110 |
| 4.3 | Convergence graphs for Problem No.9-Problem No.16 | 111 |
| 4.4 | Convergence graphs for Problem No.17-Problem No.24 | 112 |
| 4.5 | Convergence graphs for Problem No.25-Problem No.30 | 113 |
| 4.6 | Performance Index graph for Non-Scalable problems for success rate versus average number of function evaluations for successful runs | 114 |
| 4.7 | Convergence graphs for Problem No.31-Problem No.38 | 115 |
| 4.8 | Convergence graphs for Problem No.39-Problem No.46 | 116 |
| 5.1 | Performance Index graph for Scalable problems for success rate versus average number of function evaluations for successful runs | 132 |
| 5.2 | Convergence graphs for Problem No.1-Problem No.8 | 133 |
| 5.3 | Convergence graphs for Problem No.9-Problem No.16 | 134 |
| 5.4 | Convergence graphs for Problem No.17-Problem No.24 | 135 |
| 5.5 | Convergence graphs for Problem No.25-Problem No.30 | 136 |

5.6 Performance Index graph for Non-Scalable problems for success rate versus average number of function evaluations for successful runs 137
5.7 Convergence graphs for Problem No.31-Problem No.38 138
5.8 Convergence graphs for Problem No.39-Problem No.46 139

6.1 Convergence graphs of problems g01-g08 162
6.2 Convergence graphs of problems g09-g16 163
6.3 Convergence graphs of problems g17-g24 164

7.1 Performance Index graph for clusters of atoms 185
7.2 Convergence graphs for clusters of atoms 186

8.1 Efficient frontier 203

# LIST OF TABLES

| Table No. | Particulars | Page No. |
|---|---|---|
| 3.1 | Classification of problems in CEC2006 benchmark set on the basis of the type of constraints | 58 |
| 3.2 | Classification of problems in CEC2010 benchmark set on the basis of the type of constraints | 58 |
| 3.3 | Parameter setting for CSMO, ABC, CHDE, PESO | 59 |
| 3.4 | Feasibility Rate (F.R.) and Best, Median, Worst, Mean and Standard Deviation (Stdev) of the function error values obtained by CSMO with 25 independent runs on CEC2006 Benchmark Problems | 60 |
| 3.5 | Success Rate (S.R.) and Best, Median, Worst, Mean and Standard Deviation (Stdev) of the number of function evaluations obtained by CSMO with successful runs out of 25 independent runs on CEC2006 Benchmark Problems | 61 |
| 3.6 | Comparison of CSMO against ABC, CHDE and PESO in terms of Feasibility Rate (F.R.) and Success Rate (S.R.) on CEC2006 Benchmark Problems | 62 |
| 3.7 | Summary of pairwise comparison of CSMO against ABC, CHDE and PESO in terms of Feasibility Rate (F.R.) and Success Rate (S.R.) on CEC2006 benchmark problems | 63 |
| 3.8 | Comparison of CSMO against ABC, CHDE and PESO in terms of Best of function error values obtained with 25 independent runs on CEC2006 Benchmark Problems | 64 |
| 3.9 | Comparison of CSMO against ABC, CHDE and PESO in terms of Median of function error values obtained with 25 independent runs on CEC2006 Benchmark Problems | 65 |
| 3.10 | Comparison of CSMO against ABC, CHDE and PESO in terms of Worst of function error values obtained with 25 independent runs on CEC2006 Benchmark Problems | 66 |
| 3.11 | Comparison of CSMO against ABC, CHDE and PESO in terms of Mean of function error values obtained with feasible runs out of 25 independent runs on CEC2006 Benchmark Problems | 67 |

3.12 Comparison of CSMO against ABC, CHDE and PESO in terms of Standard Deviation of function error values obtained with feasible runs out of 25 independent runs on CEC2006 Benchmark Problems 68

3.13 Summary of pairwise comparison of CSMO against ABC, CHDE and PESO in terms of Best, Median, Worst, Mean and Standard Deviation (Stdev) of function error values on CEC2006 benchmark problems 69

3.14 Comparison of CSMO against ABC, CHDE and PESO in terms of Best of number of function evaluations obtained with successful runs out of 25 independent runs on CEC2006 Benchmark Problems 70

3.15 Comparison of CSMO against ABC, CHDE and PESO in terms Median of number of function evaluations obtained with successful runs out of 25 independent runs on CEC2006 Benchmark Problems 71

3.16 Comparison of CSMO against ABC, CHDE and PESO in terms Worst of number of function evaluations obtained with successful runs out of 25 independent runs on CEC2006 Benchmark Problems 72

3.17 Comparison of CSMO against ABC, CHDE and PESO in terms Mean of number of function evaluations obtained with successful runs out of 25 independent runs on CEC2006 Benchmark Problems 73

3.18 Comparison of CSMO against ABC, CHDE and PESO in terms Standard Deviation of number of function evaluations obtained with successful runs out of 25 independent runs on CEC2006 Benchmark Problems 74

3.19 Summary of pairwise comparison of CSMO against ABC, CHDE and PESO in terms of Best, Median, Worst, Mean and Standard Deviation (Stdev) of number of function evaluations with successful runs out of 25 independent runs on CEC2006 benchmark problems 75

3.20 Wilcoxon Rank sum test based on function error values with a significance level of $\alpha = 0.05$ for CEC2006 problems ('+' indicates CSMO is significantly better, '-' indicates CSMO is significantly worse and '=' indicates there is no significant difference) 76

3.21 Average execution time per run (in seconds) (CEC2006) 77

3.22 Feasibility Rate (F.R.) and Best, Median, Worst, Mean and Standard Deviation (Stdev) of the objective function values obtained by CSMO with 25 independent runs on CEC2010 Benchmark Problems for 10 dimensions 78

3.23 Comparison of CSMO against ABC, CHDE and PESO in terms Feasibility 79

Rate obtained on CEC2010 Benchmark Problems for 10 dimensions

3.24 Summary of pairwise comparison of CSMO against ABC, CHDE and PESO in terms of Feasibility Rate (F.R.) on CEC2010 benchmark problems for 10 dimensions 79

3.25 Comparison of CSMO against ABC, CHDE and PESO in terms of Best of objective function values obtained with 25 independent runs on CEC2010 Benchmark Problems for 10 dimensions 80

3.26 Comparison of CSMO against ABC, CHDE and PESO in terms of Median of objective function values obtained with 25 independent runs on CEC2010 Benchmark Problems for 10 dimensions 81

3.27 Comparison of CSMO against ABC, CHDE and PESO in terms of Worst of objective function values obtained with 25 independent runs on CEC2010 Benchmark Problems for 10 dimensions 82

3.28 Comparison of CSMO against ABC, CHDE and PESO in terms of Mean of objective function values obtained with feasible runs out of 25 independent runs on CEC2010 Benchmark Problems for 10 dimensions 83

3.29 Comparison of CSMO against ABC, CHDE and PESO in terms of Standard Deviation of objective function value obtained with feasible runs out of 25 independent runs on CEC2010 Benchmark Problems for 10 dimensions 84

3.30 Summary of pairwise comparison of CSMO against ABC, CHDE and PESO in terms of Best, Median, Worst, Mean and Standard Deviation (Stdev) of objective function values on CEC2010 benchmark problems for 10 dimensions 85

3.31 Wilcoxon Rank sum test based on objective function value with a significance level of $\alpha = 0.05$ for CEC2010 problems for 10 dimensions ('+' indicates CSMO is significantly better, '-' indicates CSMO is significantly worse and '=' indicates there is no significant difference) 86

3.32 Average execution time per run (in seconds) (CEC2010-10 dimensions) 87

3.33 Feasibility Rate (F.R.) and Best, Median, Worst, Mean and Standard Deviation (Stdev) of the objective function values obtained by CSMO with 25 independent runs on CEC2010 Benchmark Problems for 30 dimensions 88

3.34 Comparison of CSMO against ABC, CHDE and PESO in terms of Feasibility Rate obtained on CEC2010 Benchmark Problems for 30 dimensions 89

3.35 Summary of pairwise comparison of CSMO against ABC, CHDE and PESO in terms of Feasibility Rate (F.R.) on CEC2010 benchmark problems for 30 dimensions 89

3.36 Comparison of CSMO against ABC, CHDE and PESO in terms of Best of objective function values obtained with 25 independent runs on CEC2010 Benchmark Problems for 30 dimensions 90

3.37 Comparison of CSMO against ABC, CHDE and PESO in terms of Median of objective function values obtained with 25 independent runs on CEC2010 Benchmark Problems for 30 dimensions 91

3.38 Comparison of CSMO against ABC, CHDE and PESO in terms of Worst of objective function value obtained with 25 independent runs on CEC2010 Benchmark Problems for 30 dimensions 92

3.39 Comparison of CSMO against ABC, CHDE and PESO in terms of Mean of objective function values obtained with feasible runs out of 25 independent runs on CEC2010 Benchmark Problems for 30 dimensions 93

3.40 Comparison of CSMO against ABC, CHDE and PESO in terms of Standard Deviation of objective function values obtained with feasible runs out of 25 independent runs on CEC2010 Benchmark Problems for 30 dimensions 94

3.41 Summary of pairwise comparison of CSMO against ABC, CHDE and PESO in terms of Best, Median, Worst, Mean and Standard Deviation (Stdev) of objective function values on CEC2010 benchmark problems for 30 dimensions 95

3.42 Wilcoxon Rank sum test based on objective function value with a significance level of $\alpha = 0.05$ for CEC2010 problems for 30 dimensions ('+' indicates CSMO is significantly better, '-' indicates CSMO is significantly worse and '=' indicates there is no significant difference) 96

3.43 Average execution time per run (in seconds) (CEC2010-30dimensions)` 97

4.1 Success Rate and Average number of function evaluations for successful runs (Scalable problems-30 dimensions) 117

4.2 T-test results for problems having identical success rate for SMO and TS-SMO (Scalable problems-30 dimensions) 118

4.3 Best, Average and Worst of error values obtained in 100 runs (Scalable problems-30 dimensions) 119

4.4 Average execution time per run (in seconds) (Scalable problems-30 dimensions) 120
4.5 Success Rate and Average number of function evaluations for successful runs (Non-Scalable problems) 121
4.6 T-test results for problems having identical success rate for SMO and TS-SMO (Non-Scalable problems) 121
4.7 Best, Average and Worst of error values obtained in 100 runs (Non-Scalable problems) 122
4.8 Average execution time per run (in seconds) (Non-Scalable problems) 122

5.1 Success Rate and Average number of function evaluations for successful runs for SMO and QASMO (Scalable problems-30 dimensions) 140
5.2 T-test results for problems having identical number of successful runs for SMO and QASMO (Scalable problems-30 dimensions) 141
5.3 Best, average and worst of error values obtained in 100 independent runs (scalable problems-30dimensions) 142
5.4 Average execution time per run (in seconds) (Scalable problems-30 dimensions) 143
5.5 Success Rate and Average number of function evaluations for successful runs for SMO and QASMO (Non-Scalable problems) 144
5.6 T-test results for problems having identical number of successful runs for SMO and QASMO (Non-Scalable problems) 144
5.7 Best, average and worst of error values obtained in 100 independent runs (Non-Scalable problems) 145
5.8 Average execution time per run (in seconds) (Non-Scalable problems) 145
5.9 Success Rate and Average number of function evaluations for successful runs of SMO, TS-SMO and QASMO (Scalable problems-30 dimensions) 146
5.10 Best of error values obtained in 100 independent runs of SMO, TS-SMO and QASMO (Scalable problems-30 dimensions) 147

5.11 Average of error values obtained in 100 independent runs of SMO, TS-SMO and QASMO (Scalable problems-30 dimensions) 148
5.12 Worst of error values obtained in 100 independent runs of SMO, TS-SMO and QASMO (Scalable problems-30 dimensions) 149
5.13 Success Rate and Average number of function evaluations for successful 150

runs of SMO, TS-SMO and QASMO (Non-Scalable problems)

5.14 Best of error values obtained in 100 independent runs of SMO, TS-SMO and QASMO (Non-Scalable problems) 151

5.15 Average of error values obtained in 100 independent runs of SMO, TS-SMO and QASMO (Non-Scalable problems) 152

5.16 Worst of error values obtained in 100 independent runs of SMO, TS-SMO and QASMO (Non-Scalable problems) 153

5.17 Summary of results for Scalable and Non-Scalable problems of SMO, TS-SMO and QASMO 154

6.1 Comparison of Feasibility Rate and Success Rate for CEC2006 Benchmark Problems 165

6.2 Comparison Of Best, Median And Worst Of Function Error Values Obtained in 25 Independent Runs For CEC2006 Benchmark Problems 166

6.3 Comparison of Mean and Standard Deviation of function error values obtained with 25 independent runs for CEC2006 benchmark problems 167

6.4 Comparison of Best, Median and Worst of the number of function evaluations obtained with successful runs error values for CEC2006 benchmark problems 168

6.5 Comparison of Mean and Standard Deviation of the number of function evaluations obtained with successful runs error values for CEC2006 benchmark problems 169

6.6 Summary of comparison of QACSMO against CSMO for CEC2006 benchmark problems 170

6.7 Wilcoxon Rank sum test based on function error values with a significance level of $\alpha = 0.05$ for CEC2006 Benchmark Problems (‘+’ indicates QACSMO is significantly better, ‘-’ indicates QACSMO is significantly worse and ‘=’ indicates there is no significant difference) 171

6.8 Average time per run (in seconds) for CEC2006 Benchmark Problems 171

6.9 Feasibility Rate for CEC2010 Benchmark Problems (10 dimensions) 172

6.10 Best, Median and Worst of objective function values obtained in 25 independent runs for CEC2010 Benchmark Problems (10 dimensions) 173

6.11 Mean and Standard Deviation of function error values obtained with feasible runs out of 25 independent runs for CEC2010 (10 dimensions) 174

6.12 Summary of comparison of QACSMO against CSMO for CEC2010 Benchmark Problems (10 dimensions) 175
6.13 Wilcoxon Rank sum test based on objective function value with a significance level of $\alpha = 0.05$ for CEC2010 Benchmark Problems (10 dimensions) ('+' indicates QACSMO is significantly better, '-' indicates CSMO is significantly worse and '=' indicates there is no significant difference) 175
6.14 Average time per run (in seconds) for CEC2010 problems (10 dimensions) 176
6.15 Feasibility Rate for CEC2010 Benchmark Problems (30 dimensions) 176
6.16 Best, Median and Worst of function error values obtained in 25 independent runs for CEC2010 Benchmark Problems (30 dimensions) 177
6.17 Mean and Standard Deviation of function error values obtained with feasible runs out of 25 independent runs for CEC2010 (30 dimensions) 178
6.18 Summary of comparison of QACSMO against CSMO for CEC2010 Benchmark Problems (10 dimensions) 179
6.19 Wilcoxon Rank sum test based on objective function value with a significance level of $\alpha = 0.05$ for CEC2010 Benchmark Problems (30 dimensions) ('+' indicates QACSMO is significantly better, '-' indicates CSMO is significantly worse and '=' indicates there is no significant difference) 179
6.20 Average time per run (in seconds) for CEC2010 Benchmark Problems (30 dimensions) 180

7.1 Dimension and search space for different number of atoms 183
7.2 Success Rate and Average number of function evaluations for successful runs for SMO, TS-SMO and QASMO (Lennard-Jones problem) 187
7.3 Best of error values obtained in 100 independent runs (Lennard-Jones problem) 187
7.4 Average of error values obtained in 100 independent runs (Lennard-Jones problem) 188
7.5 Worst of error values obtained in 100 independent runs (Lennard-Jones problem) 188
7.6 T-test results for problems having identical success rate for SMO, TS-SMO and QASMO 189

| | | |
|---|---|---|
| 7.7 | Average execution time per run (in seconds) | 189 |
| | | |
| 8.1 | List of retail companies (assets) | 197 |
| 8.2 | Average Monthly Return of assets for the period April 01, 2015 to March 31, 2016 | 198 |
| 8.3 | Expected return of assets | 199 |
| 8.4 | Variance and Covariance Matrix | 200 |
| 8.5 | Result of portfolio selection using variance | 201 |
| 8.6 | Portfolio set for different values of $r_0$ | 202 |
| 8.7 | Average execution time taken by CSMO per run (in seconds) | 203 |

# CHAPTER 1

# INTRODUCTION

Most of the real world problems in science, engineering, economics, finance etc. can be modelled as optimization problems. However, the mathematical models of these optimization problems are characterized by the properties like non-linearity, high dimensionality, multimodality etc. which make them difficult to be solved by traditional optimization methods. Consequently, non-traditional optimization methods/algorithms come into play to deal with such optimization problems which can solve them efficiently at a low computational cost. The present thesis is an attempt to understand the concept and to design different versions of a recently proposed non-traditional optimization algorithm called Spider Monkey Optimization (SMO) for solving different types of optimization problems. The aim is to evaluate the performance of SMO in solving different types of benchmarks as well as real world optimization problems. The scope of the thesis is limited to single objective unconstrained and constrained continuous global optimization problems only.

The present chapter is introductory in nature. It provides some basic definitions of the terms and concepts that would be used in the thesis. It also highlights the motivation and objectives to carry out this work. Further, it enumerates the major contributions of the thesis. At the end, chapter wise summary of the thesis is provided.

## 1.1 WHAT IS OPTIMIZATION

In the most general terms, optimization refers to the art of performing a task in the most efficient manner with the use of available limited resources. It may have a variety of implications depending upon the context. For example, for a businessman, optimization may refer to maximizing his profit while minimizing the cost of production. For a student, optimization may refer to minimizing the study hours while maximizing the productive outcome.

Mathematical optimization refers to the approach of determining “the maximum”/ “the minimum” possible value that a given mathematical expression can attain in its specified domain. “The maximum”/ “the minimum” value is known as the optimal value of the function to be optimized. The mathematical expression that has to be optimized could be either linear, nonlinear, integer, geometric or may be fractional. In some situations, it may happen that the explicit mathematical formulation of the function is not readily defined or may not be available. Many times the mathematical function which needs to be optimized has restrictions in the form of inequality or inequality constraints. Therefore, the process of optimization can be considered as a problem of finding those values of the decision variables which satisfy all the inequality and equality constraints in such a way to provide an optimal value of the mathematical function being optimized. The mathematical techniques for determining the optimal value or value(s) ("the greatest possible value" or "the least possible value") of a mathematical function are called 'Optimization Techniques'. Determining the solution of the most realistic optimization problems may not be possible in the absence of robust optimization techniques. In literature, numerous books are available based on mathematical concepts of optimization. To name a few: Himmelblau [76], Chandra and Jayadeva [26], Jha and Hoda [81] etc.

## 1.2 DEFINITION OF AN OPTIMIZATION PROBLEM

The most general mathematical model of single objective optimization problem is as follows:

P (1.1) Minimize (or maximize) $f(\boldsymbol{x})$; $\boldsymbol{x} = (x_1, x_2, \dots, x_D)$

Subject to $\boldsymbol{x} \in F$, usually defined by

$$F = \{\boldsymbol{x} \in R^D / h_i(\boldsymbol{x}) = 0; and\ g_j(\boldsymbol{x}) \geq 0\} \tag{1.1}$$

$$i = 1,2, \dots, m \text{ and } j = m+1, m+2, \dots p$$

Where, $f, h_1, h_2, \dots, h_m, g_{m+1}, g_{m+2}, \dots, g_p$ are real valued functions defined on $R^D$.

$F$ is the feasible domain. Function $f$ that is to be optimized (maximized or minimized) is called the 'objective function'. Equations $h_i(\boldsymbol{x}) = 0$ are known as the equality constraints and inequalities $g_j(\boldsymbol{x}) \geq 0$ are called the inequality constraints (Inequality constraint of the type $g_j(\boldsymbol{x}) \geq 0$can be written as $-g_j(\boldsymbol{x}) \leq 0$). It is desired to determine those values of the independent variables $x_1, x_2, \dots, x_D$, which optimize the objective function $f(\boldsymbol{x})$ without violating any of the restrictions imposed in problem P (1.1). The variables $x_i's$ are known as 'decision variables'. A decision vector $\boldsymbol{x} = (x_1, x_2, \dots, x_D)$ which satisfies all the constraints is called a 'feasible solution'. A feasible solution $\boldsymbol{x}$ which optimizes the objective function $f$

is called a feasible optimal solution.

When all the functions $f, h_i, g_j$ appearing in the optimization problem are linear, the problem is called a ‘Linear Programming Problem’ (LPP). However if one or more of these functions are nonlinear, then the problem is called a ‘Nonlinear Optimization Problem’ or a ‘Nonlinear Programming Problem’ (NLPP).

If the decision variables of the optimization problem are continuous, then it is said be a ‘Continuous Optimization Problem’ otherwise it is called a ‘Combinatorial Optimization Problem’. If there is an additional condition that the decision variables should be integers, then it is called an ‘Integer Programming Problem’ (IPP). If some of the variables are restricted to be integers while others are real, then it is called a ‘Mixed Integer Programming Problem’ (MIPP).

If the constraints defining the range of the decision variables are the only constraints present in the optimization problem, then the problem is called ‘Unconstrained Optimization Problem’ otherwise it is called ‘Constrained Optimization Problem’.

## 1.3 LOCAL AND GLOBAL OPTIMAL SOLUTION

Let $F$ denote be the feasible region of the solution vector $\boldsymbol{x}$ that satisfies all the constraints of an optimization problem. Then, in case of a minimization problem, if for $\bar{\boldsymbol{x}} \in F$, there exists an ε-neighbourhood $N_\varepsilon(\bar{\boldsymbol{x}})$ around $\bar{\boldsymbol{x}}$ such that $f(\boldsymbol{x}) \geq f(\bar{\boldsymbol{x}})$ for each $\boldsymbol{x} \in F \cap N\varepsilon(\bar{\boldsymbol{x}})$ then $\bar{\boldsymbol{x}}$ is known as a ‘local optimal solution’ However, if, $\bar{\boldsymbol{x}} \in F$ and $f(\boldsymbol{x}) \geq f(\bar{\boldsymbol{x}})$ for all $\boldsymbol{x} \in F$ then $\bar{\boldsymbol{x}}$ is known as a ‘global optimal solution’ of the optimization problem at hand. Similar conditions hold in the case of maximization problem with inequalities reversed. Figure 1.1 shows local and global optimum solutions of a mathematical function.

In general, it may happen that there are either no optimal solutions, or a unique optimal solution or several optimal solutions, for a given optimization problem. In case, a problem has a single local optimal solution then it is also the global optimal solution. If, however, the optimization problem has several local optimal solutions, then, in general, one or more of them could be the global optimal solutions. In a Linear Programming Problem, it is for sure that, every local optimal solution is the global optimal solution. On the contrary, in case of a nonlinear optimization problem, it local optimal solution may not be global optimal solution always. But if the objective function is convex (for minimization case) and its feasible domain is also convex, then it is guaranteed that the local optimal solution is also the global optimal solution.

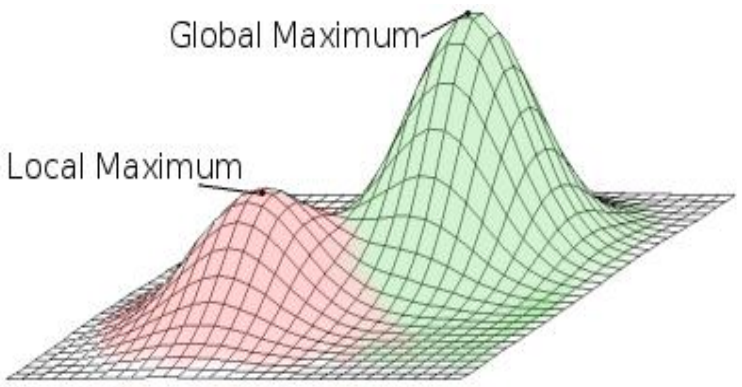

**Figure 1.1**: Demonstration of Local Maxima and Global Maxima

In many nonlinear optimization problems, it is usually desirous to determine a global optimal solution instead of a local optimal solution. But, in general, it is often difficult to obtain the global optimal solution of a nonlinear optimization problem, rather than finding the local optimal solution. However, due to its practical significance, it becomes necessary to determine the global optimal solution.

## 1.4 METHODS FOR GLOBAL OPTIMIZATION

Global optimization [43] focuses on determining the best (minimum) of the local optimal solutions. Some real world global optimization problems can be found in [24; 69; 157; 158; 163; 175] Designing global optimization techniques is not an easy task since, in general, there is no criterion for deciding whether a global optimal solution has been achieved or not. In view of the practical necessity and with the availability of fast and readily computing machines, many computational techniques are now being reported in literature for solving nonlinear optimization problems. The methods currently available in literature for solving nonlinear global optimization problems may be broadly classified as deterministic methods and probabilistic methods.

The deterministic methods try to guarantee that a neighbourhood of the global optima is attained. Such methods do not use any stochastic techniques, but rely on a thorough search of the feasible domain. However, they are applicable only to a restricted class of functions. On the other hand, probabilistic methods are used to find the near optimal solution. This is achieved by assuming that the good solutions are near to each other in the search space. This assumption is valid for most real life problem. The probabilistic methods make use of probabilistic or stochastic approach to search for the global optimal solutions. Although probabilistic methods do not give an absolute guarantee, these methods are sometimes preferred over the deterministic methods because they are applicable to a wider class of functions.

## 1.5 METAHEURISTIC FOR GLOBAL OPTIMIZATION PROBLEMS

Unlike traditional optimization methods, metaheuristics [45; 67; 85; 110; 183] are population based optimization methods. Metaheuristics have been applied to solve global optimization problems [8; 132]. In the recent decades, it is observed that biological systems are the inspiration of many of metaheuristic[1] search algorithms including Simulated Annealing [94], Ant Colony Optimization [46; 113], Genetic Algorithms [29; 39; 77], Artificial Bee Colony [16; 18; 59; 60; 143; 156; 189], Harmony Search [62; 166], Biogeography Based Optimization [79; 160; 161], Tabu Search [30; 65; 159], CMA-ES [71], Bacterial Foraging Optimization Algorithm (BFOA) [139; 117], Artificial Immune System [53], Central Force Optimization [54; 55; 56; 68], Glow Worm Swarm Optimization [96; 97; 98], Particle Swarm Optimization [5; 7; 28; 49; 90; 99; 105; 111; 144; 147; 170; 171; 173; 181; 185; 187], Differential Evolution [6; 10; 48; 80; 84; 114; 133; 142; 186], Gravitational Search Algorithm [91; 146], Evolutionary Membrane Algorithm [70], Vortex Search Algorithm [44], Kill Herd [57], Cuckoo search algorithm [66], Kidney Inspired Algorithm [78], Optics Inspired Optimization [88], Grey wolf optimizer [124], Invasive Weed Optimization [129; 130], Teaching Learning Based Optimization Algorithm [145], Water Cycle Algorithm [149], Water wave optimization [188], Lion optimization algorithm [184].

## 1.6 APPLICATIONS

The metaheuristics have many real world applications like Vehicle Routing Problem [150; 176], Predicting Colorectal Cancer [151], Pipe Network Design [63; 134], Water Network Design [61], Aeronautical and Aerospace Engineering [12], Biometric Authentication [19], Image Hiding Scheme [20], Breast Cancer Diagnosis [21; 22], Synthesis of AI/SiC nanocomposite [52], Wireless Sensor Networks [24], Distribution System [95], Data

[1] A heuristic approach to a problem is an empirical search or optimization method that often works at solving the problem, but doesn't have any of the rigorous proof that people like physicists and mathematicians expect. Nobody knows if it will *always* give the *best* answer to the problem. Meta-heuristic is a top level general strategy which guides other heuristics to search for feasible solutions in domains where the task is hard.

Clustering [174], Promotional Effort Allocation[114], Image Segmentation [9; 103], Construction of Classifiers [141], RIFD based Positioning System [104], Parameter Extraction of MESFET [148], Synthesis of Antenna Arrays [42; 64; 162 ], Economic Dispatch [48; 93], Optimal Reactive Power Dispatch [92; 182].

## 1.7 WHY FURTHER RESEARCH IS REQUIRED?

From the above discussion, it can be seen that a large number of metaheuristics have been designed and developed in the past by taking inspiration from almost every natural phenomenon. Still the research on designing and developing new metaheuristics is going on. Following are the reasons for ever expanding research in this direction.

### 1.7.1 MAIN ISSUES WITH METAHEURISTICS

Following are some of the issues which demand continuous attention from the research community to work on the development of these algorithms.

- Premature convergence

  Premature convergence refers to convergence of the population to a local optimal solution. In such a situation, though new solutions are generated in the population, but the fitness value of the global best solution does not get improved. This phenomenon is called stagnation of the swarm which causes the problem of premature convergence.

- Loss of diversity of the swarm

  Diversity of the swarm is related to the spread of the swarm in the search space. It measures how well the solutions are distributed over the search space. So, in order to find global optimal solution of complex landscapes, the diversity of the population must be maintained for most part during the run of the algorithm. Most of the metaheuristics lose diversity even during the initial runs making almost every solution in the population similar. It is one of the issues to be tackled efficiently.

- Setting of control parameters

  Every metaheuristic algorithm has some parameters associated with it. The performance of these algorithms heavily depends on the setting of their control parameters. Bad setting of the parameters can deteriorate the performance of the algorithm. The optimal setting of these control parameters is itself an optimization problem which needs extra computational cost.

- Long solution time

  Metaheuristics are applied to find satisfactory solutions of many real world optimization problems in science, engineering, finance, agriculture, economics etc. But these algorithms take much time in solving the problem due to their iterative and population based nature. Also, the search space of the problem increases with the increase in the number of decision variables, which require higher computational time to solve the problem.

So, there is the need to develop such optimization algorithms which can handle most of the above mentioned problems efficiently with minimal efforts.

### 1.7.2 NO FREE LUNCH THEOREM

A major and interesting result in optimization theory was the presentation of the "No Free Lunch (NFL) theorem". There are two versions of “No Free Lunch (NFL) theorem” [179; 180]. The version present by [180] is applied to the field of optimization. This theorem states "that the performance of all optimization (search) algorithms, amortized over the set of all possible functions, is equivalent." It means that improvement in the performance of an optimization algorithm over a class of optimization problems is exactly paid-off by its performance on the rest of the optimization problems. For example, we have two algorithms say algorithm A and algorithm B. If algorithm A performs better than algorithm B for some set of optimization problems, then the reverse will be true for rest of the optimization problems. The theorem has far reaching implications. One is that it states that no algorithm can be designed which is superior to all the other algorithms for all the optimization problems. Other implication is that given an optimization problem, one can’t say which algorithm is the best to solve this problem. So, it is an issue of concern for the practitioners to decide which algorithm should be used from a huge class of optimization algorithms to solve the problem at hand. Although, the theorem is defined over finite search spaces only, however, it is not proved if the result is applicable to infinite search spaces, e.g. $\Re^D$. All computer implementations of search algorithms will, in general, operate on finite search spaces, therefore the theorem is applicable to all existing algorithms.

## 1.8 MOTIVATION AND OBJECTIVES OF THE THESIS

The scope of this Thesis is Spider monkey optimization (SMO), a recently developed metaheuristic for global optimization problems. It is important to understand the usefulness of working on this new metaheuristic when there are already well established metaheuristics for solving different types of optimization problems. The reasons can be the increasing complexity of modern world optimization problems which demands for the development of efficient search techniques for solving them at low computational cost. Also, No Free Lunch theorem makes room for the development of new algorithms by stating that there is no best algorithm for all the optimization problems. So, a new algorithm showing competitive performance in comparison to other state-of-the-art algorithms on most of the optimization problems in the benchmark set deserves to be explored and developed. Moreover, the main question is why to use SMO for optimization purpose? The advantage of SMO over other well established metaheuristics like PSO, ABC and DE is that it has the provision to handle problems like stagnation or premature convergence in its original design. Such mechanism is not present in the original designs of other metaheuristics like PSO, ABC and DE. Though these algorithms have been improved enough to handle these problems very well. In simple words, we can say, advanced versions of PSO, ABC and DE are available which can handle the problem of stagnation or premature convergence very efficiently. But not every user who needs to solve an optimization problem has enough knowledge to deal with these algorithms. They do not even know about the occurrence of such problems (stagnation or premature convergence) during the execution of these algorithms. Even if they do know about these problems, they may not know which version to select in order to avoid the problem of stagnation or premature convergence. So, SMO is beneficial for such users who are not having much knowledge about metaheuristics and want to use basic version of a metaheuristic to solve an optimization problem using black box approach.

Also, all the modified versions of SMO proposed in literature are meant to solve unconstrained continuous optimizations problems only. No attempt has been made to solve the constrained optimization problems using SMO. Moreover, the performance of SMO has not been evaluated on standard benchmark problems like IEEE CEC benchmark problems.

Based on these observations, the following objectives for the thesis have been defined:

- To design different versions of Spider Monkey Optimization Algorithm for solving unconstrained and constrained continuous optimization problems.

- To test the performance of proposed algorithms for solving benchmarks as well as real world optimization problems.

## 1.9 MAJOR CONTRIBUTIONS OF THE THESIS

The major contributions of the thesis can be summarized as:

- Two new versions of SMO namely Tournament Selection based Spider Monkey Optimization (TS-SMO) and Quadratic Approximation based Spider Monkey Optimization (QASMO) have been designed for solving unconstrained continuous optimization problems. The performance of these proposed versions has been tested over a benchmark set of 46 unconstrained optimization problems and results are compared with basic SMO.

- Two new versions of SMO namely Constrained Spider Monkey Optimization (CSMO) and Quadratic Approximation based Constrained Spider Monkey Optimization (QACSMO) have been designed for solving constrained continuous optimization problems. The performance of proposed versions has been investigated over the constrained benchmark problems of IEEE CEC sessions 2006 and 2010 and results have been compared with some state-of-the-art algorithms.

- The applicability of proposed algorithms has been investigated over two real world optimization problems. One is Lennard-Jones problem which is an unconstrained continuous optimization problem from computational chemistry and the other is portfolio optimization problem which is a constrained continuous optimization problem from finance.

## 1.10 ORGANIZATION OF THE THESIS

In order to give a general overview of the contents of this thesis, brief description of each chapter is given below:

**CHAPTER 1. INTRODUCTION:** The current chapter is introductory in nature. It gives a brief introduction to optimization problems, optimization methods, metaheuristic algorithms followed by motivation, objectives and contributions of the thesis.

**CHAPTER 2. SPIDER MONKEY OPTIMIZATION: CONCEPT, DEVELOPMENT AND LITERATURE REVIEW:** This chapter provides the motivation behind the development of Spider Monkey Optimization (SMO) algorithm, its working and a brief review of the literature available on it.

**CHAPTER 3. NOVEL SPIDER MONKEY OPTIMIZATION ALGORITHM FOR CONSTRAINED OPTIMIZATION:** This is the actual start of our thesis. In this chapter, a modified version of SMO has been proposed for solving constrained continuous optimization problems. The proposed algorithm is named as Constrained Spider Monkey Optimization (CSMO). Deb's technique based on tournament selection has been used for handling constraints. The performance of CSMO is evaluated on constrained benchmark problems of varying difficulty level defined in IEEE CEC 2006 and CEC 2010. The results have been compared with constrained versions of Artificial Bee Colony (ABC), Differential Evolution (DE) and Particle Swarm Optimization (PSO). The conclusion made after comparing the results on numerical (tables), graphical (convergence graphs) and statistical (Wilcoxon's rank sum test) grounds proves the supremacy of CSMO in optimizing the constrained benchmark functions over other algorithms in comparison.

**CHAPTER 4. TOURNAMENT SELECTION BASED SPIDER MONKEY OPTIMIZATION ALGORITHM FOR UNCONSTRAINED OPTIMIZATION:** In this chapter, a modified version of basic SMO called as Tournament Selection based Spider Monkey Optimization (TS-SMO) has been proposed for solving unconstrained continuous optimization problems. In order to improve the exploration ability of basic SMO, tournament selection based probability scheme has been used in TS-SMO in place of fitness based probability scheme in basic SMO. The performance of TS-SMO is evaluated on a benchmark set of 46 unconstrained continuous optimization problems broadly classified as scalable and non-scalable problems and the results have been compared with basic SMO. The comparison of results on numerical (tables), graphical (convergence graphs and performance index) and statistical (t-test) grounds states that TS-SMO performs better than SMO over most of the scalable optimization problems, while its performance is moderate on non-scalable problems.

**CHAPTER 5. QUADRATIC APPROXIMATION BASED SPIDER MONKEY OPTIMIZATION ALGORITHM FOR UNCONSTRAINED OPTIMIZATION:** In this chapter**,** a modified version of SMO called as Quadratic Approximation based Spider Monkey Optimization (QASMO) has been proposed. The aim of this proposed approach is to improve the exploitation ability of SMO by incorporating quadratic approximation operator in it. The performance of QASMO is evaluated on the same benchmark set with the same evaluation criteria as mentioned in section 4.3 of chapter 4 and the results have been compared with basic SMO. The comparison of results on numerical (tables), graphical (convergence graphs and performance index) and statistical (t-test) grounds states that QASMO performs better than SMO over most of the scalable optimization problems, while its performance is moderate SMO over non-scalable problems. At the end of the chapter, the comparison has been made among SMO, TS-SMO and QASMO and it has been found that QASMO performs best among all the three on most of the optimization problems in the benchmark set.

**CHAPTER 6. QUADRATIC APPROXIMATION BASED CONSTRAINED SPIDER MONKEY OPTIMIZATION ALGORITHM:** In this chapter, a modified version of CSMO has been proposed. This modified version is named as Quadratic Approximation based Spider Monkey Optimization (QACSMO). This proposed approach is motivated from QASMO proposed chapter 5. The reason behind the incorporation of Quadratic approximation is the conclusion made at the end of the chapter 5 which states that QASMO performs better than SMO and TS-SMO. The performance of QACSMO has been evaluated over constrained benchmark problems and results are compared with CSMO.

**CHAPTER 7. APPLICATION OF SPIDER MONKEY OPTIMIZATION TO SOLVE LENNARD-JONES PROBLEM:** Lennard-Jones (L-J) problem is an optimization problem from computational chemistry which deals with finding the relative position of atoms in a cluster in the three dimensional Euclidean space in such a way that that potential energy is minimum. The main obstacles in solving L-J problem is the non-linearity and non-convexity of the objective function and exponentially increasing number of local minima with increase in the number of dimensions. Despite these difficulties, solution of this problem is very important to facilitate drug design, synthesis and utilization of pharmaceutical products. In this chapter, clusters of 3 to 10 atoms have been considered for the experiment. SMO, TS-SMO and QASMO have been applied to solve Lennard-Jones problem and the results have been compared numerically (tables), graphically (convergence graphs and performance indices) and statistically (t-test). The conclusion of the results implies two things. One is that all the three algorithms applied here to solve L-J problem attain the solution close to the optimal solution. Second is that QASMO performs best among all the three algorithms.

**CHAPTER 8. APPLICATION OF CONSTRAINED SPIDER MONKEY OPTIMIZATION TO SOLVE PORTFOLIO OPTIMIZATION PROBLEM:** Portfolio optimization is an optimization problem from finance which deals with deciding the proportion of wealth to be invested in a portfolio such that the return is maximized and the risk can be minimized. In this chapter, we have considered the Markowitz's Mean Variance portfolio optimization model for the experiment and the data has been taken from National Stock Exchange (NSE), Mumbai. Data of 11 retail industries listed on NSE for the financial year 2015-16 has been taken for the experiment. CSMO has been applied to solve portfolio optimization problem. The results have been presented numerically (tables) and graphically (Efficient Frontier). The conclusion of the results shows that CSMO solves this problem efficiently.

**CHAPTER 9. CONCLUSION AND FUTURE SCOPE:** This chapter provides the chapter wise summary as well as concluding remarks of the proposed algorithms. At the end, some future directions are suggested to carry out research on this algorithm.

There are 4 Appendices in the thesis. The contents of these Appendices are given below:

**Appendix I:** LIST OF CONSTRAINED BENCHMARK PROBLEMS FROM IEEE CEC 2006

**Appendix II:** LIST OF CONSTRAINED BENCHMARK PROBLEMS FROM IEEE CEC 2010

**Appendix III:** LIST OF UNCONSTRAINED BENCHMARK PROBLEMS

**Appendix IV:** PERFORMANCE INDEX (PI)

# CHAPTER 2

# SPIDER MONKEY OPTIMIZATION: CONCEPT, DEVELOPMENT AND LITERATURE REVIEW

SMO is a new member of the class of swarm intelligent optimization algorithms. It is inspired by the social behaviour of spider monkeys. SMO was first introduced by Bansal et al. [17]. This algorithm is easy to understand and implement. It is based on the learning, information sharing and position updating strategy adopted by spider monkeys for the search of their food. It can be used to solve various kinds of function optimization problems or the problems which can be transformed to function optimization problems. SMO has been designed in such a way that it can handle the problem of stagnation or premature convergence very efficiently.

The present chapter has been divided into following sections: Section 2.1 provides an introduction to swarm intelligent optimization algorithms. In section 2.2, social behaviour of spider monkeys and the development of SMO have been discussed. Section 2.3 provides comprehensive literature review on the advancement and applications of SMO. The chapter has been concluded in section 2.4.

## 2.1 SWARM INTELLIGENT OPTIMIZATION ALGORITHMS

Swarm intelligent optimization algorithms belong to the class of metaheuristics. The term swarm refers to the group of creatures like birds, ants, animals etc. The members of the swarm are known as agents. There is no central authority to supervise the behaviour of these agents. Each of these agents learns from the surrounding agents and makes advances according to this learning. This social learning and adaptation lead exhibit a kind of intelligent behaviour known as swarm intelligence. Swarm intelligence (SI) is defined as the collective intelligent behaviour of decentralized and self-organized swarms. Commonly known examples of such behaviour are bird flocks and social insects like ants, bees, termites etc. The intelligence emerges from the pattern of interaction among these agents and the environment. According to Bonabeau, swarm intelligence is “any attempt to design algorithms or distributed problem solving devices inspired by the collective behaviour of social insect colonies and other animal societies” [23]. Self-organization and division of labour have been stated as key features of swarm systems in [23].

**Self-organization**: It is a process where a global pattern takes place by local interactions among agents. There is no central authority to regulate the swarm. Therefore, the whole structure of the swarm is decentralized and distributed among all the agents. In [23], four following characteristics of self-organisation in a swarm have been defined:

1. **Positive feedback**: It refers to deriving the information from the output and reapplying it as input for the further process. It helps in creating the convenient structures based on the information received.

2. **Negative feedback**: It counterbalances the effect produced by positive feedback and keeps the system stabilized. It helps in avoiding the saturation caused by the positive feedback.

3. **Fluctuations**: It refers to the random changes being made in the swarm including random walks, random task switching among the members of the swarm. Randomness is significant with a view to improve creativity as it facilitates the discovery of new solutions.

4. **Multiple interactions**: It refers to a way of learning by sharing the information among the individuals of the swarm and thus helps in increasing the combined intelligence of every individual involved in the interaction.

**Division of labour**: It refers to dividing the whole task into different smaller tasks and assigning those different smaller tasks to specialized agents of the swarm. The objective behind the division of labour is to improve the overall efficiency by getting those smaller tasks done from the specialized agents. It is believed that simultaneous task performance by the cooperation of specialized individuals improves the overall work efficiency in comparison to the same task performed sequentially by the unspecialized agents.

There are various types of swarms exist in this world. It is not possible to consider all of them as intelligent. The first and foremost condition to develop a swarm intelligence based optimization algorithm is to see whether the collective behaviour exhibit by the swarm of creatures is intelligent or not. According to Millonas [121], the following five principles should be satisfied in order to consider swarm behaviour as intelligent:

**1. The proximity principle**: The swarm should be able to do basic space and time computation.

2. **The quality principle**: The swarm should be able to compute quality factors in the environment like quality of the food stuff, safety of locations etc.

3. **The principle of diverse response**: The swarm should maintain diversity in the allocation of its resources in order to gain insurance against the sudden changes may be due to environmental fluctuations.

4. **The principle of stability**: The swarm should not change its mode of behaviour in response to every environmental fluctuation as it may not produce worthy results every time.

5. **The principle of adaptability**: The swarm must be able to change its mode of behaviour if it results in something worthy.

Swarm intelligent optimization algorithms are developed by simulating the social or foraging behaviour of the swarm of creatures like birds, ants, animals etc. Social learning and adaptation behaviour of these creatures are the motivation behind the development of swarm intelligent algorithms which are meant to solve various types of complex optimization problems. Some established swarm intelligent optimization algorithms are Particle Swarm Optimization (PSO) [49], Artificial Bee Colony (ABC) [86], Ant Colony Optimization (ACO) [46] etc. The main reason behind the growing interest of researchers from multiple fields in the development of swarm intelligent optimization algorithms is the naturally intelligent swarm performing the tasks (dividing the labour among the members of the swarm, sharing the information about the food source among each other etc.) which can be similar to the problem solving techniques in real world optimization problems. Some recently proposed swarm intelligent optimization algorithms include [11; 122; 123].

## 2.2 SPIDER MONKEY OPTIMIZATION

### 2.2.1 SOCIAL AND FORAGING BEHAVIOUR OF SPIDER MONKEYS

Spider monkeys belong to the class of fission-fusion social structure (FFSS) [172] based animals. They live in a group of 40-50 individuals. The animals in the category of fission-fusion social structure change the size and composition of their social group from time to time. For example, they split the group (fission) for food foraging and merge the group (fusion) for sleeping in one place.

Spider monkeys live in tropical rain forests of central and South America and in northern parts of Maxico. They are called spider monkeys because they look like spiders when they hang upside down from their tails with their arms and legs dangling. Tail is the most prominent feature of spider monkeys. They have a very long and gripping tail which is used as a fifth limb to facilitate their movement through the dense vegetation. They are social animals which belong to the category of fission-fusion structure based social animals. Being a part of fission fusion social structure, they live in groups and divide themselves into different subgroups to search for their food in different directions and then come together to share the food with the whole group. Spider monkeys survive mainly on plant based food like fruits, seeds, nuts and flowers. When fruits are not available, they eat bird’s eggs and insects. They are active during day and sleep during the night. Spider monkeys mainly live on treetops and very less on ground. They sleep on trees to avoid predators. They communicate through different sounds. They can survive up to 22 to 25 years in the wild and 35 years in captivity. It is also a species which is in danger as people hunt them for their food. Also, their living space is shrinking.

There is a female leader of the group known as global leader who is responsible for the availability of food to the group members. During the scarcity of food, the female leader divides the group into smaller subgroups. Each of these subgroups has a leader known as local leader. The spider monkeys in each subgroup search for food in different directions under the guidance of their local leader. Then these monkeys gather at a place to share the food they have collected with their group and sleep at night. All the above information regarding spider monkeys have been taken from [17; 167].

In case of spider monkeys, the four characteristics of self-organization can be expressed as follows:

1. **Positive feedback**: All the members a subgroup move in the direction of their local leader and the global leader.
2. **Negative feedback**: The exploitation process of poor food source is stopped by spider monkeys.
3. **Fluctuations**: Stagnated subgroups are redirected for discovering new food sources.
4. **Multiple interactions**: Local leaders and global leader share their information about food sources with other members of the group.

The foraging behaviour of spider monkeys has been re-examined and it has been found that it satisfies all the principles of a swarm intelligent behaviour stated by Millonas [121].

### 2.2.2 BASIC DEFINITIONS

**Definition 1: Search space**: It is the set of all the possible solutions of an optimization problem.

**Definition 2: Swarm**: The collection of solutions which perform the search for the optima in the search space.

**Definition 3: Spider Monkey**: An agent of the swarm who carries out the search process by keeping the track of its position in each iteration (generation).

**Definition 4: Position Vector**: A D-dimensional vector which represents the position of a spider monkey in the search space. Position of a spider monkey represents a solution of an optimization problem.

**Definition 5: Swarm size**: It is the total number of spider monkeys in the swarm.

**Definition 6: Fitness value**: It is the measure of the quality of the current position of the spider monkey. It indicates the quality of the solution which is represented by the position of a spider monkey.

**Definition 7: Fitness function**: It is a mathematical function $f : R^D \to R$ which provides the fitness value of the position of a spider monkey.

### 2.2.3 ALGORITHMIC STRUCTURE OF SMO

In SMO, solution swarm is compared to the swarm of spider monkeys. Search space of solutions is the food searching area of the spider monkeys. A solution of the optimization problem is represented by the position of a spider monkey. Position of a food source represents the optimal solution to the problem and amount of food available at the food source corresponds to the quality of the associated solution. Movement of the swarm of spider monkeys in the search region is equivalent to the improvement in the quality of the solutions in the search space. The objective is to move in the search space in such a way that the food source is found or equivalently the optimal solution is to be found.

The nomenclature used for the SMO algorithm is given below:

**Nomenclature**

| | |
|---|---|
| $N$ | Swarm size |
| $D$ | no. of dimensions |
| *U(a,b)* | uniformly generated random number between a and b |
| $NG$ | number of groups in the current swarm |
| $MG$ | maximum number of groups allowed in the swarm |
| $Pr$ | perturbation rate |
| *GS[k]* | number of members in the *k*th group |
| *G[k][0]* | index of the first member of the *k*th group in the swarm |
| *G[k][1]* | index of the last member of the *k*th group in the swarm |
| $SM_i$ | position vector of *i*th spider monkey in the swarm |
| $SM_{new}$ | a trial vector for creating a new position of a spider monkey |
| $SM_{newlocal}$ | a trial vector for creating a new position of a spider monkey in Local Leader Phase |
| $SM_{newglobal}$ | a trial vector for creating a new position of a spider monkey in Global Leader Phase |
| $SM_r$ | position vector of randomly selected member of the group |
| $SM_{worstglobal}$ | position vector of worst member of the swarm in Global Leader Learning Phase |
| $SM_{worstlocal}$ | position vector of worst member of a group in Local Leader Learning Phase |
| $LL_k$ | position vector of local leader of *k*th group |

| | |
|---|---|
| $GL$ | position vector of global leader of the swarm |
| $sm_{minj}$ | lower bound on the *j*th decision variable |
| $sm_{maxj}$ | upper bound on the *j*th decision variable |
| $sm_{ij}$ | *j*th decision variable of *i*th spider monkey |
| $sm_{newj}$ | *j*th decision variable of new trial position of spider monkey |
| $fit_i$ | fitness of the position of *i*th spider monkey |
| $maxfit$ | best fitness value in the group |
| $prob_i$ | probability of the position of *i*th spider monkey |
| $GLlt$ | global leader limit |
| $LLlt$ | local leader limit |
| $GLC$ | limit count of global leader |
| $LLC_k$ | limit count of local leader of *k*th group |

SMO has been developed by simulating the food searching behaviour of spider monkeys. There are four control parameters of SMO namely perturbation rate (*Pr*), Maximum number of groups (*MG*), Global Leader Limit (*GLlt*) and Local Leader Limit (*LLlt*).

1. **Perturbation rate *(Pr)***: It decides the amount of change in the current solution.
2. **Maximum number of groups *(MG)***: It decides the maximum number of groups that can be formed in the entire swarm.
3. **Global leader limit (*GLlt)*:** It checks for stagnation in the swarm.
4. **Local leader limit *(LLlt*):** It checks for stagnation in the local groups.

In addition to initialization of the swarm, SMO has six iterative steps namely Local Leader Phase (LLP), Global Leader Phase (GLP), Global Leader Learning Phase (GLLP), Local Leader Learning Phase (LLLP), Local Leader Decision Phase (LLDP) and Global Leader Decision Phase (GLDP).

Before the description of each step of the algorithm in detail, here are some of the terms which have been used during iterative steps in different phases. A D-dimensional trial vector say $SM_{new} = (sm_{new1}, sm_{new2}, \dots, sm_{newD})$ for creating a new position of a spider monkey. $LL_k = (ll_{k1}, ll_{k2}, \dots, ll_{kD})$ is the position vector of the local leader of the $k^{th}$ group and $GL = (gl_1, gl_2, \dots gl_D)$ is the position vector of the global leader of the entire swarm. Also, it should be noted that updation in the position of a spider monkey is done dimension wise. If during updation process, value of a decision variable is out of predefined limits (lower and upper bounds), then the value of that decision variable can be set to the predefined limit or randomly between the predefined limits. Detailed description of each phase is given below:

**Initialization of the swarm**

During initialization of the swarm, SMO generates a uniformly distributed D-dimensional initial positions vectors of $N$ spider monkeys. The position $SM_i = (sm_{i1}, sm_{i2}, \dots, sm_{iD})$ of $i^{th}$ spider monkey is initialized using the following equation:

$$sm_{ij} = sm_{minj} + U(0,1) \times (sm_{maxj} - sm_{minj}) \tag{2.1}$$

**Local Leader Phase**

In this phase, a new trial position say $SM_{new} = (sm_{new1}, sm_{new2}, \dots, sm_{newD})$ is generated for each Spider Monkey based on the information of its current position, local leader's experience as well as local group members' experience. The fitness value of the newly generated position of the $i^{th}$ spider monkey is compared with that of its old position. If the fitness of the newly generated position is higher than that of old position, then the spider monkey updates its position with the new one, otherwise it retains its old position.

Position update equation in this phase is given below:

$$sm_{newj} = sm_{ij} + U(0,1) \times \left(ll_{kj} - sm_{ij}\right) + U(-1,1) \times (sm_{rj} - sm_{ij}) \tag{2.2}$$

Here, $SM_r = (sm_{r1}, sm_{r2}, \dots sm_{rD})$ is the position of the randomly selected member of the current group. Though, it is a randomly selected member of the group, yet it should be different from the member of the group which has to be updated. The execution steps of this phase are given in Algorithm 2.1.

```
Begin
    For $k$ = 1 to $NG$ Do
        For $i = G[k][0]$ to $G[k][1]$ Do
            For $j$ = 1 to $D$ Do
                If $U(0, 1) \geq Pr$ Then
                    $sm_{newj} = sm_{ij} + U(0,1) \times (ll_{kj} - sm_{ij}) + U(-1,1) \times (sm_{rj} - sm_{ij})$
                Else
                    $sm_{newj} = sm_{ij}$
                End If
            End For
            If $(fit_{new} > fit_i)$
                $SM_i = SM_{new}$
            End If
        End For
    End For
End
```

**Algorithm 2.1**: Local Leader Phase

**Global Leader Phase**

In this phase, a new trial position is created for each spider monkey using its own experience as well as the information from the experience of the global leader and other members of the group. In this phase, a spider monkey gets a chance to update its position based on its probability which is directly proportional to the fitness of its current position.

The probability of the $i^{th}$ spider monkey in the swarm has been calculated using following expression

$$prob_i = 0.9 * \left(\frac{fit_i}{maxfit}\right) + 0.1 \tag{2.3}$$

The position update process of this phase is given in Algorithm 2.2. From the update process in this phase, it is clear that spider monkeys having higher fitness values will have better chance to improve their position as compared to other members of the swarm.

```
Begin
    For $k$ = 1 to $NG$ Do
        let count = 1;
          While count < $GS[k]$ Do
              For $i = G[k][0]$ to $G[k][1]$ Do
                  If ($U(0, 1) < prob[i]$) Then
                          count = count + 1.
                          Randomly select j ∈ {1...$D$}.
                          Randomly select $SM_r$ from $k^{th}$ group s.t. $r \neq i$
                          $sm_{newj} = sm_{ij} + U(0,1) \times (gl_j - sm_{ij}) + U(-1,1) \times (sm_{rj} - sm_{ij})$
                  End If
                  If ($fit_{new} > fit_i$)
                      $SM_i = SM_{new}$
                  End If
              End For
              If ( $i = G[k][1]$ ) Then
                  $i = G[k][0]$
              End If
          End While
    End For
End
```

**Algorithm 2.2**: Global Leader Phase

### Global Leader Learning Phase

In this phase, position of the global leader is updated by applying greedy selection in the swarm. A spider monkey whose position is having best fitness value will be updated as global leader of the swarm and the position of global leader will be known as the global best solution. This phase has been described in Algorithm 2.3. *GLC* records how many times position of the global leader has not been updated since last updation. This information is useful to check if there is stagnation in the group.

```
Begin
   //update position of the global leader of the swarm by applying greedy selection
   If(position of global leader is updated from previous position) Then
      $GLC$ = 0
   Else
      $GLC = GLC$ + 1
   End If
End
```

**Algorithm 2.3**: Global Leader Learning phase

**Local Leader Learning Phase**

In this phase, position of every local leader is updated by applying greedy selection in the group it belongs. A spider monkey whose position is having best fitness value in the group will be updated as local leader of that group. This phase has been described in Algorithm 2.4. $LLC_k$ records how many times position of the local leader of $k^{th}$ group has not been updated since last updation.

```
Begin
    For k=1 to $NG$ Do
       //update position of the leader of the group
        If(position of local leader is updated from previous position) Then
           $LLC_k$= 0
        Else
           $LLC_k = LLC_k$ + 1
        End If
    End For
End
```

**Algorithm 2.4**: Local Leader Learning phase

## Local Leader Decision Phase

If the limit count of the local leader of $k^{th}$ group ($LLC_k$) exceeds its local leader limit ($LLlt$), then all the members of the group will be re-initialized. Local Leader Limit is predefined. Algorithm 2.5 shows the Re-initialization process in the Local Leader decision phase.

```
Begin
  For $k$ = 1 to $NG$ Do
    If ($LLC_k > LLlt$) Then
        $LLC_k = 0$
        For $i = G[k][0]$ to $G[k][1]$ Do
          For $j$ = 1 to $D$ Do
            If ($U(0,1) \geq Pr$) Then
              $sm_{ij} = sm_{minj} + U(0,1) \times (sm_{maxj} - sm_{minj})$
            Else
              $sm_{ij} = sm_{ij} + U(0,1) \times \left(gl_j - sm_{ij}\right) + U(0,1) \times (sm_{ij} - ll_{kj})$
            End If
          End For
        End For
    End If
  End For
End
```

**Algorithm 2.5:** Local Leader Decision phase

## Global Leader Decision Phase

If the limit count of the global leader of the swarm (*GLC*) exceeds its global leader limit (*GLlt*), then the swarm is divided into groups. Global leader limit is predefined. This procedure has been explained in Algorithm 2.6.

Pseudocode for SMO has been provided in Algorithm 2.7.

```
Begin
    If ($GLC > GLlt$) Then
        $GLC = 0$
        If ($NG < MG$) Then
            $NG = NG+1$
        Else
            $NG = 1$
        End If
        Apply Local Leader Learning Phase
    End If
End
```

**Algorithm 2.6**: Global Leader Decision phase

```
Begin
    Initialize the swarm using equation (2.1)
    Initialize $LLlt, GLlt, Pr, MG$
    Iteration = 0
    Calculate fitness value of the position of each spider monkey in the swarm
    Select global leader and local leaders by applying greedy selection
    While (termination criterion is not satisfied) do
        //Local Leader Phase
        //Calculate Probability of each spider monkey
        //Global Leader Phase
        //Global Leader Learning Phase
        //Local Leader Learning Phase
        //Local Leader Decision Phase
        //Global Leader Decision Phase
        Iteration = iteration +1
    End While
End
```

**Algorithm 2.7:** Pseudocode for SMO

## 2.3 LITERATURE REVIEW ON SMO

SMO is very easy to understand and implement. So, it can be applied directly to solve optimization problems. It has been shown in Bansal et al. [17] that SMO has been very effective in producing competitive results at low computational cost. Such results have been a source of inspiration for the researchers to explore more about the potential of SMO. So, many studies have been carried out after the initial publication on SMO. The number of publications on SMO is increasing every year showing the rising interest of research community in this algorithm. These publications can be grouped together in three categories: modifications and comparisons, hybridisation with other algorithms and applications. The literature on SMO is reviewed under these subtitles. In the entire thesis, the first version of SMO [17] has been called as basic SMO.

### 2.3.1 MODIFICATIONS AND COMPARISONS

In its original form, SMO has been designed to solve unconstrained continuous optimization problems. Therefore, the objective of the first studies on SMO was to evaluate its performance on a set of unconstrained continuous benchmark problems and to compare it with other well-established stochastic optimization algorithms such as DE [168], PSO [32], ABC [86] and CMA-ES [72]. A benchmark set of 26 scalable and non-scalable problems was considered for the experiment. The results were compared numerically and statistically and it was concluded that SMO is a strong competitor of the algorithms used for comparison.

Exploitation and exploration are two important characteristics of a metaheuristic algorithm. Exploration facilitates the search of new random solutions in different regions of the search space, while exploitation helps to refine the quality of the current solution by searching for better solutions in its neighbourhood. These two features seem to be contradictory in nature and performance of a metaheuristic algorithm depends heavily on the balance between them. Though, SMO has been designed in such a manner to maintain a proper balance between these two, still modifications have been made to improve its performance by working on its exploration and exploitation ability.

Kumar and Kumari [100] proposed first modified version of SMO with an objective to enhance its performance by using golden section search. This proposed version was named as Modified Position Update in Spider Monkey Optimization (MPU-SMO). Modifications were made in the position update equations in Local Leader Phase and Global Leader Phase by adding a new component in them. This new component contained a variable whose value was determined by using golden section search method. Same parameter setting was adopted for MPU-SMO as described for SMO in Bansal et al. [17]. The performance of proposed algorithm was tested over a benchmark set of nine unconstrained continuous optimization problems and results were compared with basic SMO.

Kumar et al. [101] proposed another modification of SMO with the name Self-adaptive Spider Monkey Optimization (SaSMO). Modifications were made in the Local Leader Phase, Global Leader Phase and Local Leader Decision Phase. The performance of the proposed algorithm was tested on a benchmark set of 42 optimization problems. Results were compared with basic SMO [17] and MPU-SMO [100].

Kumar et al. [102] proposed a fitness based location update strategy in SMO to improve its exploitation ability. The proposed version was named Fitness Based Position Update in SMO (FPSMO) as it updates position of the individuals based on their fitness. Position update equations in Local Leader Phase, Global Leader Phase and Local Leader Decision Phase were modified to implement the proposed strategy. The objective behind choosing the fitness based position update scheme was to increase the convergence speed of SMO. The performance of FPSMO was tested over a benchmark set of 19 unconstrained continuous optimization problems and the results were compared with basic SMO.

Singh and Salgotra [163] proposed MSMO to improve the performance of basic SMO. The proposed algorithm was tested on a benchmark set 19 optimization problems. Modifications were made in Local Leader Phase, Global Leader Phase and Local Leader Decision Phase. The results showed that MSMO performed better than other state-of-the-art algorithms used for comparison.

Sharma et al. [153] proposed variant of SMO named as Ageist Spider Monkey Optimization (ASMO) algorithm. This proposed variant was inspired from the idea that a group of spider monkeys contains members of different age groups which looked more practical in biological terms. Age was a prominent factor in deciding the performance of a spider monkey. Experiments with different parameter settings were carried out and the best suitable parameter setting was chosen. Two variants of SMO named as ASMO and AMSMO were proposed. In ASMO, modification was made only in Local Leader Phase and in AMSMO, modifications were made in Local Leader Phase and Global Leader Phase. The results of ASMO and AMSMO were compared with basic SMO. Results demonstrated the improvement in convergence rate of SMO with these modifications. Though it can be observed proposed variants converged much faster than the original SMO, there is not much difference among the variants (ASMO and AMSMO). AMSMO with 4 mini groups turned out to be the best from the results and hence been chosen as a base for statistical comparison. Also, AMSMO is compared with five state-of-the art algorithms and significance of results was validated by Wilcoxon signed rank test. Also, AMSMO was compared with two newly proposed variants of SMO namely MPU-SMO [100] and SaSMO Kumar et al. [101]. Also, AMSMO was compared with MVMO which is the winner of CEC2014 on nine benchmark functions. Conclusion of results demonstrated the positive effect of considering age factor in the movement of spider monkeys.

Sharma et al. [155] proposed Limacon inspired Spider Monkey Optimization (LSMO) for function optimization. In LSMO, there was one more phase in addition to the phases of original SMO. This phase was executed after Global Leader Decision Phase. The results had been compared with some state-of-the-art algorithms.

Sharma et al. [154] used power law based local search strategy (PLSS) in SMO to improve the exploitation ability of basic SMO. The proposed version was named as power law based SMO (PLSMO). There were seven phases in PLSMO. PLLS was introduced after Global Leader Decision Phases. The performance of PLSMO was tested over a benchmark suite of 20 problems and results were compared with basic SMO. Results demonstrated that the PLSS made a positive effect on the performance of SMO.

Agarwal and Jain [1] proposed fast convergent spider monkey optimization algorithm (FCSMO) by using a new acceleration coefficient based strategy with an objective to improve the exploitation ability of basic SMO. Modifications were made in the update equations in Global Leader Phase and Local Leader Decision Phase. The performance of FCSMO was tested over 14 benchmark functions. The results were compared with basic SMO and it was concluded that the proposed variant is a better version of basic SMO.

Hazrati et al. [74] proposed SMO based on metropolis principle (SMOM) with an objective to improve the exploration ability of basic SMO. The performance of the proposed algorithm was evaluated on a set of 12 benchmark problems and results were compared with self-adaptive spider monkey optimization (SaSMO) and particle swarm optimization (PSO). It was concluded that the proposed variant is a better version of basic SMO.

Hazrati et al. [73] proposed Adaptive size based Spider Monkey Optimization (AsSMO) algorithm with modifications proposed in Global Leader Phase and Global Leader Decision Phase. Update equations were modified in both the phases with an objective to have better solutions. The performance of the proposed algorithm was evaluated on 15 benchmark problems and results were compared with SMO and AsSMO. The results concluded that the proposed modification had positive impact on the performance of SMO.

Dhar and Arora [41] proposed a cooperative Spider Monkey Optimization algorithm. The proposed strategy dealt with implementing SMO in a cooperative framework with two slight modifications known as steady state update and re-initialization of least performing monkeys. The performance of the proposed algorithm was validated by its application on a real life problem.

The success of SMO for solving continuous optimization problems has motivated the researchers to extend its use to solve other types of optimization problems also. Singh et al. [164] proposed a variant of SMO called binSMO for solving binary optimization problems. This variant made use of logical operators to represent the solutions. This proposed version was applied to optimize the thinning of concentric circular antenna arrays. The results were compared with six state-of-the-art algorithms and it was concluded that binSMO performed better than the other algorithms.

Arora et al. [13] made conceptual comparison among basic SMO [17], HPSOWM [108] and Krill Herd [57].

### 2.3.2 HYBRIDIZATION

In order to improve the efficiency of SMO, Agrawal et al. [2] hybridized SMO with GA, which is an evolutionary algorithm. Two hybridized versions of SMO and GA namely SMOGA (SMO followed by GA) and GASMO (GA followed by SMO) were proposed. The performances of the proposed algorithms were compared with GA and basic SMO.

### 2.3.3 APPLICATIONS

SMO has been applied to solve many real world optimization problems in different areas. Researchers have used original versions and its modified version to solve these real world optimization problems. Kumar et al. [101] employed SaSMO to solve four real world problems namely Pressure Vessel Design Problem, Lennard-Jones Problem, Parameter estimation for frequency-modulated sound wave and Compression Spring Problem. FPSMO [102] was applied to solve pressure vessel design problem. The objective was to minimize the overall cost of the manufacturing of cylinders. Later, Lenin et al. [106] proposed a variant of SMO called Modified Monkey Optimization (MMO), but the modification proposed in this algorithm was exactly the same as proposed by Kumar and Kumari [100] using golden section search, so, it cannot be considered as a new modification. MMO was applied to solve the optimal reactive dispatch problem on standard IEEE 30 bus test system. Further, Pal et al. [135] applied SMO to the problem of multilevel thresholding segmentation. Standard images were used to test the performance of SMO and results were compared with PSO. Al- azza et al. [3] applied SMO for solving antenna optimization problem and it was found that SMO was most efficient in reaching the optima as compared to other algorithms reported in the literature to solve this problem. Singh and Salgotra [163] used MSMO to solve the problem of synthesis of linear antenna array for three different cases. Singh et al. [164] employed binSMO to optimize the thinning of concentric circular antenna arrays. Sharma et al. [154] made use of the PLSMO to solve lower order system modelling problem. Further, Sharma et al. [155] employed LSMO to solve problem of optimal placement and sizing of capacitors. Cheruku et al. [31] designed SMO based ruler miner called as SM-RuleMiner for diabetes classification. Mittal et al. [125] used Boolean SMO in Wireless sensor networks to improve the network lifetime with an objective to extend the stability period of the network. Selvam and Kumar [152] used a combination of ley flights and SMO for optimizing the gains of PI controllers employed in the frequency regulating circuit of micro grid. The proposed strategy had been validated by two case studies. Sivalingam and Chinnamuthu [165] used hybrid SaSMO for designing PID controllers for automatic generation control. The results were compared with DE and SMO. Ali [4] used MDSMO for economic load dispatch optimization problem. Kaur et al. [89] used a combination of PSO and SMO for image compression problem. Dhar and Arora[41] applied cooperative SMO to optimize a fuzzy rule base.

## 2.4 CONCLUSIONS

From the discussion in section 2.3, it can be concluded that the growth of this field has exceeded the expectations, despite the fact that SMO is just three years old. By looking at the literature available on this topic, it can be concluded that the core of the work on SMO has focused on algorithmic and application aspects, it should be mentioned that there is still much more to do in this area. We believe that some topics are worth investigating within the next years. Particularly, self-adaptation of control parameters and theoretical studies are the first topics to be interested in. The design of SMO with no parameters that have to be fine-tuned by the user is another topic which is worth studying. There is not much theoretical work on SMO in general and the lack of research on theoretical aspects of SMO is, by no means, surprising. It would be interesting to perform a theoretical study of the run-time and convergence properties of a SMO. Other aspects such as the fitness landscapes and dynamics of a SMO are also very attractive theoretical research topics.

CHAPTER 3

# NOVEL SPIDER MONKEY OPTIMIZATION ALGORITHM FOR CONSTRAINED OPTIMIZATION

Most of the real world optimization problems are constrained in nature. Solving such optimization problems involves finding the values of a set of decision variables such that the objective function value is optimized while satisfying all the constraints and variable bounds. Constrained nonlinear optimization problems are more difficult to solve in comparison to unconstrained optimization problems because of the presence of additional conditions called constraints. In constrained optimization problems, the aim is not only to find the optimal solution but also to keep the focus on the feasibility of the solutions. Due to the inability of the traditional optimization algorithms to solve these complex problems, metaheuristics have been applied in the past to solve constrained optimization problems [15; 25; 51; 58; 75; 82; 83; 116; 119; 120; 127; 140; 169; 191].

In Bansal et al. [17], it has been concluded that SMO is showing good performance in solving unconstrained optimization problems. So, its ability should be extended for solving constrained optimization problems. In this chapter, an attempt has been made in this direction by designing a modified version of basic SMO for solving constrained optimization problems. To the best of author's knowledge, this is the first attempt to design a version of SMO to solve constrained optimization problems. The proposed algorithm is named as Constrained Spider Monkey Optimization (CSMO). The performance of CSMO is investigated on well-defined constrained optimization problems of IEEE CEC2006 and CEC2010 benchmark sets. The results of the proposed CSMO are compared with constrained versions of Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC) and Differential Evolution (DE) using various performance metrics. The chapter is organized as follows: Section 3.1 gives brief description about constraint handling techniques. Section 3.2 discusses the proposed algorithm. Section 3.3 provides the details of experimental setup. Section 3.4 discusses the experimental results of the tested algorithms based on various evaluation criteria. The chapter is closed with concluding remarks in section 3.5.

---

The content of this chapter contributes to the paper entitled:

**Gupta, K.**, Deep, K., & Bansal, J. C. (2016). Spider monkey optimization algorithm for constrained optimization problems. *Soft Computing* (**IF. 2.472**), Springer, 1-30.

## 3.1 CONSTRAINT HANDLING TECHNIQUES

Search algorithm and constraint handling technique both play an important role in solving constrained optimization problems. Since metaheuristics are formulated in a way in their indigenous design that they are suitable only for unconstrained optimization, various constraint handling techniques have been developed over the decades to be incorporated in these algorithms to remove this deficiency. Though the method for handling constraints is different in all the constraint handling techniques, yet the objective of mostly all of these techniques is same. The objective is to prefer feasible solutions over infeasible solutions. This objective is the key force to drive the search towards feasible region of the search space. Each constrained handling technique has its own advantages and disadvantages. So, which constrained handling technique is the most suitable for a particular metaheuristic algorithm is still an open research problem. Coello [33] presents a comprehensive survey of different constraint handling techniques which has been developed to incorporate constraint handling mechanism in metaheuristics. In [33], constraint handling techniques have been divided into five categories: 1) penalty approach, 2) special representations and operators, 3) repair algorithms, 4) separation of objective function and constraints and 5) hybrid methods. Each technique has been discussed in detail along with its usage in different metaheuristics. There is a vast literature available in which several constraint handling techniques have been proposed and developed, but there are only few constraint handling techniques which have been proved competitive in solving constrained optimization problems. Since its inception, Deb's technique is one of the most widely used constraint handling technique because of its ease of implementation and parameter free approach. Though, penalty approach is still popular in handling constraints, but requirement of additional penalty parameter sometimes make it difficult to implement. GA is the oldest, yet one of the most popular choices among researchers these days too.  Mostly used constraint handling mechanism with GA is the penalty approach. Most popular constraint handling technique with DE is Deb's feasibility rules. Also, with swarm intelligent techniques like PSO and ABC, Deb's technique for constrained optimization is the most popular one.

## 3.2 CONSTRAINED SPIDER MONKEY OPTIMIZATION (CSMO)

Deb's technique [35] has been used for constraint handling in CSMO. In spite of having so many constraint handling techniques existing in literature, there are some reasons for using Deb's technique in CSMO. This technique is easy to understand and implement. Since its inception, it is the most widely used constraint handling technique [33]. It has been used with various metaheuristics and has shown its potential over other constraint handling techniques. It does not require additional parameters. Also, it retains the original characteristics of the search algorithm in which it is used as a constraint handling mechanism.

### 3.2.1 DEB'S TECHNIQUE FOR CONSTRAINT HANDLING

Deb's technique is based on the following three rules, popularly known as *Three feasibility Rules*:

- A feasible solution is always preferred over an infeasible solution.
- Between two feasible solutions, the one having higher fitness value is preferred.
- Between two infeasible solutions, the one having less constraint violation is preferred.

Here, the constraint violation $viol_i$ of any solution $x$ for the constrained optimization problem P (1.1) defined in chapter 1 is calculated as follows:

$$viol_i = \sum_{j=1}^{p} G_j(\boldsymbol{x_i}) + \sum_{j=p+1}^{m} H_j(\boldsymbol{x_i}) \qquad (3.1)$$

such that

$$G_j(\boldsymbol{x_i}) = \begin{cases} g_j(\boldsymbol{x_i}) & if\ g_j(\boldsymbol{x_i}) > 0 \\ 0 & if\ g_j(\boldsymbol{x_i}) \leq 0 \end{cases}$$

$$H_j(\boldsymbol{x_i}) = \begin{cases} |h_j(\boldsymbol{x_i})| & if\ |h_j(\boldsymbol{x_i})| - \mathcal{E} > 0 \\ 0 & if\ |h_j(\boldsymbol{x_i})| - \mathcal{E} \leq 0 \end{cases}$$

Where, $\mathcal{E}$ is the tolerance limit for equality constraints.

$$\text{Mean violation at } \boldsymbol{x_i} = \frac{\left(\sum_{j=1}^{p} G_j(\boldsymbol{x_i}) + \sum_{j=p+1}^{m} H_j(\boldsymbol{x_i})\right)}{m} \qquad (3.2)$$

Where, *m* is the total number of constraints.

### 3.2.2 THE PROPOSED CONSTRAINED SPIDER MONKEY OPTIMIZATION (CSMO) ALGORITHM

The proposed CSMO differs from basic SMO only in two ways: calculation of fitness value and selection of the better solution while comparing two solutions. In basic SMO, fitness of a solution is based on its objective function value while in CSMO; fitness of a solution is based on whether the solution is feasible or infeasible. Steps for calculating fitness value in CSMO are given in Algorithm 3.1.

```
Begin
        For $i$ = 1 to $N$ Do
                If $(viol_i = 0)$ Then
                        $fit_i = f(SM_i)$
                Else
                        $fit_i = f_{worst} + \sum_{i=1}^{m} viol_i$
                End If
        End For

End
```

**Algorithm 3.1:** Steps for calculating fitness of solutions

In basic SMO, the comparison of two solutions is done on the basis of their objective function value. In CSMO, the comparison of two solutions is done on the basis of Deb's *Three feasibility rules* mentioned in subsection 3.2.1. Execution steps of the proposed algorithm are given below:

**Initialization:** This is the first step in the execution of the algorithm. CSMO does not make any assumption about the feasibility of the initial swarm as initialization of the swarm with feasible solutions may require high computational cost depending on the size of the feasible region. Also, it is near to impossible to generate initial feasible population in some cases where the ratio of feasible region to its search space is very small. So, in CSMO, initial swarm is randomly generated between lower and upper bounds of the decision variables using uniform distribution accepting both feasible and infeasible solutions. Initialization process is CSMO is same as that of basic SMO provided in eq. (2.1) in chapter 2.

**Local Leader Phase**: Local Leader Phase in CSMO is same as in basic SMO except the selection of better solution during comparison. In CSMO, position updating of spider monkeys is based on Deb's three rules of feasibility. New position and current position of a spider monkey are compared according to these rules and then updating or retaining of the current position of the spider monkey is performed. Working steps of Local Leader Phase is provided in Algorithm 3.2.

```
Begin
    For $k$ = 1 to $NG$ Do
        For $i$ = $G[k][0]$ to $G[k][1]$ Do
            For $j$ = 1 to $D$ Do
                If $U(0, 1) \geq Pr$ Then
                    $sm_{newj} = sm_{ij} + U(0,1) \times (ll_{kj} - sm_{ij})$
                        $+U(-1,1) \times (sm_{rj} - sm_{ij})$
                Else
                    $sm_{newj} = sm_{ij}$
                End If
            End For
            Apply the selection process between $SM_{new}$ and $SM_i$ based on Deb's
            Three Feasibility Rules
        End For
    End For
End
```

**Algorithm 3.2:** Local Leader Phase

**Global Leader Phase:** In this phase, a solution gets a chance to be updated based on its probability. Probability of a solution is based on its fitness value. In CSMO, fitness of every solution is calculated in a manner such that feasible solution will always get the preference over the infeasible solutions. Execution steps of Global Leader Phase are provided in Algorithm 3.3.

**Global Leader Learning Phase:** In this phase, a solution having best fitness value in the swarm is found using the three feasibility rules and updated as global leader of the swarm.

Procedure of selecting global leader is provided in Algorithm 3.4.

**Local Leader Learning Phase**: In this phase, local leader selection is performed in every group. In each group, the local leader of the group is selected by applying three feasibility rules. The member of the group having best fitness is selected as the local leader of that group. Selection procedure of local leaders is provided in Algorithm 3.5.

```
Begin
    For $k$ = 1 to $NG$ Do
        $GS$ = $k^{th}$ group size
        $t$ = 0, $i$ = 1
        While ($t$<N) Do
            For $i$ = 1 to $GS$ Do
                If ($U(0,1) < prob_i$) Then
                    $t = t+1$
                    Randomly select $j$ from $\{1,2,\dots, D\}$
                    Randomly select $SM_r$ from $k^{th}$ group
                    Generate $sm_{newj}$ using eq.(4)
                End If
                Apply the selection process between $SM_{new}$ and $SM_i$ based on
                Deb's Three Feasibility Rules
            End For
            $i = i+1$
            If ($i = N$) Then
                $i$ = 1
            End If
        End While
    End For
End
```

**Algorithm 3.3:** Global Leader Phase

**Local Leader Decision Phase:** This phase helps in the re-initialization of a group if its local leader is not updating its position for the specified local leader limit. Re-initialization of the group may cause the member of the group to enter into infeasible region from a feasible region. But it is also necessary sometimes if the feasible regions of the search space are disjoint and optimum lies in the other feasible region. This phase maintains the exploration capability of the algorithm. The execution steps of this phase are same as explained in Algorithm 2.5 in chapter 2.

```
Begin
        //update position of the global leader of the swarm by applying Deb's
         Three Feasibility Rules
         If (position of global leader is updated from previous position) Then
                  $GLC$ = 0
         Else
                  $GLC = GLC + 1$
         End If
End
```

**Algorithm 3.4:** Global Leader Learning Phase

```
Begin
         For $k$ =1 to $NG$ do
                  //update position of the leader of the group Deb's Three Feasibility Rules
                   If (position of local leader is updated from previous position) Then
                           $LLC_k = 0$
                   Else
                           $LLC_k = LLC_k + 1$
                   End If
         End For
End
```

**Algorithm 3.5:** Local Leader Learning Phase

**Global leader decision phase**: In this phase, the whole swarm is divided into groups if the global leader is not updated for the specified global leader limit. Execution steps of this phase are same as provided in Algorithm 2.6 in chapter 2.

Pseudocode of CSMO is provided in Algorithm 3.6.

```
Begin
  Initialize the swarm using eq. (2.1)
  Initialize LLlt, GLlt, Pr, MG
  Iteration = 0
  Calculate fitness value of the position of each spider monkey in the swarm using
  Algorithm 3.1
  Select Global Leader and Local Leaders by applying Deb's Three feasibility rules
  while (termination criterion is not satisfied) Do
      //Local Leader Phase (Algorithm 3.2)
      //Calculate Probability of each spider monkey (using eq. (2.3))
      //Global Leader Phase (Algorithm 3.3)
      //Global Leader Learning Phase (Algorithm 3.4)
      //Local Leader Learning Phase (Algorithm 3.5)
      //Local Leader Decision Phase (Algorithm 2.5)
      //Global Leader Decision Phase (Algorithm 2.6)
      Iteration = Iteration +1
  end while
End
```

**Algorithm 3.6** Pseudocode for CSMO

## 3.3 EXPERIMENTAL SETUP

### 3.3.1 BENCHMARK PROBLEMS AND EVALUATION CRITERIA

In this chapter, IEEE CEC2006 [107] and CEC2010 [112] benchmark sets have been considered for evaluating the performance of the algorithms. Both the sets contain single objective constrained optimization problems. These two sets have also been used for experimentation in [14; 138].

In CEC2006 benchmark set, there are 24 problems g01-g24. The number of design variables is different for each optimization problem. The mathematical definition of each problem has been given in Appendix I. Classification of problems on the basis of type of constraints has been provided in Table 3.1. As per the instruction given in CEC 2006 benchmark set, 25 independent runs have been performed for every problem. The stopping criteria is when 5, 00,000 number of function evaluations have been performed. For comparison of results, the feasibility rate and success rate is recorded. The formulas for calculating them are given below:

$$feasibility\ rate = \frac{number\ of\ feasible\ runs}{total\ number\ of\ runs} \times 100 \tag{3.3}$$

A run is said to be a 'feasible run' if atleast one feasible solution is found.

$$sucess\ rate = \frac{number\ of\ successful\ runs}{total\ number\ of\ runs} \times 100 \tag{3.4}$$

A run is declared as 'successful' if $|f(x) - f(x^*)| \leq 0.0001$, where $f(x)$ is the objective function value of the global leader and $f(x^*)$ is the known global optimal value.

Some statistical measures have been used for the comparison of results as suggested in [107]. For this purpose, best, median, worst, average and standard deviation of the function error values $|f(x) - f(x^*)|$ for the objective function value of global leader after 5,00,000 function evaluations in 25 runs are recorded. The method for sorting error values of achieved best solutions of all the runs is given below:

- Feasible solutions are given priority over infeasible solutions.
- Feasible solutions are sorted in an increasing order according to their function error value $|f(x) - f(x^*)|$
- Infeasible solutions are sorted in an increasing order according to their mean value of the violations of all constraints given in eq. (3.2).

Also, the best, median, worst, average and standard deviation of the number of function evaluations for the successful runs have been recorded.

There are 18 problems C01-C18 in CEC2010 benchmark set. The mathematical definition of each problem in this benchmark set has been given in Appendix II. Table 3.2 provides a classification of problems on the basis of type of constraints. Unlike CEC2006, in CEC2010, all the problems are of same dimension. In CEC2010, results have been taken for 10 dimensions and 30 dimensions. Optimal value of the test problems has not been provided. Fixed number of function evaluations for 10 dimensions and 30 dimensions are 2, 00,000 and 6, 00,000 respectively following the instructions given in CEC 2010 benchmark set. A total of 25 independent runs have been conducted for each problem. For comparison of results, feasibility rate and best, median, worst, average and standard deviation of objective function values of global leader in 25 runs have been calculated. Following criterion has been adopted to sort the objective function values of achieved best solutions obtained in 25 runs:

- Feasible solutions are given priority over infeasible solutions.
- Feasible solutions are sorted in an increasing order according to their objective function value
- Infeasible solutions are sorted in an increasing order according to their mean value of the violations of all constraints given in eq. (3.2).

Definition of mean value of constraints violations and feasibility rate is same for CEC2010 benchmark problems as mentioned in eq. (3.2) and eq. (3.3) respectively.

### 3.3.2 STATE-OF-THE-ART ALGORITHMS USED FOR COMPARISON OF RESULTS

The results of CSMO have been compared with ABC [87], CHDE [118] and PESO [128]. All the algorithms have been implemented in C. ABC, DE and PSO are some of the most widely used algorithms for solving constrained optimization problems and SMO has some features similar to these algorithms. These factors led to the selection of constrained versions of these algorithms (ABC, CHDE and PESO) for comparison with constrained SMO.

Since the performance of a metaheuristic is highly sensitive to the constraint handling technique used, it will be more appropriate to compare the performance of SMO with those versions of these algorithms which have used Deb’s technique for handling constraints. The constraint handling technique used in the algorithms selected for comparison is also Deb’s technique. Using the same constraints handling technique helps to maintain the consistency in the comparison of algorithms and to access the potential of an algorithm.

### 3.3.3 SETTING OF CONTROL PARAMETERS

Search efficiency of an algorithm is highly sensitive to the choice of its control parameters. The set of parameters which produces optimal solution for a particular problem may result in a complete failure for the other problem. So, optimal parameter setting is also an important issue while considering the performance of different algorithms. Parameter setting also requires analysis to be performed in order to find optimal parameter setting. In this chapter, such an analysis has been avoided and none of the algorithm has been meta optimized to improve its performance on a particular benchmark problem as the aim of this chapter is not to the find best algorithm for a particular benchmark problem, but to get an idea of search potential of CSMO for solving constrained optimization problems.

Parameter setting for every algorithm used for the experiment has been adopted as it is mentioned in their respective papers except the population size or swarm size and it is provided in Table 3.3. The parameter setting for CSMO has been kept same as that of basic SMO [17]. The population size or swarm size is kept same for all the algorithms for a fair comparison.

## 3.4 DISCUSSION OF EXPERIMENTAL RESULTS

The results of algorithms on both the benchmark sets (CEC2006 and CEC2010) have been discussed separately. First, the performance of algorithms have been analyzed on CEC2006 benchmark problems and then on CEC2010 benchmark problems. Results have been presented in the form of tables and graphs. In order to observe whether the results are significantly different or not, Wilcoxon rank sum test at 5% ($\alpha = 0.05$) significance level is performed between CSMO-ABC, CSMO-CHDE and CSMO-PESO for both CEC2006 and CEC2010 benchmark sets. The null hypothesis assumed for this statistical test is “if there is no difference in the performance of the algorithms” and alternative hypothesis being “there is a difference in the performance of the algorithms”. The test has been applied to the sample containing results of 25 independent runs performed by each algorithm for each benchmark problem. The output of the applied test has been presented in tabular form. If there is no significant difference between the results, ‘=’ sign appears and both the algorithms are considered equivalent. When there is significant difference between the results, '+' or '-' sign appears based on CSMO is performing better or worse than the other algorithm.

Tables 3.4-3.21, Tables 3.22-3.32 and Tables 3.33-3.43 present the results for CEC2006 benchmark problems, CEC2010 benchmark problems for 10 dimensions and CEC2010 benchmark problems for 30 dimensions respectively. The entries in the cells of these tables contain both feasible and infeasible solutions. Infeasible solutions are the entries which are followed by a parenthesis. Number in the parenthesis indicates the number of constraints violated by that infeasible solution. N.A. entry in a cell indicates non-availability of results. Pairwise comparison of CSMO has been done with ABC, CHDE and PESO on all the performance metrics considered for the evaluation of results. Summary of pairwise comparison is provided in the form of tables. Better, equal and worse in the summary table indicates the number of problems on which CSMO is better, equal or worse respectively than the compared algorithm. This pairwise comparison is inspired from Karaboga and Akay [87]. Also, the reason of doing pairwise comparison is that the aim of this experimental study is not to find the best algorithm for solving CEC2006 and CEC2010 benchmark problems but to access the performance of CSMO in comparison to other three algorithms. Convergence graphs for CEC 2006 and CEC2010 benchmark problems have been plotted in Figures 3.1-3.9. The graphs have been plotted using logarithmic scale as the range of values is very large.

## 3.4.1 DISCUSSION OF RESULTS FOR CEC2006 BENCHMARK PROBLEMS

Tables 3.4 and 3.5 present the results obtained by CSMO on CEC2006 benchmark problems. From Table 3.4, it can be seen that CSMO has 100 percent feasibility rate in seventeen problems (g01, g02, g03, g04, g06, g07, g08, g09, g10, g11, g12, g14, g15, g16, g18, g19, g24) and zero percent feasibility rate in three (g20, g21, g22) problems. In the problems having inequality constraints only (g01, g02, g04, g06, g07, g08, g09, g10, g12, g16, g18, g19, g24), CSMO has obtained 100 percent feasibility rate. Also, CSMO has good feasibility rate on the problems with equality constraints only (g03, g11, g13, g14, g15, g17). The problems where CSMO has failed completely to enter the feasible region in any run (g20, g21, g22) are the problems having both equality and inequality constraints. From Table 3.5, it can be seen CSMO has 100 percent success rate in six problems (g01, g04, g08, g12, g16, g24). Though, CSMO successfully enters the feasible region in the problems having equality constraints, it fails to obtain near optimal solution in any of these problems (zero percent success rate) except the problem g11. Among the problems, where CSMO has positive success rate are the ones having inequality constraints only except problem g11. Table 3.6 presents the feasibility rate and success rate of all the algorithms. It can be seen that all the algorithms have 100 percent feasibility rate on nine (g1, g2, g3, g4, g8, g12, g16, g19 and g24) and 100 percent success rate on two (g8 and g12) problems. g21 and g22 can be considered as difficult problems for these algorithms as all the algorithms fail to reach the feasibility region in all the runs. All the algorithms have zero percent success rate on eight (g3, g5, g10, g13, g15, g17, g21 and g22) problems. Summary of pairwise comparison of CSMO against each of ABC, CHDE and PESO in terms of feasibility rate and success rate has been provided in Table 3.7 which shows that CSMO has better feasibility rate than ABC and CHDE, but it performs inferior to PESO. The performance of all the algorithms is equivalent in terms of success rate. So, from the pairwise comparison of CSMO with all the algorithms in terms of feasibility rate and success rate, it can be concluded that though CSMO performs better than ABC and CHDE in locating the feasible region, it performs almost equal to these two in locating a near optimal solution. But this is just an observation which needs further experimentation. Tables 3.8-3.10 present the best, median and worst of function error values respectively obtained by all the algorithms in 25 runs. Tables 3.11 and 3.12 provide the mean and standard deviation of function error values obtained with feasible runs only. Summary of pairwise comparison of CSMO with ABC, CHDE and PESO on the basis of best, median, worst, mean and standard deviation values have been provided in Table 3.13 which shows that CSMO performs better than ABC, CHDE and PESO in terms of function error values on all the comparisons criteria.

Tables 3.14-3.18 show the best, median, worst, mean and standard deviation respectively of the number of function evaluations for successful runs of all the algorithms. Summary of pairwise comparison of CSMO with ABC, CHDE and PESO on the basis of number of function evaluations is provided in Table 3.19. From this table, it can be concluded that CSMO performs better than ABC, worse than CHDE and almost equivalent to PESO respectively.

Results of Wilcoxon rank sum test for CEC 2006 problems based on function error value is provided in Table 3.20. It can be observed from the outcome of Wilcoxon rank sum test that CSMO performs equivalent or significantly better than other algorithms on most of the problems.

Table 3.21 provides the average execution (in seconds) taken per run by all the algorithms. From this table, it can be observed that the time taken by all the algorithms is very small. It means all the algorithms solve the CEC2006 benchmark problems at a low computational cost. Also, there is a very small difference in the execution time of all the algorithms for each benchmark problem, so the comparison of these algorithms on the basis of execution time can be avoided. This table can be considered for information purpose rather than comparison purpose.

Convergence graphs for problems g01-g24 have been plotted in Figures 3.1-3.3. In the convergence graphs, value on the horizontal axis represents the number of iterations and the vertical axis shows the function error value. The logarithmic graphs have been plotted as the range of the function error values of the benchmark problems is large. The sudden rise in the graph indicates that the solution has entered the feasible region.

### 3.4.2 DISCUSSION OF RESULTS FOR CEC2010 BENCHMARK PROBLEMS FOR 10 DIMENSIONS

Results obtained by CSMO on CEC2010 problems for 10 dimensions have been presented in Table 3.22. From this table, it can be seen that CSMO has 100 percent feasibility rate in twelve problems (C01, C02, C03, C07, C08, C12, C13, C14, C15, C16, C17, C18). In three problems (C05, C06 and C11), CSMO fails to enter the feasible region in any run. CSMO has 100 percent feasibility rate in all the problems having inequality constraints only (C01, C07, C08, C13, C14, C15) and both inequality and equality constraints (C02, C12, C16, C17, C18). CSMO has either low or zero percent feasibility rate on the problems having equality constraints only (C03, C04, C05, C06, C09, C10, C11). Feasibility rate of all the algorithms has been provided in Table 3.23. Summary of pairwise comparison of CSMO against ABC, CHDE and PESO in terms of feasibility rate has been provided in Table 3.24. This table demonstrates that CSMO has higher feasibility rate in more number of problems than ABC and CHDE, while PESO outperforms CSMO.

Tables 3.25-3.27 present the best, median and worst of objective function values obtained by all the algorithms in 25 runs. Tables 3.28-3.29 present the mean and standard deviation of objective function values for feasible runs only. Summary of pairwise comparison of best, median, worst, mean and standard deviation of objective function value is provided in Table 3.30 and it can be seen from it that CSMO outperforms the other three algorithms. Results of Wilcoxon rank sum test based on objective function value for 10 dimensions problems is presented in Table 3.31. It shows CSMO performs significantly better than other algorithms on most of the problems.

Table 3.32 provides the average execution (in seconds) taken per run by all the algorithms. It can be seen from the table that the time taken by all the algorithms is almost similar. So, this table should be considered for information purpose rather than comparison purpose.

Convergence graphs for problems C01-C18 for 10 dimensions have been plotted in Figures 3.4-3.6. In the graphs, the fitness value of the global best solutions of all the algorithms at any iteration is plotted for all the algorithms. The fitness value here denotes the constraint violation if the global best solution lies in the infeasible region and the objective function value once the global best solution enters the feasible region. The sudden rise in the graph indicates that the solution has entered the feasible region.

### 3.4.3 DISCUSSION OF RESULTS FOR CEC2010 BENCHMARK PROBLEMS FOR 30 DIMENSIONS

Table 3.33 presents the result obtained by CSMO on CEC2010 problems for 30 dimensions. CSMO has 100 percent feasibility rate in ten problems (C01, C02, C07, C08, C13, C14, C15, C16, C17, C18). These are the problems with inequality constraints only and both inequality and equality constraints. Table 3.34 presents the feasibility rate of all the algorithms obtained for 30 dimensions. From Table 3.35, it can be seen that CSMO outperforms ABC and CHDE while performs inferior to PESO.

Tables 3.36-3.38 present the results for best, median and worst of objective function values in 25 runs respectively. Tables 3.39-3.40 present the mean and standard deviation of objective function values obtained in feasible runs only. From Table 3.41 which provides the summary of comparison, it is clear that CSMO perform better than the other three algorithms in terms of objective function values for 30 dimensions also.

Table 3.42 provides outcome of Wilcoxon rank sum test for 30 dimension problems showing CSMO performing significantly better than other algorithms on most of the problems.

Average execution time taken by all the algorithms for 30 dimensions has been given in Table 3.43. From this table, it can be seen that all the algorithms are taking more time for execution as compared to 10 dimensions because of the obvious reasons of increase in the dimension as well as more number of function evaluations. But there is not much difference among the four algorithms on most of the problems. So, this table should be considered for information purpose rather than comparison purpose.

Convergence graphs for problems C01-C18 for 30 dimensions have been plotted in Figures 3.7-3.9.

## 3.5 CONCLUSIONS

In this chapter, a constrained version of SMO named as CSMO has been proposed. Results have been compared against PESO [128], ABC [87] and CHDE [118] on CEC2006 and CEC2010 benchmark problems. Most of the results demonstrate supremacy of CSMO over compared algorithms. For both CEC2006 and CEC2010 benchmark sets, it can be concluded that CSMO enters the feasibility region in every run in the problems having inequality constraints only. But it finds it difficult to reach the feasibility region in problems having equality constraints only. Even if it enters, it fails to locate the near optimal solution as in case of CEC2006 problems. But this is just an observation which cannot be generalized. From the discussion of results, it can be observed that increase in the dimensions (10 to 30) does not deteriorate the performance of CSMO. CSMO has shown good performance on 30 dimensions even with the same population size. It is a general notion that performance of an algorithm generally deteriorates (though not always) with increase in the dimensions. This is commonly known as curse of dimensionality. But if we observe carefully, for problems C09, C10 and C11, feasibility rate though it is still less than 20 percent, has been improved. So, no comments can be made about the performance of CSMO if it is not performing well on lower dimensions, then it will not be performing well on higher dimensions too.

Such an insight into the performance of CSMO is important for its further development. CSMO has been developed with least possible modification in the basic structure of SMO without involving any additional parameters other than those of basic SMO. In future efforts, effect of different constraint handling mechanism on the performance of SMO can be investigated to find out most compatible constraint handling mechanism for SMO while solving a particular class of optimization problems.

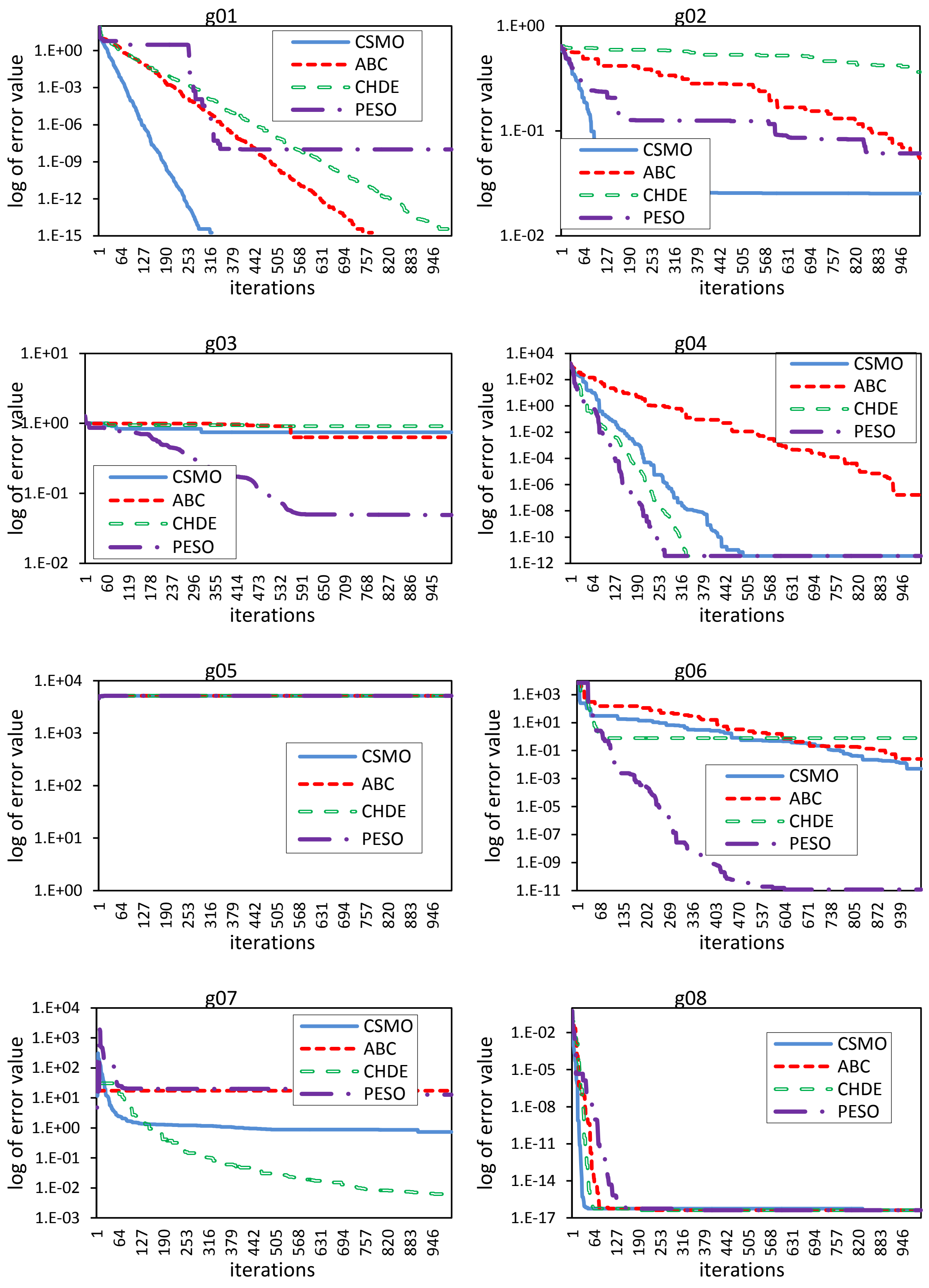

**Figure 3.1**: Convergence graphs of problems g01-g08

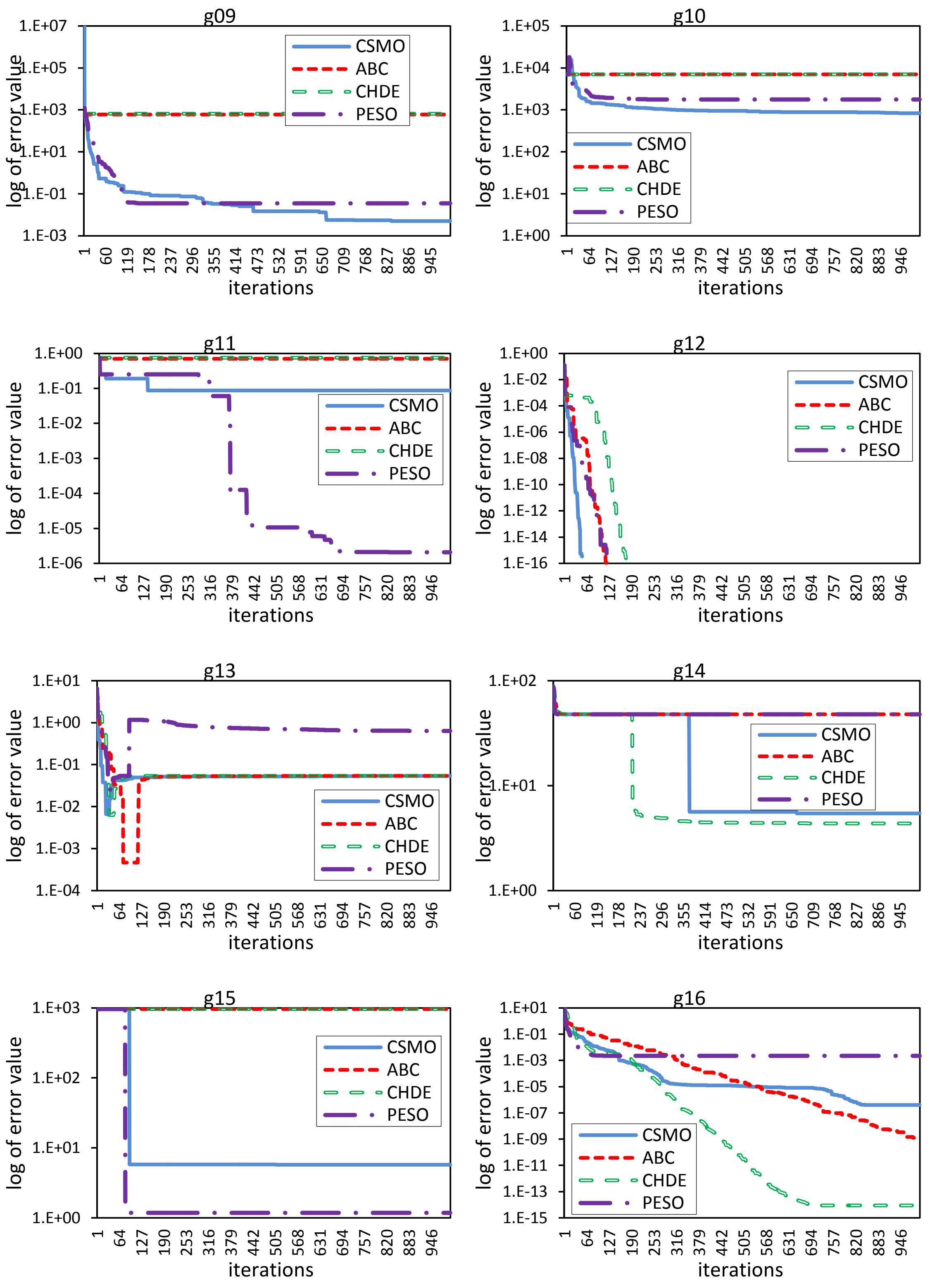

**Figure 3.2**: Convergence graphs of problems g09-g16

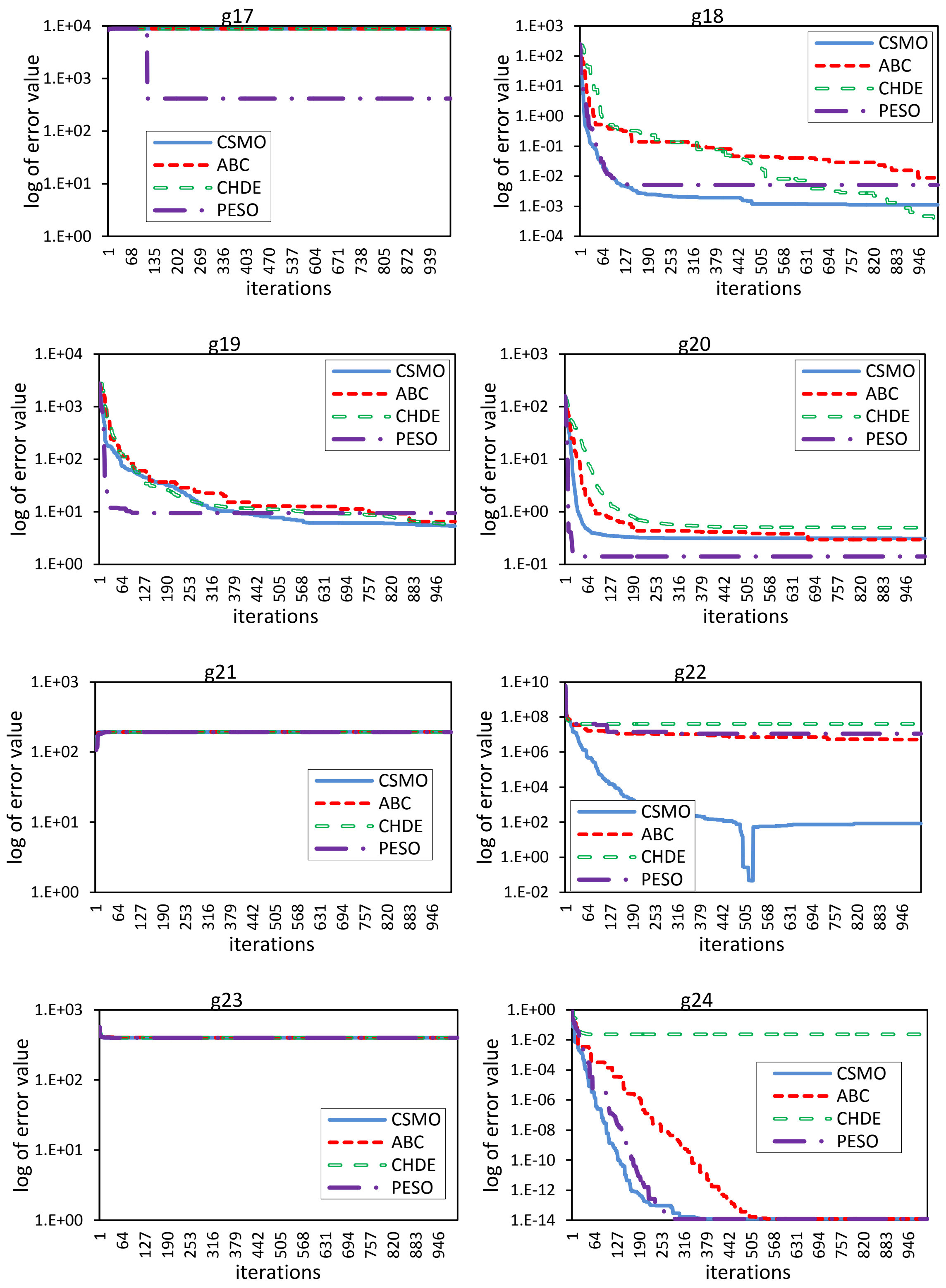

**Figure 3.3**: Convergence graphs of problems g17-g24

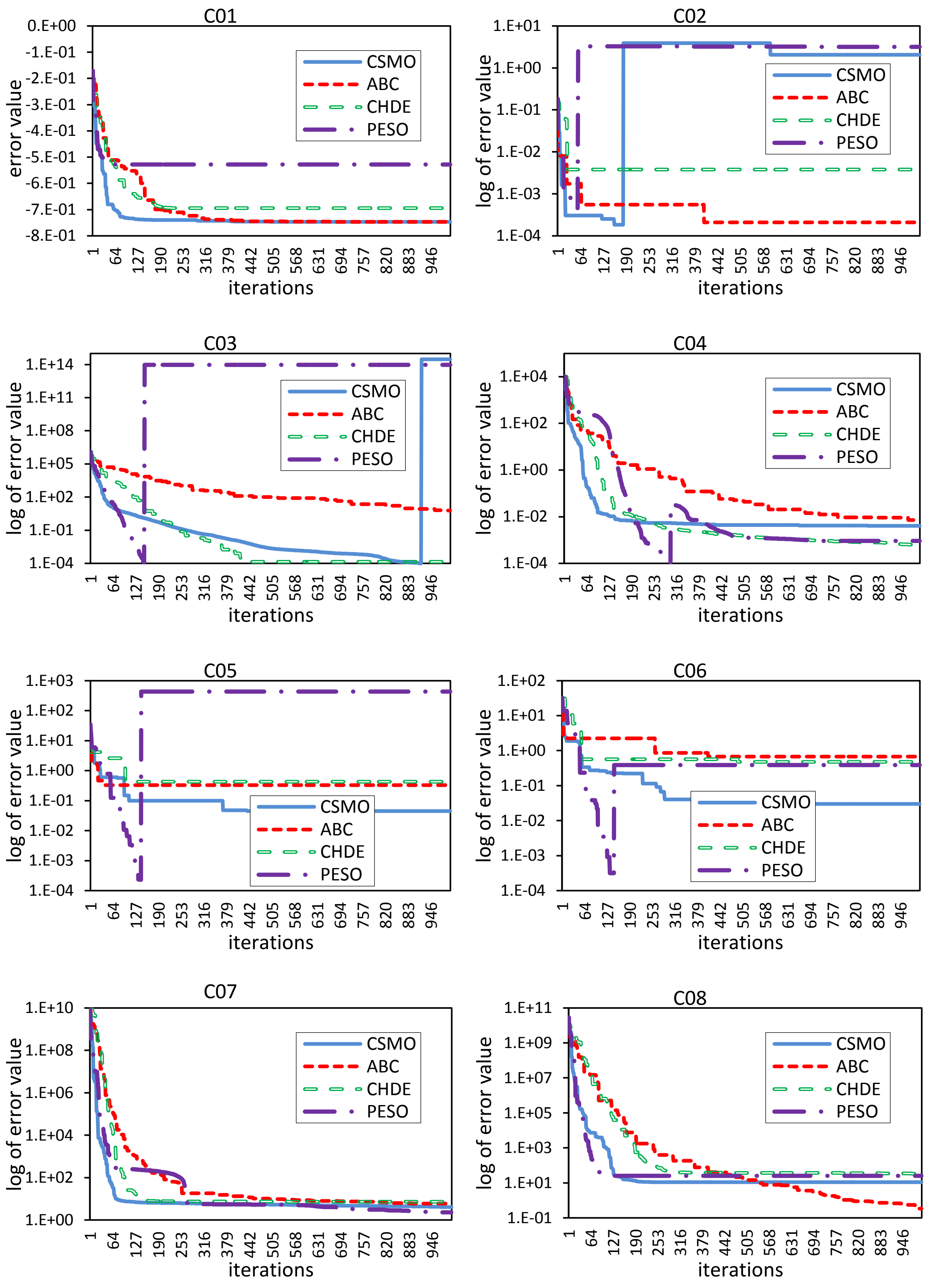

**Figure 3.4**: Convergence graphs of problems C01-C08 (10 dimensions)

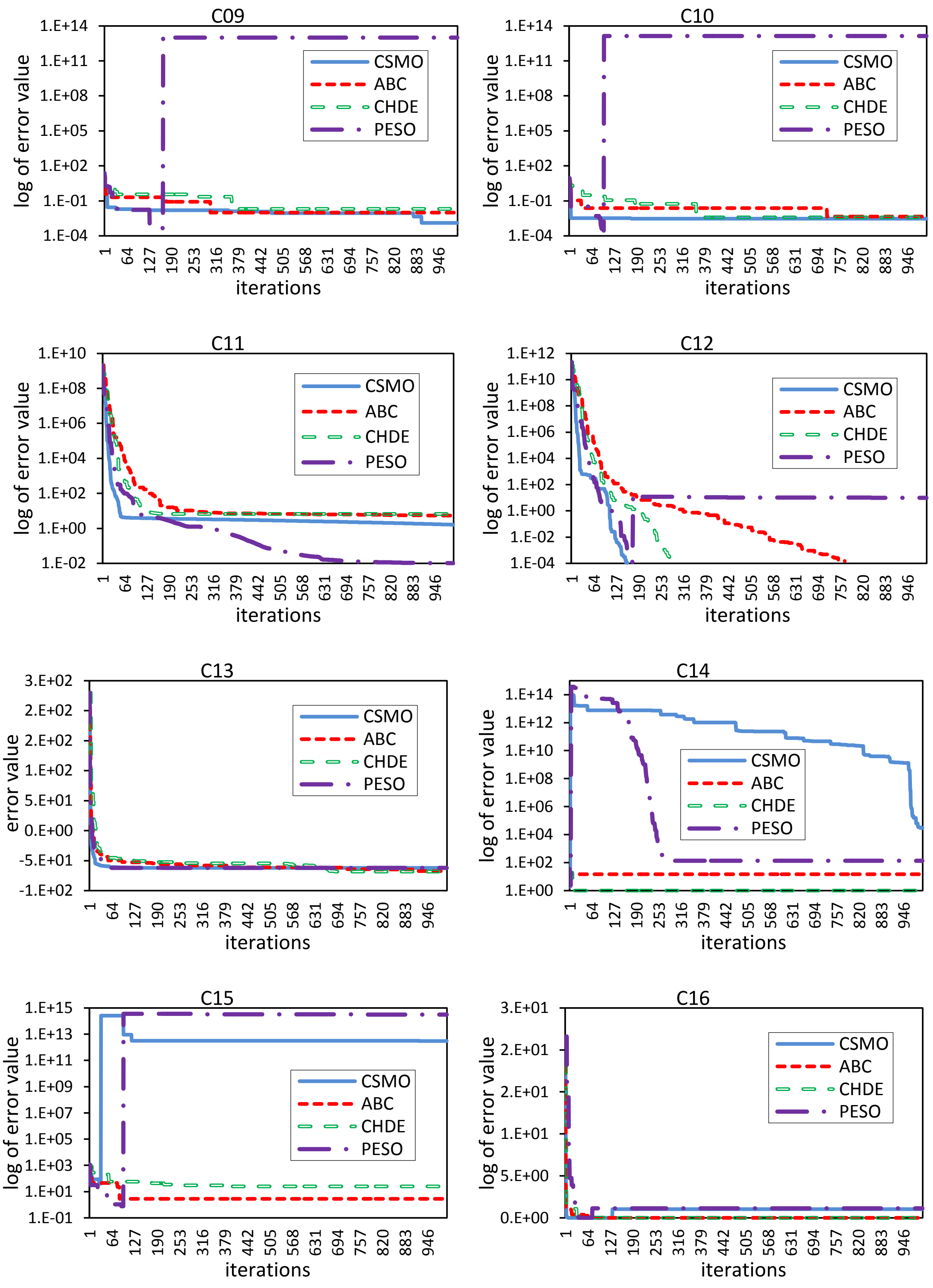

**Figure 3.5**: Convergence graphs of problems C09-C16 (10 dimensions)

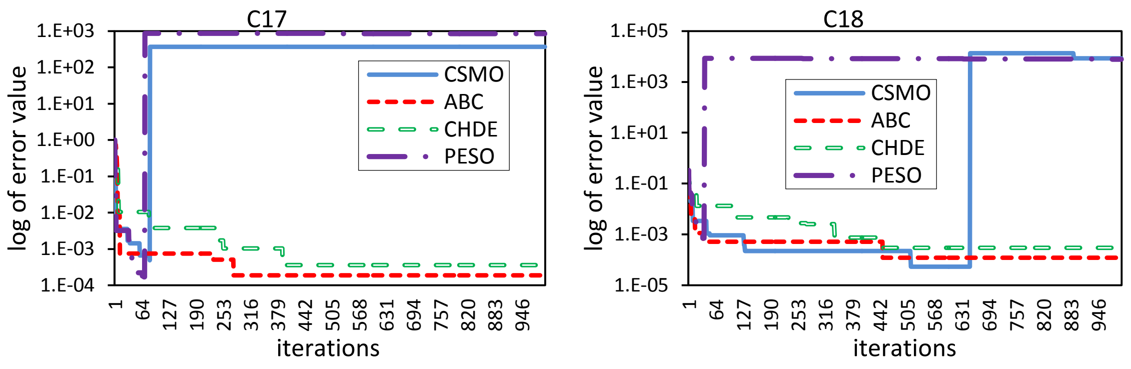

**Figure 3.6**: Convergence graphs of problems C017-C18 (10 dimensions)

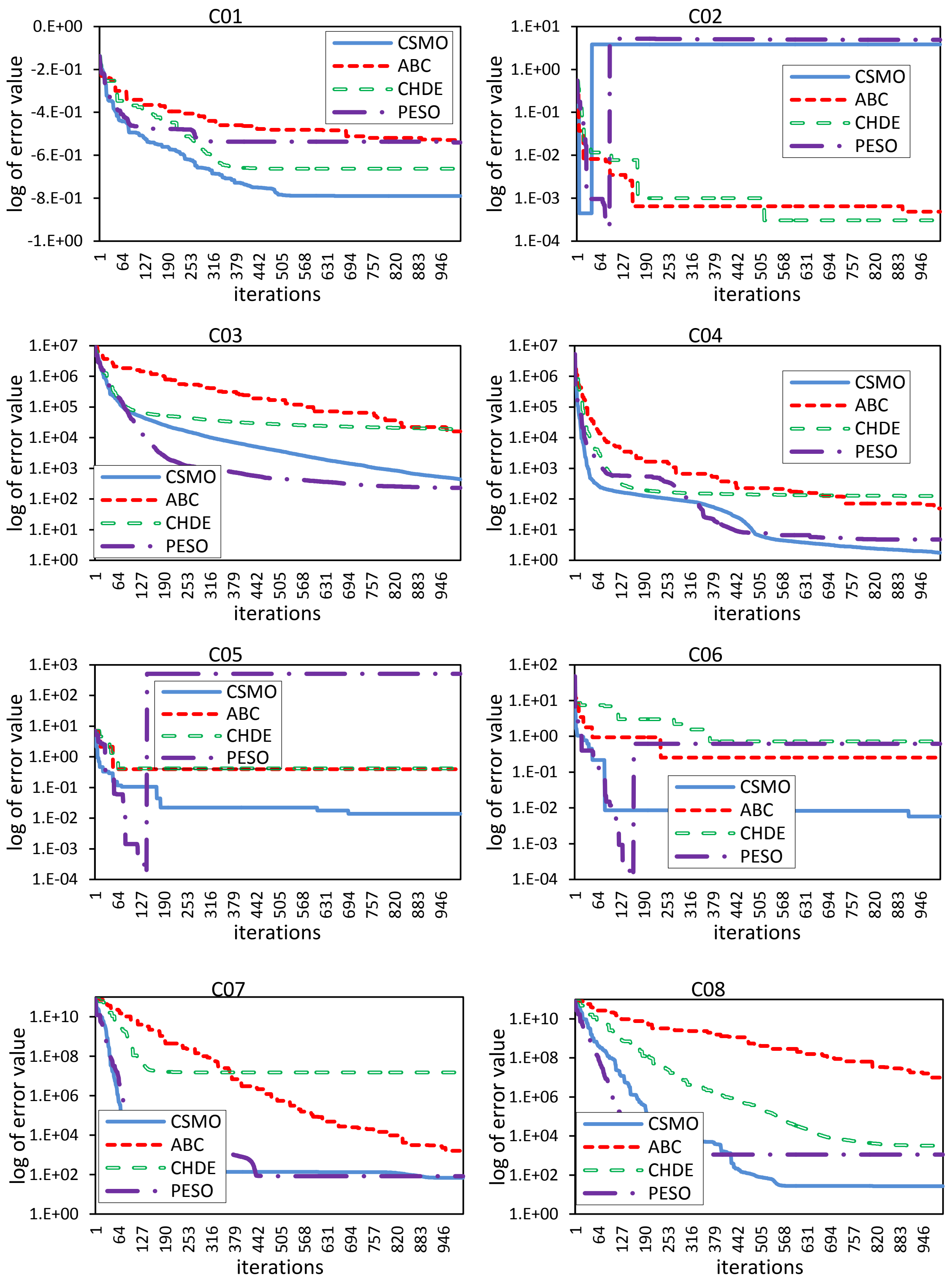

**Figure 3.7**: Convergence graphs of problems C01-C08 (30 dimensions)

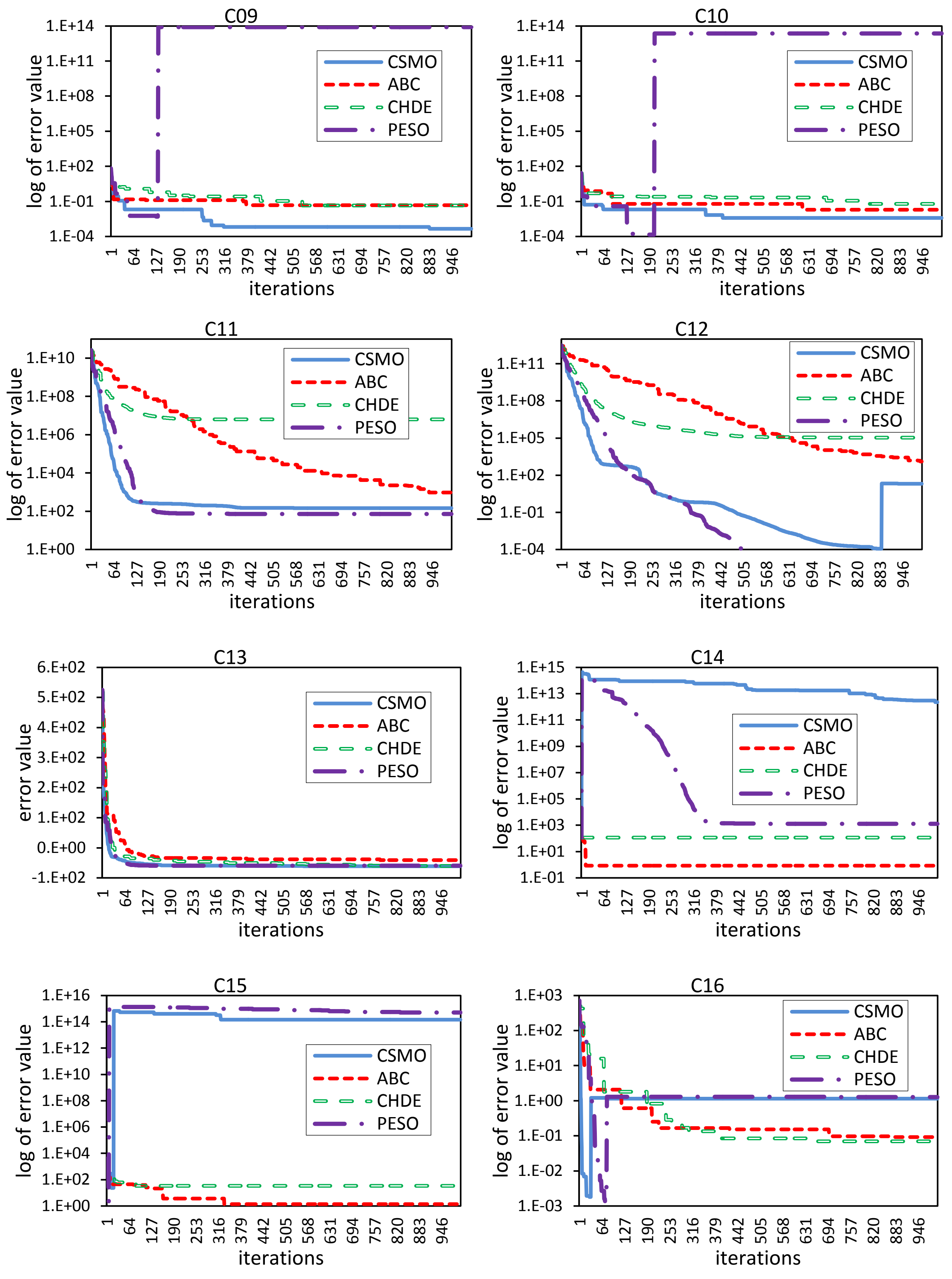

**Figure 3.8**: Convergence plots of problems C09-C16 (30 dimensions)

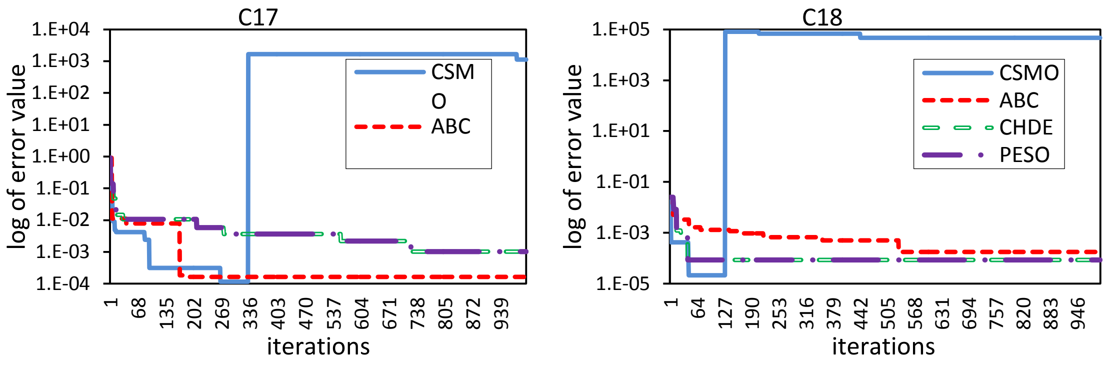

**Figure 3.9**: Convergence plots of problems C17-C18 (30 dimensions)

**Table 3.1:** Classification of problems in CEC2006 benchmark set on the basis of the type of constraints

| Type of Constraints | Problems |
|---|---|
| Only Equalities | g03, g11, g13, g14, g15, g17 |
| Only Inequalities | g01, g02, g04, g06, g07, g08, g09, g10, g12, g16, g18, g19, g24 |
| Both Equalities and Inequalities | g05, g20, g21, g22, g23 |

**Table 3.2:** Classification of problems in CEC2010 benchmark set on the basis of the type of constraints

| Type of Constraints | Problems |
|---|---|
| Only Equalities | C03, C04, C05, C06, C09, C10, C11, |
| Only Inequalities | C01, C07, C08, C13, C14, C15, |
| Both Equalities and Inequalities | C02, C12, C16, C17, C18 |

**Table 3.3:** Parameter setting for CSMO, ABC, CHDE, PESO

| Algorithm | Parameter setting |
|---|---|
| CSMO | Perturbation rate (*Pr*) = linearly increasing ([0.1, 0.4])<br>Maximum number of groups(MG) = 5<br>Local leader limit = 1500<br>Global leader limit = 50 |
| ABC | Modification Rate (MR) = 0.8<br>Maximum Cycle Number (MCN) = (Maximum number of function evaluations)/100<br>Limit = 0.5× ss × D, where ss is the swarm size and D is the dimension of the problem<br>SPP = 0.5× ss × D |
| CHDE | *F* = generated randomly between [0.3, 0.9] per run using a uniform distribution<br>*CR* = generated randomly between [0.8, 1.0] per run using a uniform distribution |
| PESO | c1 = 0.1<br>c2 = 1<br>inertia weight (*w*) = generated randomly between [0.5,1] using uniform distribution |

**Table 3.4:** Feasibility Rate (F.R.) and Best, Median, Worst, Mean and Standard Deviation (Stdev) of the function error values obtained by CSMO with 25 independent runs on CEC2006 Benchmark Problems

| Problems | F.R. | Best | Median | Worst | Mean | Stdev |
|---|---|---|---|---|---|---|
| g01 | 100 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 |
| g02 | 100 | 2.55E-05 | 1.12E-02 | 4.96E-02 | 1.54E-02 | 1.52E-02 |
| g03 | 100 | 2.39E-01 | 4.88E-01 | 8.86E-01 | 4.91E-01 | 1.45E-01 |
| g04 | 100 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 |
| g05 | 80 | 4.79E-02 | 5.49E+02 | 5.30E-03(1) | 3.68E+02 | 3.92E+02 |
| g06 | 100 | 2.79E-08 | 6.93E-07 | 2.40E-04 | 1.43E-05 | 4.80E-05 |
| g07 | 100 | 3.05E-02 | 2.81E-01 | 7.18E-01 | 3.01E-01 | 2.09E-01 |
| g08 | 100 | 4.16E-17 | 4.16E-17 | 4.16E-17 | 4.16E-17 | 6.29E-33 |
| g09 | 100 | 6.75E-04 | 5.10E-03 | 1.07E-02 | 5.27E-03 | 2.99E-03 |
| g10 | 100 | 1.70E+01 | 2.06E+02 | 9.91E+02 | 2.69E+02 | 2.20E+02 |
| g11 | 100 | 5.99E-06 | 1.55E-02 | 2.50E-01 | 8.49E-02 | 1.03E-01 |
| g12 | 100 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 |
| g13 | 88 | 4.91E-01 | 9.09E-01 | 2.14E-04(1) | 8.58E-01 | 1.46E-01 |
| g14 | 100 | 5.10E-01 | 4.09E+00 | 6.09E+00 | 3.90E+00 | 1.43E+00 |
| g15 | 100 | 1.66E-03 | 6.46E-01 | 1.06E+01 | 2.98E+00 | 3.65E+00 |
| g16 | 100 | 6.49E-12 | 6.68E-10 | 1.49E-07 | 9.79E-09 | 3.13E-08 |
| g17 | 96 | 5.49E+01 | 1.59E+02 | 8.49E-05(1) | 2.21E+02 | 1.39E+02 |
| g18 | 100 | 4.34E-05 | 1.94E-04 | 2.42E-03 | 3.47E-04 | 4.74E-04 |
| g19 | 100 | 1.30E+00 | 5.95E+00 | 1.49E+01 | 6.33E+00 | 3.77E+00 |
| g20 | 0 | 3.72E-03(6) | 1.84E-02(11) | 3.04E-02(15) | N.A. | N.A. |
| g21 | 0 | 1.51E-03(2) | 5.44E-03(3) | 1.71E-02(3) | N.A. | N.A. |
| g22 | 0 | 2.31E-01(16) | 5.06E+00(10) | 4.26E+0419) | N.A. | N.A. |
| g23 | 16 | 4.00E+02 | 7.36E-04(4) | 8.32E-03(4) | 4.00E+02 | 0.00E+00 |
| g24 | 100 | 1.24E-14 | 1.24E-14 | 1.24E-14 | 1.24E-14 | 3.22E-30 |

**Table 3.5:** Success Rate (S.R.) and Best, Median, Worst, Mean and Standard Deviation (Stdev) of the number of function evaluations obtained by CSMO with successful runs out of 25 independent runs on CEC2006 Benchmark Problems

| Problems | S.R. | Best | Median | Worst | Mean | Stdev |
|---|---|---|---|---|---|---|
| g01 | 100 | 8950 | 10750 | 28150 | 13714 | 6026 |
| g02 | 8 | 217736 | 255548 | 293361 | 255548 | 53474 |
| g03 | 0 | N.A. | N.A. | N.A. | N.A. | N.A. |
| g04 | 100 | 17550 | 21850 | 30150 | 22690 | 2973 |
| g05 | 0 | N.A. | N.A. | N.A. | N.A. | N.A. |
| g06 | 96 | 165267 | 201324 | 326597 | 216177 | 42756 |
| g07 | 0 | N.A. | N.A. | N.A. | N.A. | N.A. |
| g08 | 100 | 650 | 950 | 1250 | 934 | 146 |
| g09 | 0 | N.A. | N.A. | N.A. | N.A. | N.A. |
| g10 | 0 | N.A. | N.A. | N.A. | N.A. | N.A. |
| g11 | 24 | 274204 | 359310 | 476918 | 374996 | 80233 |
| g12 | 100 | 350 | 1050 | 1650 | 1026 | 274 |
| g13 | 0 | N.A. | N.A. | N.A. | N.A. | N.A. |
| g14 | 0 | N.A. | N.A. | N.A. | N.A. | N.A. |
| g15 | 0 | N.A. | N.A. | N.A. | N.A. | N.A. |
| g16 | 100 | 12850 | 26050 | 80724 | 34930 | 20422 |
| g17 | 0 | N.A. | N.A. | N.A. | N.A. | N.A. |
| g18 | 20 | 219036 | 372280 | 476428 | 348277 | 109348 |
| g19 | 0 | N.A. | N.A. | N.A. | N.A. | N.A. |
| g20 | 0 | N.A. | N.A. | N.A. | N.A. | N.A. |
| g21 | 0 | N.A. | N.A. | N.A. | N.A. | N.A. |
| g22 | 0 | N.A. | N.A. | N.A. | N.A. | N.A. |
| g23 | 0 | N.A. | N.A. | N.A. | N.A. | N.A. |
| g24 | 100 | 2950 | 4250 | 5850 | 4258 | 796 |

**Table 3.6:** Comparison of CSMO against ABC, CHDE and PESO in terms of Feasibility Rate (F.R.) and Success Rate (S.R.) on CEC2006 Benchmark Problems

| Feasibility Rate (F.R.) | | | | | Success Rate (S.R.) | | | |
|---|---|---|---|---|---|---|---|---|
| Problems | CSMO | ABC | CHDE | PESO | CSMO | ABC | CHDE | PESO |
| g01 | **100** | **100** | **100** | **100** | **100** | **100** | 56 | 88 |
| g02 | **100** | **100** | **100** | **100** | 8 | **52** | 12 | 0 |
| g03 | **100** | **100** | **100** | **100** | 0 | 0 | 0 | 0 |
| g04 | **100** | **100** | **100** | **100** | **100** | **100** | 92 | **100** |
| g05 | **80** | 0 | 0 | 72 | 0 | 0 | 0 | 0 |
| g06 | **100** | **100** | 92 | **100** | 96 | **100** | 80 | **100** |
| g07 | **100** | 4 | 12 | **100** | 0 | 0 | 4 | 0 |
| g08 | **100** | **100** | **100** | **100** | **100** | **100** | **100** | **100** |
| g09 | **100** | 40 | 40 | **100** | 0 | 0 | **36** | 0 |
| g10 | **100** | 0 | 0 | **100** | 0 | 0 | 0 | 0 |
| g11 | **100** | 0 | 0 | **100** | 24 | 0 | 0 | **88** |
| g12 | **100** | **100** | **100** | **100** | **100** | **100** | **100** | **100** |
| g13 | 88 | 0 | 0 | **100** | 0 | 0 | 0 | 0 |
| g14 | **100** | 0 | 96 | 80 | 0 | 0 | **40** | 0 |
| g15 | **100** | 0 | 0 | **100** | 0 | 0 | 0 | 0 |
| g16 | **100** | **100** | **100** | **100** | **100** | **100** | 76 | 28 |
| g17 | 96 | 0 | 0 | **100** | 0 | 0 | 0 | 0 |
| g18 | **100** | **100** | 96 | **100** | 20 | 12 | **88** | 12 |
| g19 | **100** | **100** | **100** | **100** | 0 | 0 | **48** | 0 |
| g20 | 0 | 0 | 0 | **72** | 0 | 0 | 0 | **36** |
| g21 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| g22 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| g23 | 16 | 28 | **56** | 24 | 0 | 0 | **16** | 0 |
| g24 | **100** | **100** | **100** | **100** | **100** | **100** | 96 | **100** |

**Table 3.7:** Summary of pairwise comparison of CSMO against ABC, CHDE and PESO in terms of Feasibility Rate (F.R.) and Success Rate (S.R.) on CEC2006 benchmark problems

| CSMO vs. | Criteria | Better | Equal | Worse |
|---|---|---|---|---|
| ABC | Feasibility Rate (F.R.) | 9 | 14 | 1 |
| | Success Rate (S.R.) | 2 | 19 | 3 |
| | Total | 11 | 33 | 4 |
| CHDE | Feasibility Rate (F.R.) | 11 | 12 | 1 |
| | Success Rate (S.R.) | 6 | 11 | 7 |
| | Total | 17 | 23 | 8 |
| PESO | Feasibility Rate (F.R.) | 2 | 18 | 4 |
| | Success Rate (S.R.) | 4 | 17 | 3 |
| | Total | 6 | 35 | 7 |

**Table 3.8:** Comparison of CSMO against ABC, CHDE and PESO in terms of Best of function error values obtained with 25 independent runs on CEC2006 Benchmark Problems

| Problems | CSMO | ABC | CHDE | PESO |
|---|---|---|---|---|
| g01 | **0.00E+00** | **0.00E+00** | **0.00E+00** | **0.00E+00** |
| g02 | 2.55E-05 | 3.00E-05 | **9.23E-08** | 1.27E-02 |
| g03 | 2.39E-01 | 5.14E-01 | 3.86E-01 | **4.85E-02** |
| g04 | **0.00E+00** | **0.00E+00** | **0.00E+00** | **0.00E+00** |
| g05 | **4.79E-02** | 2.02E-05(1) | 2.01E-05(1) | 1.06E-01 |
| g06 | 2.79E-08 | **1.18E-11** | **1.18E-11** | **1.18E-11** |
| g07 | 3.05E-02 | 8.81E-02 | **1.81E-13** | 7.66E-01 |
| g08 | **4.16E-17** | **4.16E-17** | **4.16E-17** | **4.16E-17** |
| g09 | 6.75E-04 | 2.72E-03 | **0.00E+00** | 1.94E-03 |
| g10 | **1.70E+01** | 2.11E-05(1) | 2.14E-04(1) | 5.35E+01 |
| g11 | 5.99E-06 | 3.38E-04(1) | 1.37E-04(1) | **4.99E-09** |
| g12 | **0.00E+00** | **0.00E+00** | **0.00E+00** | **0.00E+00** |
| g13 | 4.91E-01 | 3.34E-05(1) | 3.35E-05(1) | **9.06E-02** |
| g14 | 5.10E-01 | 4.51E-05(1) | **2.13E-14** | 4.16E+00 |
| g15 | 1.66E-03 | 5.00E-05(1) | 5.15E-05(1) | **1.29E-03** |
| g16 | 6.49E-12 | **4.88E-15** | **4.88E-15** | **4.88E-15** |
| g17 | **5.49E+01** | 2.61E-05(1) | 2.55E-05(1) | 9.97E+01 |
| g18 | 4.34E-05 | 3.61E-05 | **3.33E-16** | 7.22E-06 |
| g19 | 1.30E+00 | 1.40E+00 | **2.13E-14** | 4.36E+00 |
| g20 | 3.72E-03(6) | 8.46E-03(7) | 7.19E-03(5) | **3.20E-03** |
| g21 | 1.51E-03(2) | 2.56E-04(2) | **1.71E-05(1)** | 2.59E-04(2) |
| g22 | **2.31E-01(16)** | 3.54E+03(19) | 3.98E+00(6) | 1.99E+01(11) |
| g23 | 4.00E+02 | 4.00E+02 | **0.00E+00** | 3.26E+02 |
| g24 | **1.24E-14** | **1.24E-14** | **1.24E-14** | **1.24E-14** |

**Table 3.9:** Comparison of CSMO against ABC, CHDE and PESO in terms of Median of function error values obtained with 25 independent runs on CEC2006 Benchmark Problems

| Problems | CSMO | ABC | CHDE | PESO |
|---|---|---|---|---|
| g01 | **0.00E+00** | **0.00E+00** | 1.78E-15 | 1.44E-13 |
| g02 | 1.12E-02 | **1.02E-04** | 2.58E-02 | 5.15E-02 |
| g03 | **4.88E-01** | 8.15E-01 | 8.17E-01 | 5.56E-01 |
| g04 | **0.00E+00** | **0.00E+00** | **0.00E+00** | **0.00E+00** |
| g05 | **5.49E+02** | 2.41E-05(1) | 2.89E-05(1) | 9.86E+02 |
| g06 | 6.93E-07 | 2.36E-11 | **1.18E-11** | **1.18E-11** |
| g07 | **2.81E-01** | 2.96E-01(1) | 3.84E-01(1) | 3.57E+00 |
| g08 | **4.16E-17** | **4.16E-17** | **4.16E-17** | **4.16E-17** |
| g09 | **5.10E-03** | 1.64E+00(1) | 2.51E+00(1) | 8.34E-02 |
| g10 | **2.06E+02** | 9.34E-03(1) | 8.90E-03(1) | 1.52E+03 |
| g11 | 1.55E-02 | 1.35E-02(1) | 6.09E-03(1) | **4.82E-06** |
| g12 | **0.00E+00** | **0.00E+00** | **0.00E+00** | **0.00E+00** |
| g13 | 9.09E-01 | 4.26E-05(1) | 4.76E-05(1) | **7.65E-01** |
| g14 | 4.09E+00 | 1.41E-03(3) | **1.46E-03** | 1.03E+01 |
| g15 | **6.46E-01** | 5.14E-05(1) | 7.62E-05(1) | 4.14E+00 |
| g16 | 6.68E-10 | **4.88E-15** | **4.88E-15** | 1.64E-03 |
| g17 | **1.59E+02** | 3.08E-03(3) | 3.21E-05(1) | 1.72E+02 |
| g18 | 1.94E-04 | 7.50E-04 | **1.11E-11** | 2.47E-03 |
| g19 | 5.95E+00 | 1.93E+00 | **2.73E-05** | 8.61E+00 |
| g20 | 1.84E-02(11) | 1.71E-02(15) | 9.14E-03(6) | **3.60E-02** |
| g21 | 5.44E-03(3) | 1.63E-03(2) | **3.25E-05(1)** | 4.07E-02(1) |
| g22 | **5.06E+00(10)** | 6.71E+04(19) | 6.50E+05(19) | 2.04E+05(11) |
| g23 | 7.36E-04(4) | 3.18E-03(1) | **4.00E+02** | 1.11E-01(1) |
| g24 | **1.24E-14** | **1.24E-14** | **1.24E-14** | **1.24E-14** |

**Table 3.10:** Comparison of CSMO against ABC, CHDE and PESO in terms of Worst of function error values obtained with 25 independent runs on CEC2006 Benchmark Problems

| Problems | CSMO | ABC | CHDE | PESO |
|---|---|---|---|---|
| g01 | **0.00E+00** | **0.00E+00** | 6.00E+00 | 3.00E+00 |
| g02 | 4.96E-02 | **1.10E-02** | 3.44E-01 | 1.02E-01 |
| g03 | **8.86E-01** | 9.24E-01 | 9.97E-01 | 1.00E+00 |
| g04 | **0.00E+00** | **0.00E+00** | 1.43E+02 | 3.64E-12 |
| g05 | 5.30E-03(1) | 3.81E-03(1) | **1.40E-04(3)** | 5.40E-03(1) |
| g06 | 2.40E-04 | 4.76E-10 | 1.10E+00(2) | **1.55E-11** |
| g07 | **7.18E-01** | 1.59E+00(2) | 2.73E+00(4) | 1.91E+01 |
| g08 | **4.16E-17** | **4.16E-17** | 5.55E-17 | 5.55E-17 |
| g09 | **1.07E-02** | 4.80E+01(2) | 3.95E+01(2) | 2.64E-01 |
| g10 | **9.91E+02** | 1.04E-01(3) | 1.09E-01(2) | 2.52E+03 |
| g11 | **2.50E-01** | 1.18E-01(1) | 1.18E-01(1) | **2.50E-01** |
| g12 | **0.00E+00** | **0.00E+00** | **0.00E+00** | **0.00E+00** |
| g13 | 2.14E-04(1) | 7.36E-04(1) | 1.30E-04(1) | **5.32E+00** |
| g14 | **6.09E+00** | 2.19E-03(3) | 6.23E-02(1) | 6.67E-05(1) |
| g15 | **1.06E+01** | 8.15E-05(1) | 3.50E-04(2) | **1.06E+01** |
| g16 | 1.49E-07 | **6.88E-15** | 4.42E-01 | 3.34E-03 |
| g17 | 8.49E-05(1) | 2.73E-02(3) | 1.58E-04(2) | **4.59E+02** |
| g18 | **2.42E-03** | 3.25E-03 | 2.05E+00(7) | 2.14E-01 |
| g19 | 1.49E+01 | **2.70E+00** | 4.15E+02 | 3.78E+01 |
| g20 | 3.04E-02(15) | **1.98E-02(20)** | 3.45E-02(7) | 8.30E-02(1) |
| g21 | 1.71E-02(3) | 2.20E-02(2) | **8.18E-03(3)** | 1.22E-01(1) |
| g22 | **4.26E+04(19)** | 1.69E+05(19) | 2.38E+06(19) | 1.07E+06(13) |
| g23 | **8.32E-03(4)** | 7.40E-02(3) | 1.48E+00(1) | 5.83E-01(2) |
| g24 | **1.24E-14** | **1.24E-14** | 2.35E-02 | **1.24E-14** |

**Table 3.11:** Comparison of CSMO against ABC, CHDE and PESO in terms of Mean of function error values obtained with feasible runs out of 25 independent runs on CEC2006 Benchmark Problems

| Problems | CSMO | ABC | CHDE | PESO |
|---|---|---|---|---|
| g01 | **0.00E+00** | **0.00E+00** | 1.34E+00 | 2.80E-01 |
| g02 | 1.54E-02 | **1.62E-03** | 6.32E-02 | 5.13E-02 |
| g03 | **4.91E-01** | 8.01E-01 | 7.69E-01 | 5.49E-01 |
| g04 | **0.00E+00** | **0.00E+00** | 7.26E+00 | 4.37E-13 |
| g05 | **3.68E+02** | N.A. | N.A. | 5.96E+02 |
| g06 | 1.43E-05 | 6.45E-11 | 9.10E+00 | **1.22E-11** |
| g07 | 3.01E-01 | 8.81E-02 | **1.04E-03** | 5.27E+00 |
| g08 | **4.16E-17** | **4.16E-17** | 4.22E-17 | 4.72E-17 |
| g09 | **5.27E-03** | 6.55E-03 | 1.52E+00 | 8.99E-02 |
| g10 | **2.69E+02** | N.A. | N.A. | 1.42E+03 |
| g11 | 8.49E-02 | N.A. | N.A. | **1.00E-02** |
| g12 | **0.00E+00** | **0.00E+00** | **0.00E+00** | **0.00E+00** |
| g13 | 8.58E-01 | N.A. | N.A. | **8.51E-01** |
| g14 | 3.90E+00 | N.A. | **8.09E-01** | 1.01E+01 |
| g15 | **2.98E+00** | N.A. | N.A. | 4.69E+00 |
| g16 | 9.79E-09 | **5.68E-15** | 2.32E-02 | 1.52E-03 |
| g17 | **2.21E+02** | N.A. | N.A. | 2.53E+02 |
| g18 | **3.47E-04** | 8.56E-04 | 8.02E-03 | 1.50E-02 |
| g19 | 6.33E+00 | **1.92E+00** | 2.34E+01 | 1.07E+01 |
| g20 | N.A. | N.A. | N.A. | **4.84E-02** |
| g21 | N.A. | N.A. | N.A. | N.A. |
| g22 | N.A. | N.A. | N.A. | N.A. |
| g23 | 4.00E+02 | 4.00E+02 | **2.30E+02** | 5.76E+02 |
| g24 | **1.24E-14** | **1.24E-14** | 9.40E-04 | **1.24E-14** |

**Table 3.12:** Comparison of CSMO against ABC, CHDE and PESO in terms of Standard Deviation of function error values obtained with feasible runs out of 25 independent runs on CEC2006 Benchmark Problems

| Problems | CSMO | ABC | CHDE | PESO |
|---|---|---|---|---|
| g01 | **0.00E+00** | **0.00E+00** | 1.97E+00 | 7.92E-01 |
| g02 | 1.52E-02 | **3.04E-03** | 8.82E-02 | 2.69E-02 |
| g03 | 1.45E-01 | **1.09E-01** | 2.03E-01 | 3.40E-01 |
| g04 | **0.00E+00** | **0.00E+00** | 2.93E+01 | 1.21E-12 |
| g05 | **3.92E+02** | N.A. | N.A. | 4.39E+02 |
| g06 | 4.80E-05 | 9.73E-11 | 3.92E+01 | **1.23E-12** |
| g07 | 2.09E-01 | N.A. | **1.70E-03** | 4.84E+00 |
| g08 | **6.29E-33** | **6.29E-33** | 2.78E-18 | 6.95E-18 |
| g09 | 2.99E-03 | **2.91E-03** | 4.81E+00 | 7.08E-02 |
| g10 | **2.20E+02** | N.A. | N.A. | 7.28E+02 |
| g11 | 1.03E-01 | N.A. | N.A. | **5.00E-02** |
| g12 | **0.00E+00** | **0.00E+00** | **0.00E+00** | **0.00E+00** |
| g13 | **1.46E-01** | N.A. | N.A. | 9.70E-01 |
| g14 | 1.43E+00 | N.A. | **1.32E+00** | 3.52E+00 |
| g15 | **3.65E+00** | N.A. | N.A. | 3.94E+00 |
| g16 | 3.13E-08 | **1.00E-15** | 8.95E-02 | 1.36E-03 |
| g17 | 1.39E+02 | N.A. | N.A. | **1.20E+02** |
| g18 | **4.74E-04** | 7.57E-04 | 3.90E-02 | 4.25E-02 |
| g19 | 3.77E+00 | **3.03E-01** | 8.63E+01 | 7.21E+00 |
| g20 | N.A. | N.A. | N.A. | **6.34E-02** |
| g21 | N.A. | N.A. | N.A. | N.A. |
| g22 | N.A. | N.A. | N.A. | N.A. |
| g23 | **0.00E+00** | **0.00E+00** | 1.82E+02 | 2.39E+02 |
| g24 | **3.22E-30** | **3.22E-30** | 4.70E-03 | **3.22E-30** |

**Table 3.13:** Summary of pairwise comparison of CSMO against ABC, CHDE and PESO in terms of Best, Median, Worst, Mean and Standard Deviation (Stdev) of function error values on CEC2006 benchmark problems

| CSMO vs. | Criteria | Better | Equal | Worse |
|---|---|---|---|---|
| ABC | Best | 14 | 6 | 4 |
| | Median | 13 | 5 | 6 |
| | Worst | 13 | 5 | 6 |
| | Mean | 3 | 6 | 5 |
| | Stdev | 2 | 6 | 5 |
| | Total | 45 | 28 | 26 |
| CHDE | Best | 9 | 5 | 10 |
| | Median | 12 | 4 | 8 |
| | Worst | 20 | 1 | 3 |
| | Mean | 11 | 1 | 3 |
| | Stdev | 12 | 1 | 2 |
| | Total | 64 | 12 | 26 |
| PESO | Best | 9 | 5 | 10 |
| | Median | 16 | 4 | 4 |
| | Worst | 17 | 4 | 3 |
| | Mean | 16 | 2 | 3 |
| | Stdev | 16 | 2 | 3 |
| | Total | 74 | 17 | 23 |

**Table 3.14:** Comparison of CSMO against ABC, CHDE and PESO in terms of Best of number of function evaluations obtained with successful runs out of 25 independent runs on CEC2006 Benchmark Problems

| Problems | CSMO | ABC | CHDE | PESO |
|---|---|---|---|---|
| g01 | **8950** | 23850 | 9150 | 17300 |
| g02 | 217736 | 316453 | **154950** | N.A. |
| g03 | N.A. | N.A. | N.A. | N.A. |
| g04 | 17550 | 65050 | **7900** | 15950 |
| g05 | N.A. | N.A. | N.A. | N.A. |
| g06 | 165267 | 159380 | **5250** | 29600 |
| g07 | N.A. | N.A. | **70100** | N.A. |
| g08 | 650 | 950 | **250** | 1700 |
| g09 | N.A. | N.A. | **17200** | N.A. |
| g10 | N.A. | N.A. | N.A. | N.A. |
| g11 | 274204 | N.A. | N.A. | **31100** |
| g12 | **350** | 850 | 1600 | 800 |
| g13 | N.A. | N.A. | N.A. | N.A. |
| g14 | N.A. | N.A. | **40300** | N.A. |
| g15 | N.A. | N.A. | N.A. | N.A. |
| g16 | 12850 | 41050 | **8200** | 13850 |
| g17 | N.A. | N.A. | N.A. | N.A. |
| g18 | 219036 | 422568 | **16300** | 21800 |
| g19 | N.A. | N.A. | **64500** | N.A. |
| g20 | N.A. | N.A. | N.A. | **8900** |
| g21 | N.A. | N.A. | N.A. | N.A. |
| g22 | N.A. | N.A. | N.A. | N.A. |
| g23 | N.A. | N.A. | **95850** | N.A. |
| g24 | 2950 | 7151 | **2150** | 6800 |

**Table 3.15:** Comparison of CSMO against ABC, CHDE and PESO in terms Median of number of function evaluations obtained with successful runs out of 25 independent runs on CEC2006 Benchmark Problems

| Problems | CSMO | ABC | CHDE | PESO |
|---|---|---|---|---|
| g01 | **10750** | 24950 | 14100 | 83825 |
| g02 | **255548.5** | 376953 | 290100 | N.A. |
| g03 | N.A. | N.A. | N.A. | N.A. |
| g04 | 21850 | 77150 | **9950** | 17450 |
| g05 | N.A. | N.A. | N.A. | N.A. |
| g06 | 201324 | 175484 | **6500** | 33200 |
| g07 | N.A. | N.A. | **70100** | N.A. |
| g08 | **950** | 1950 | 1100 | 3050 |
| g09 | N.A. | N.A. | **29250** | N.A. |
| g10 | N.A. | N.A. | N.A. | N.A. |
| g11 | 359310 | N.A. | N.A. | **86750** |
| g12 | **1050** | 2450 | 4300 | 2300 |
| g13 | N.A. | N.A. | N.A. | N.A. |
| g14 | N.A. | N.A. | **77775** | N.A. |
| g15 | N.A. | N.A. | N.A. | N.A. |
| g16 | 26050 | 45250 | **12650** | 14600 |
| g17 | N.A. | N.A. | N.A. | N.A. |
| g18 | 372280 | 427268 | 81375 | **23000** |
| g19 | N.A. | N.A. | **165750** | N.A. |
| g20 | N.A. | N.A. | N.A. | **12200** |
| g21 | N.A. | N.A. | N.A. | N.A. |
| g22 | N.A. | N.A. | N.A. | N.A. |
| g23 | N.A. | N.A. | **196775** | N.A. |
| g24 | 4250 | 12151 | **2925** | 9350 |

**Table 3.16:** Comparison of CSMO against ABC, CHDE and PESO in terms Worst of number of function evaluations obtained with successful runs out of 25 independent runs on CEC2006 Benchmark Problems

| Problems | CSMO | ABC | CHDE | PESO |
|---|---|---|---|---|
| g01 | 28150 | 27250 | **20300** | 366800 |
| g02 | **293361** | 441355 | 372900 | N.A. |
| g03 | N.A. | N.A. | N.A. | N.A. |
| g04 | **30150** | 85451 | 35750 | 97100 |
| g05 | N.A. | N.A. | N.A. | N.A. |
| g06 | 326597 | 218293 | **9150** | 53600 |
| g07 | N.A. | N.A. | **70100** | N.A. |
| g08 | **1250** | 2750 | 1650 | 4550 |
| g09 | N.A. | N.A. | **54800** | N.A. |
| g10 | N.A. | N.A. | N.A. | N.A. |
| g11 | 476918 | N.A. | N.A. | **329300** |
| g12 | **1650** | 4950 | 6450 | 3500 |
| g13 | N.A. | N.A. | N.A. | N.A. |
| g14 | N.A. | N.A. | **323750** | N.A. |
| g15 | N.A. | N.A. | N.A. | N.A. |
| g16 | 80724 | 53050 | 43600 | **20150** |
| g17 | N.A. | N.A. | N.A. | N.A. |
| g18 | 476428 | 465770 | 236350 | **23000** |
| g19 | N.A. | N.A. | **437950** | N.A. |
| g20 | N.A. | N.A. | N.A. | **192500** |
| g21 | N.A. | N.A. | N.A. | N.A. |
| g22 | N.A. | N.A. | N.A. | N.A. |
| g23 | N.A. | N.A. | **267900** | N.A. |
| g24 | 5850 | 14650 | **4600** | 10850 |

**Table 3.17:** Comparison of CSMO against ABC, CHDE and PESO in terms Mean of number of function evaluations obtained with successful runs out of 25 independent runs on CEC2006 Benchmark Problems

| Problems | CSMO | ABC | CHDE | PESO |
|---|---|---|---|---|
| g01 | **13714** | 25198 | 14642 | 120840 |
| g02 | **255548** | 379636 | 272650 | N.A. |
| g03 | N.A. | N.A. | N.A. | N.A. |
| g04 | 22690 | 76624 | **11165** | 21038 |
| g05 | N.A. | N.A. | N.A. | N.A. |
| g06 | 216177 | 179669 | **6830** | 33992 |
| g07 | N.A. | N.A. | **70100** | N.A. |
| g08 | **934** | 1886 | 1086 | 3068 |
| g09 | N.A. | N.A. | **30311** | N.A. |
| g10 | N.A. | N.A. | N.A. | N.A. |
| g11 | 374996 | N.A. | N.A. | **115400** |
| g12 | **1026** | 2526 | 4142 | 2264 |
| g13 | N.A. | N.A. | N.A. | N.A. |
| g14 | N.A. | N.A. | **113065** | N.A. |
| g15 | N.A. | N.A. | N.A. | N.A. |
| g16 | 34930 | 45022 | **14402** | 15457 |
| g17 | N.A. | N.A. | N.A. | N.A. |
| g18 | 348277 | 438535 | 94531 | **22600** |
| g19 | N.A. | N.A. | **173146** | N.A. |
| g20 | N.A. | N.A. | N.A. | **37450** |
| g21 | N.A. | N.A. | N.A. | N.A. |
| g22 | N.A. | N.A. | N.A. | N.A. |
| g23 | N.A. | N.A. | **189325** | N.A. |
| g24 | 4258 | 11878 | **3064** | 9116 |

**Table 3.18:** Comparison of CSMO against ABC, CHDE and PESO in terms Standard Deviation of number of function evaluations obtained with successful runs out of 25 independent runs on CEC2006 Benchmark Problems

| Problems | CSMO | ABC | CHDE | PESO |
|---|---|---|---|---|
| g01 | 6026 | **869** | 3308 | 99405 |
| g02 | 53474 | **37772** | 110017 | N.A. |
| g03 | N.A. | N.A. | N.A. | N.A. |
| g04 | **2973** | 4726 | 5623 | 16017 |
| g05 | N.A. | N.A. | N.A. | N.A. |
| g06 | 42756 | 15270 | **1131** | 4401 |
| g07 | N.A. | N.A. | N.A. | N.A. |
| g08 | **146** | 396 | 338 | 813 |
| g09 | N.A. | N.A. | **13775** | N.A. |
| g10 | N.A. | N.A. | N.A. | N.A. |
| g11 | 80233 | N.A. | N.A. | **75240** |
| g12 | **274** | 1092 | 1181 | 673 |
| g13 | N.A. | N.A. | N.A. | N.A. |
| g14 | N.A. | N.A. | **84753** | N.A. |
| g15 | N.A. | N.A. | N.A. | N.A. |
| g16 | 20422 | 2879 | 7981 | **2159** |
| g17 | N.A. | N.A. | N.A. | N.A. |
| g18 | 109348 | 23702 | 75516 | **692** |
| g19 | N.A. | N.A. | **98839** | N.A. |
| g20 | N.A. | N.A. | N.A. | **59787** |
| g21 | N.A. | N.A. | N.A. | N.A. |
| g22 | N.A. | N.A. | N.A. | N.A. |
| g23 | N.A. | N.A. | **70820** | N.A. |
| g24 | 796 | 2062 | **542** | 987 |

**Table 3.19:** Summary of pairwise comparison of CSMO against ABC, CHDE and PESO in terms of Best, Median, Worst, Mean and Standard Deviation (Stdev) of number of function evaluations with successful runs out of 25 independent runs on CEC2006 Benchmark Problems

| CSMO vs. | Criteria | Better | Equal | Worse |
|---|---|---|---|---|
| ABC | Best | 8 | 0 | 1 |
| | Median | 8 | 0 | 1 |
| | Worst | 5 | 0 | 4 |
| | Mean | 8 | 0 | 1 |
| | Stdev | 4 | 0 | 5 |
| | Total | 33 | 0 | 12 |
| CHDE | Best | 2 | 0 | 7 |
| | Median | 4 | 0 | 5 |
| | Worst | 4 | 0 | 5 |
| | Mean | 4 | 0 | 5 |
| | Stdev | 4 | 0 | 5 |
| | Total | 18 | 0 | 27 |
| PESO | Best | 5 | 0 | 4 |
| | Median | 4 | 0 | 5 |
| | Worst | 5 | 0 | 4 |
| | Mean | 4 | 0 | 5 |
| | Stdev | 5 | 0 | 4 |
| | Total | 23 | 0 | 22 |

**Table 3.20:** Wilcoxon Rank sum test based on function error values with a significance level of $\alpha = 0.05$ for CEC2006 Benchmark Problems ('+' indicates CSMO is significantly better, '-' indicates CSMO is significantly worse and '=' indicates there is no significant difference)

| Problems | Pairwise comparison of CSMO versus | | |
|---|---|---|---|
| | ABC | CHDE | PESO |
| g01 | = | + | + |
| g02 | - | + | + |
| g03 | + | + | = |
| g04 | = | + | = |
| g05 | + | + | = |
| g06 | - | + | - |
| g07 | = | - | + |
| g08 | = | = | + |
| g09 | + | + | + |
| g10 | + | + | + |
| g11 | = | = | - |
| g12 | = | = | = |
| g13 | + | + | - |
| g14 | + | - | + |
| g15 | + | + | + |
| g16 | - | + | + |
| g17 | + | + | = |
| g18 | + | + | + |
| g19 | - | + | + |
| g20 | * | * | = |
| g21 | * | * | * |
| g22 | * | * | * |
| g23 | = | = | + |
| g24 | = | = | = |
| * represents that there is no comparison available | | | |

**Table 3.21:** Average execution time per run (in seconds) on CEC2006 Benchmark Problems

| Problems | CSMO | ABC | CHDE | PESO |
|---|---|---|---|---|
| g01 | 0.41768 | 0.56896 | 0.36908 | 0.97888 |
| g02 | 6.40724 | 6.78896 | 6.88476 | 9.9992 |
| g03 | 0.35684 | 0.41096 | 0.33932 | 0.60008 |
| g04 | 0.21896 | 0.24388 | 0.20716 | 0.41756 |
| g05 | 0.4378 | 0.45488 | 0.91476 | 1.0518 |
| g06 | 0.2148 | 0.25324 | 1.021 | 0.84124 |
| g07 | 0.36836 | 0.44588 | 0.32352 | 0.65116 |
| g08 | 1.03548 | 1.3826 | 1.35944 | 1.0588 |
| g09 | 1.30812 | 1.35272 | 1.71036 | 2.15828 |
| g10 | 0.31108 | 0.4072 | 0.26892 | 0.5256 |
| g11 | 0.13656 | 0.14232 | 0.12696 | 0.12892 |
| g12 | 3.90484 | 3.36936 | 3.65544 | 3.49316 |
| g13 | 1.35976 | 1.372 | 1.24464 | 1.4324 |
| g14 | 0.34888 | 0.41824 | 0.44376 | 0.9868 |
| g15 | 1.65564 | 1.72408 | 2.27784 | 1.99696 |
| g16 | 1.88088 | 2.27888 | 2.613 | 2.48844 |
| g17 | 2.35924 | 2.43636 | 3.49396 | 2.92316 |
| g18 | 9.66048 | 9.66428 | 9.76708 | 9.81096 |
| g19 | 4.32468 | 3.62732 | 4.35808 | 4.86152 |
| g20 | 1.12376 | 1.32712 | 0.95596 | 3.83296 |
| g21 | 1.50956 | 1.34956 | 1.13352 | 1.48948 |
| g22 | 2.59196 | 2.42244 | 1.78892 | 2.99968 |
| g23 | 0.34364 | 0.39484 | 0.2788 | 0.62116 |
| g24 | 3.0542 | 3.45708 | 3.6466 | 3.65356 |

**Table 3.22:** Feasibility Rate (F.R.) and Best, Median, Worst, Mean and Standard Deviation (Stdev) of the objective function values obtained by CSMO with 25 independent runs on CEC2010 Benchmark Problems for 10 dimensions

| Problems | F.R. | Best | Median | Worst | Mean | Stdev |
|---|---|---|---|---|---|---|
| C01 | 100 | -7.47E-01 | -7.47E-01 | -7.47E-01 | -7.47E-01 | 0.00E+00 |
| C02 | 100 | 4.96E-01 | 2.10E+00 | 3.55E+00 | 2.22E+00 | 9.03E-01 |
| C03 | 100 | 1.29E+08 | 2.73E+12 | 5.22E+14 | 4.82E+13 | 1.16E+14 |
| C04 | 16 | 1.55E-03 | 9.02E-04(2) | 2.28E+00(2) | 7.00E+00 | 8.08E+00 |
| C05 | 0 | 3.08E-04(2) | 5.92E-03(2) | 1.73E-02(2) | N.A. | N.A. |
| C06 | 0 | 6.02E-04(2) | 9.14E-03(2) | 2.31E-02(2) | N.A. | N.A. |
| C07 | 100 | 6.37E-02 | 1.57E+00 | 7.20E+01 | 9.63E+00 | 1.88E+01 |
| C08 | 100 | 4.12E-05 | 1.06E+01 | 1.02E+03 | 6.27E+01 | 2.03E+02 |
| C09 | 8 | 1.35E+12 | 3.90E-04(1) | 5.79E-03(1) | 1.50E+13 | 1.93E+13 |
| C10 | 8 | 3.15E+11 | 5.29E-04(1) | 2.88E-03(1) | 4.68E+11 | 2.16E+11 |
| C11 | 0 | 7.24E-04(1) | 8.53E-01(1) | 1.89E+01(1) | N.A. | N.A. |
| C12 | 100 | -5.70E+02 | -2.62E+02 | 2.14E+01 | -2.94E+02 | 2.75E+02 |
| C13 | 100 | -6.84E+01 | -6.74E+01 | -6.23E+01 | -6.66E+01 | 2.21E+00 |
| C14 | 100 | 4.95E-03 | 5.43E-01 | 7.42E+00 | 1.34E+00 | 1.95E+00 |
| C15 | 100 | 8.23E+10 | 1.15E+12 | 8.53E+12 | 1.90E+12 | 2.27E+12 |
| C16 | 100 | 8.33E-01 | 1.02E+00 | 1.05E+00 | 9.96E-01 | 5.27E-02 |
| C17 | 100 | 8.19E+01 | 3.04E+02 | 1.01E+03 | 3.83E+02 | 2.33E+02 |
| C18 | 100 | 2.32E+03 | 6.78E+03 | 1.70E+04 | 7.48E+03 | 3.54E+03 |

**Table 3.23:** Comparison of CSMO against ABC, CHDE and PESO in terms Feasibility Rate obtained on CEC2010 Benchmark Problems for 10 dimensions

| Problems | CSMO | ABC | CHDE | PESO |
|---|---|---|---|---|
| C01 | 100 | 100 | 100 | 100 |
| C02 | **100** | 0 | 0 | **100** |
| C03 | **100** | 0 | 44 | 88 |
| C04 | 16 | 0 | **44** | 4 |
| C05 | 0 | 0 | 0 | **100** |
| C06 | 0 | 0 | 0 | **100** |
| C07 | 100 | 100 | 100 | 100 |
| C08 | 100 | 100 | 100 | 100 |
| C09 | 8 | 0 | 0 | **100** |
| C10 | 8 | 0 | 0 | **100** |
| C11 | 0 | 0 | **60** | 12 |
| C12 | **100** | 56 | 76 | 36 |
| C13 | 100 | 100 | 100 | 100 |
| C14 | **100** | 12 | 16 | **100** |
| C15 | **100** | 0 | 0 | **100** |
| C16 | **100** | 0 | 0 | **100** |
| C17 | **100** | 0 | 0 | **100** |
| C18 | **100** | 0 | 0 | **100** |

**Table 3.24:** Summary of pairwise comparison of CSMO against ABC, CHDE and PESO in terms of Feasibility Rate (F.R.) on CEC2010 benchmark problems for 10 dimensions

| CSMO vs. | Criteria | Better | Equal | Worse |
|---|---|---|---|---|
| ABC | Feasibility Rate (F.R.) | 11 | 7 | 0 |
| CHDE | Feasibility Rate (F.R.) | 10 | 6 | 2 |
| PESO | Feasibility Rate (F.R.) | 3 | 10 | 5 |

**Table 3.25:** Comparison of CSMO against ABC, CHDE and PESO in terms of Best of objective function values obtained with 25 independent runs on CEC2010 Benchmark Problems for 10 dimensions

| Problems | CSMO | ABC | CHDE | PESO |
|---|---|---|---|---|
| C01 | **-7.47E-01** | **-7.47E-01** | **-7.47E-01** | -7.37E-01 |
| C02 | 4.96E-01 | 3.40E-05(1) | 3.53E-05(1) | **-1.69E-02** |
| C03 | 1.29E+08 | 1.02E-02(1) | **0.00E+00** | 2.58E+09 |
| C04 | 1.55E-03 | 1.48E-04(2) | **-1.00E-05** | 9.22E-04 |
| C05 | 3.08E-04(2) | 9.80E-03(2) | 5.25E-05(1) | **1.80E+02** |
| C06 | 6.02E-04(2) | 3.55E-02(2) | 5.55E-05(1) | **3.88E-01** |
| C07 | 6.37E-02 | 9.24E-02 | **0.00E+00** | 9.16E-06 |
| C08 | 4.12E-05 | 7.24E-03 | **0.00E+00** | 2.34E-05 |
| C09 | **1.35E+12** | 1.00E-04(1) | 1.82E-04(1) | 2.82E+12 |
| C10 | **3.15E+11** | 1.64E-04(1) | 4.57E-04(1) | 4.45E+12 |
| C11 | 7.24E-04(1) | 1.15E-03(1) | **-1.52E-03** | -1.17E-03 |
| C12 | **-5.70E+02** | -1.99E-01 | -3.05E+02 | -1.99E-01 |
| C13 | **-6.84E+01** | **-6.84E+01** | **-6.84E+01** | -6.56E+01 |
| C14 | 4.95E-03 | 2.70E+11 | 3.50E+00 | **4.57E-07** |
| C15 | **8.23E+10** | 5.40E-03(1) | 3.40E-01(1) | 4.02E+12 |
| C16 | 8.33E-01 | 5.28E-05(2) | 5.15E-05(2) | **7.14E-01** |
| C17 | **8.19E+01** | 3.40E-05(1) | 3.57E-05(1) | 1.77E+02 |
| C18 | **2.32E+03** | 2.07E-08(1) | 3.32E-06(1) | 4.15E+03 |

**Table 3.26:** Comparison of CSMO against ABC, CHDE and PESO in terms of Median of objective function values obtained with 25 independent runs on CEC2010 Benchmark Problems for 10 dimensions

| Problems | CSMO | ABC | CHDE | PESO |
|---|---|---|---|---|
| C01 | **-7.47E-01** | **-7.47E-01** | **-7.47E-01** | -6.69E-01 |
| C02 | **2.10E+00** | 5.10E-05(1) | 1.27E-04(1) | 3.78E+00 |
| C03 | **2.73E+12** | 3.36E-01(1) | 1.01E-04(1) | 8.99E+13 |
| C04 | 9.02E-04(2) | 2.98E-03(4) | **2.60E-05(1)** | 6.77E-02(3) |
| C05 | 5.92E-03(2) | 1.16E-01(2) | 1.62E-01(2) | **5.06E+02** |
| C06 | 9.14E-03(2) | 2.83E-01(2) | 2.28E-01(2) | **3.88E-01** |
| C07 | 1.57E+00 | 3.44E+00 | **2.31E-19** | 1.21E-01 |
| C08 | 1.06E+01 | **2.12E-01** | 1.06E+01 | 4.51E+01 |
| C09 | 3.90E-04(1) | 2.66E-03(1) | 8.22E-03(1) | **1.71E+13** |
| C10 | 5.29E-04(1) | 4.45E-03(1) | 5.48E-03(1) | **1.86E+13** |
| C11 | 8.53E-01(1) | 3.16E+00(1) | **-1.52E-03** | 4.34E-01(1) |
| C12 | **-2.62E+02** | -1.68E-01 | -1.99E-01 | 1.41E+01(1) |
| C13 | -6.74E+01 | -6.82E+01 | **-6.84E+01** | -6.23E+01 |
| C14 | **5.43E-01** | 2.34E+00(1) | 1.59E+00(1) | 3.99E+00 |
| C15 | **1.15E+12** | 1.03E+00(1) | 2.08E+00(1) | 8.29E+13 |
| C16 | **1.02E+00** | 1.38E-04(2) | 2.93E-04(2) | 1.05E+00 |
| C17 | **3.04E+02** | 6.97E-05(1) | 1.19E-04(1) | 5.93E+02 |
| C18 | **6.78E+03** | 1.49E-05(1) | 1.18E-04(2) | 1.38E+04 |

**Table 3.27:** Comparison of CSMO against ABC, CHDE and PESO in terms of Worst of objective function values obtained with 25 independent runs on CEC2010 Benchmark Problems for 10 dimensions

| Problems | CSMO | ABC | CHDE | PESO |
|---|---|---|---|---|
| C01 | **-7.47E-01** | **-7.47E-01** | -3.27E-01 | -5.05E-01 |
| C02 | **3.55E+00** | 1.10E-04(1) | 1.28E-03(1) | 5.35E+00 |
| C03 | **5.22E+14** | 2.86E+01(1) | 1.27E+04(1) | 3.56E+00(1) |
| C04 | 2.28E+00(2) | **2.63E-02(4)** | 9.83E+00(4) | 3.65E+00(2) |
| C05 | 1.73E-02(2) | 2.76E-01(2) | 1.09E+00(2) | **5.93E+02** |
| C06 | 2.31E-02(2) | 7.00E-01(2) | 6.75E-01(2) | **3.88E-01** |
| C07 | 7.20E+01 | **6.07E+00** | 5.39E+04 | 1.54E+02 |
| C08 | 1.02E+03 | **4.16E+01** | 1.79E+06 | 1.60E+04 |
| C09 | 5.79E-03(1) | 1.33E-02(1) | 6.54E-02(1) | **3.03E+13** |
| C10 | 2.88E-03(1) | 1.61E-02(1) | 5.80E-02(1) | **2.85E+13** |
| C11 | 1.89E+01(1) | **6.13E+00(1)** | 1.13E+06(1) | 1.08E+03(1) |
| C12 | **2.14E+01** | 2.71E-02(1) | 3.15E+07(2) | 5.75E+02(2) |
| C13 | -6.23E+01 | **-6.56E+01** | -6.21E+01 | -5.96E+01 |
| C14 | **7.42E+00** | 4.70E+01(2) | 2.54E+01(1) | 5.62E+04 |
| C15 | **8.53E+12** | 1.46E+02(1) | 3.47E+01(1) | 3.61E+14 |
| C16 | **1.05E+00** | 9.75E-04(2) | 7.50E-03(2) | 1.13E+00 |
| C17 | 1.01E+03 | 2.26E-04(1) | 2.99E-03(1) | **4.92E+02** |
| C18 | **1.70E+04** | 7.25E-05(1) | 2.43E-02(1) | 3.81E+04 |

**Table 3.28:** Comparison of CSMO against ABC, CHDE and PESO in terms of Mean of objective function values obtained with feasible runs out of 25 independent runs on CEC2010 Benchmark Problems for 10 dimensions

| Problems | CSMO | ABC | CHDE | PESO |
|---|---|---|---|---|
| C01 | **-7.47E-01** | **-7.47E-01** | -7.15E-01 | -6.41E-01 |
| C02 | **2.22E+00** | N.A. | N.A. | 3.34E+00 |
| C03 | 4.82E+13 | N.A. | **9.64E-07** | 1.90E+14 |
| C04 | 7.00E+00 | N.A. | **-9.51E-06** | 9.22E-04 |
| C05 | N.A. | N.A. | N.A. | **4.82E+02** |
| C06 | N.A. | N.A. | N.A. | **3.88E-01** |
| C07 | 9.63E+00 | **3.34E+00** | 2.23E+03 | 1.84E+01 |
| C08 | 6.27E+01 | **2.53E+00** | 7.16E+04 | 8.36E+02 |
| C09 | **1.50E+13** | N.A. | N.A. | 1.58E+13 |
| C10 | **4.68E+11** | N.A. | N.A. | 1.88E+13 |
| C11 | N.A. | N.A. | **-1.52E-03** | -4.24E-04 |
| C12 | **-2.94E+02** | -1.84E-01 | -4.87E+01 | 1.10E+00 |
| C13 | -6.66E+01 | **-6.79E+01** | -6.62E+01 | -6.09E+01 |
| C14 | **1.34E+00** | 3.82E+11 | 4.95E+00 | 8.13E+03 |
| C15 | **1.90E+12** | N.A. | N.A. | 1.18E+14 |
| C16 | **9.96E-01** | N.A. | N.A. | 1.03E+00 |
| C17 | **3.83E+02** | N.A. | N.A. | 6.43E+02 |
| C18 | **7.48E+03** | N.A. | N.A. | 1.43E+04 |

**Table 3.29:** Comparison of CSMO against ABC, CHDE and PESO in terms of Standard Deviation of objective function value obtained with feasible runs out of 25 independent runs on CEC2010 Benchmark Problems for 10 dimensions

| Problems | CSMO | ABC | CHDE | PESO |
|---|---|---|---|---|
| C01 | **0.00E+00** | **0.00E+00** | 8.44E-02 | 7.46E-02 |
| C02 | **9.03E-01** | N.A. | N.A. | 1.33E+00 |
| C03 | 1.16E+14 | N.A. | **3.20E-06** | 3.74E+14 |
| C04 | 8.08E+00 | N.A. | **1.62E-06** | N.A. |
| C05 | N.A. | N.A. | N.A. | **1.05E+02** |
| C06 | N.A. | N.A. | N.A. | **5.67E-17** |
| C07 | 1.88E+01 | **1.62E+00** | 1.08E+04 | 4.38E+01 |
| C08 | 2.03E+02 | **8.32E+00** | 3.58E+05 | 3.18E+03 |
| C09 | 1.93E+13 | N.A. | N.A. | **8.50E+12** |
| C10 | **2.16E+11** | N.A. | N.A. | 9.42E+12 |
| C11 | N.A. | N.A. | **2.24E-19** | 8.29E-04 |
| C12 | 2.75E+02 | **1.80E-02** | 7.84E+01 | 3.27E+00 |
| C13 | 2.21E+00 | **8.11E-01** | 2.63E+00 | 2.77E+00 |
| C14 | 1.95E+00 | 1.06E+11 | **1.30E+00** | 1.84E+04 |
| C15 | **2.27E+12** | N.A. | N.A. | 1.02E+14 |
| C16 | **5.27E-02** | N.A. | N.A. | 7.22E-02 |
| C17 | **2.33E+02** | N.A. | N.A. | 2.62E+02 |
| C18 | **3.54E+03** | N.A. | N.A. | 7.79E+03 |

**Table 3.30:** Summary of pairwise comparison of CSMO against ABC, CHDE and PESO in terms of Best, Median, Worst, Mean and Standard Deviation (Stdev) of objective function values on CEC2010 benchmark problems for 10 dimensions

| CSMO vs. | Criteria | Better | Equal | Worse |
|---|---|---|---|---|
| ABC | Best | 16 | 2 | 0 |
| | Median | 15 | 1 | 2 |
| | Worst | 12 | 1 | 5 |
| | Mean | 2 | 1 | 3 |
| | Stdev | 1 | 1 | 4 |
| | Total | 46 | 6 | 14 |
| CHDE | Best | 9 | 2 | 7 |
| | Median | 12 | 2 | 4 |
| | Worst | 18 | 0 | 0 |
| | Mean | 6 | 0 | 2 |
| | Stdev | 4 | 0 | 4 |
| | Total | 49 | 4 | 17 |
| PESO | Best | 9 | 0 | 9 |
| | Median | 12 | 0 | 6 |
| | Worst | 13 | 0 | 5 |
| | Mean | 14 | 0 | 1 |
| | Stdev | 12 | 0 | 2 |
| | Total | 60 | 0 | 23 |

**Table 3.31:** Wilcoxon Rank sum test based on objective function value with a significance level of $\alpha = 0.05$ for CEC2010 Benchmark Problems for 10 dimensions ('+' indicates CSMO is significantly better, '-' indicates CSMO is significantly worse and '=' indicates there is no significant difference)

| | Pairwise comparison of CSMO versus | | |
|---|---|---|---|
| Problems | ABC | CHDE | PESO |
| C01 | = | + | + |
| C02 | + | + | + |
| C03 | + | - | = |
| C04 | = | - | = |
| C05 | * | * | - |
| C06 | * | * | - |
| C07 | - | + | + |
| C08 | - | = | + |
| C09 | + | + | + |
| C10 | + | + | + |
| C11 | = | - | = |
| C12 | + | + | + |
| C13 | = | = | + |
| C14 | + | + | + |
| C15 | + | + | + |
| C16 | + | + | + |
| C17 | + | + | + |
| C18 | + | + | + |
| * represents that there is no comparison available | | | |

**Table 3.32**: Average execution time per run (in seconds) on CEC2010 Benchmark Problems for 10 dimensions

| Problems | CSMO | ABC | CHDE | PESO |
|---|---|---|---|---|
| C01 | 1.57704 | 1.55108 | 1.7424 | 1.90884 |
| C02 | 0.62848 | 0.68408 | 0.62172 | 0.73736 |
| C03 | 2.99672 | 2.27116 | 3.56624 | 3.48196 |
| C04 | 2.26852 | 2.32996 | 2.27464 | 2.35688 |
| C05 | 0.66564 | 0.7434 | 0.67532 | 0.76536 |
| C06 | 0.9182 | 1.03692 | 0.96604 | 1.033 |
| C07 | 4.56004 | 4.54504 | 3.87796 | 4.6046 |
| C08 | 3.45228 | 3.12884 | 3.44736 | 3.27608 |
| C09 | 0.39936 | 0.45472 | 0.3972 | 0.58512 |
| C10 | 0.6866 | 0.77276 | 0.78496 | 0.87532 |
| C11 | 4.61672 | 4.67012 | 4.6762 | 4.74156 |
| C12 | 2.44728 | 2.4662 | 2.47104 | 2.53948 |
| C13 | 0.69888 | 0.71472 | 0.73828 | 0.82768 |
| C14 | 2.34368 | 1.02048 | 2.62536 | 3.36856 |
| C15 | 1.22948 | 1.3252 | 3.11072 | 1.36908 |
| C16 | 0.8308 | 0.89376 | 0.894 | 0.95584 |
| C17 | 0.41632 | 0.45752 | 0.55272 | 0.57392 |
| C18 | 0.67808 | 0.75412 | 0.983 | 0.81752 |

**Table 3.33:** Feasibility Rate (F.R.) and Best, Median, Worst, Mean and Standard Deviation (Stdev) of the objective function values obtained by CSMO with 25 independent runs on CEC2010 Benchmark Problems for 30 dimensions

| Problems | F.R. | Best | Median | Worst | Mean | Stdev |
|---|---|---|---|---|---|---|
| C01 | 100 | -8.18E-01 | -8.18E-01 | -8.04E-01 | -8.17E-01 | 2.80E-03 |
| C02 | 100 | 1.38E+00 | 3.03E+00 | 4.05E+00 | 2.97E+00 | 6.29E-01 |
| C03 | 0 | 1.12E-03(1) | 2.61E+01(1) | 4.53E+02(1) | N.A. | N.A. |
| C04 | 0 | 4.14E-03(3) | 4.27E-02(3) | 1.22E+00(4) | N.A. | N.A. |
| C05 | 0 | 2.32E-04(2) | 9.78E-04(2) | 2.22E-03(2) | N.A. | N.A. |
| C06 | 0 | 3.52E-04(2) | 2.04E-03(2) | 4.22E-03(2) | N.A. | N.A. |
| C07 | 100 | 4.72E-04 | 1.37E+01 | 1.03E+02 | 2.60E+01 | 3.50E+01 |
| C08 | 100 | 2.11E-03 | 7.99E+01 | 1.51E+04 | 1.09E+03 | 3.23E+03 |
| C09 | 16 | 1.78E+13 | 3.36E-04(1) | 1.26E-03(1) | 3.36E+13 | 2.15E+13 |
| C10 | 12 | 1.05E+12 | 3.06E-04(1) | 1.53E-03(1) | 2.59E+13 | 2.33E+13 |
| C11 | 4 | 2.61E-04 | 6.91E+00(1) | 1.38E+02(1) | 2.61E-04 | - |
| C12 | 92 | -8.85E+02 | -1.99E-01 | 3.23E-01(1) | -4.19E+02 | 4.20E+02 |
| C13 | 100 | -6.75E+01 | -6.40E+01 | -6.19E+01 | -6.43E+01 | 1.27E+00 |
| C14 | 100 | 1.30E-02 | 1.28E+01 | 2.87E+03 | 1.85E+02 | 5.85E+02 |
| C15 | 100 | 7.57E+12 | 2.60E+13 | 2.22E+13 | 2.32E+13 | 1.10E+13 |
| C16 | 100 | 1.06E+00 | 1.11E+00 | 1.17E+00 | 1.11E+00 | 2.48E-02 |
| C17 | 100 | 8.09E+02 | 1.36E+03 | 2.44E+03 | 1.43E+03 | 3.33E+02 |
| C18 | 100 | 1.80E+04 | 2.63E+04 | 3.52E+04 | 2.65E+04 | 4.57E+03 |

**Table 3.34:** Comparison of CSMO against ABC, CHDE and PESO in terms of Feasibility Rate obtained on CEC2010 Benchmark Problems for 30 dimensions

| Problems | CSMO | ABC | CHDE | PESO |
|---|---|---|---|---|
| C01 | 100 | 100 | 100 | 100 |
| C02 | **100** | 0 | 0 | **100** |
| C03 | 0 | 0 | 0 | 0 |
| C04 | 0 | 0 | 0 | 0 |
| C05 | 0 | 0 | 0 | **100** |
| C06 | 0 | 0 | 0 | **100** |
| C07 | 100 | 100 | 100 | 100 |
| C08 | 100 | 100 | 100 | 100 |
| C09 | 16 | 0 | 0 | **100** |
| C10 | 12 | 0 | 0 | **100** |
| C11 | 4 | 0 | **24** | 0 |
| C12 | **92** | 48 | 36 | 32 |
| C13 | 100 | 100 | 100 | 100 |
| C14 | **100** | 0 | 24 | **100** |
| C15 | **100** | 0 | 0 | **100** |
| C16 | **100** | 0 | 0 | **100** |
| C17 | **100** | 0 | 0 | **100** |
| C18 | **100** | 0 | 0 | **100** |

**Table 3.35:** Summary of pairwise comparison of CSMO against ABC, CHDE and PESO in terms of Feasibility Rate (F.R.) on CEC2010 benchmark problems for 30 dimensions

| CSMO vs. | Criteria | Better | Equal | Worse |
|---|---|---|---|---|
| ABC | Feasibility Rate (F.R.) | 10 | 8 | 0 |
| CHDE | Feasibility Rate (F.R.) | 9 | 8 | 1 |
| PESO | Feasibility Rate (F.R.) | 2 | 12 | 4 |

**Table 3.36:** Comparison of CSMO against ABC, CHDE and PESO in terms of Best of objective function values obtained with 25 independent runs on CEC2010 Benchmark Problems for 30 dimensions

| Problems | CSMO | ABC | CHDE | PESO |
|---|---|---|---|---|
| C01 | -8.18E-01 | **-8.22E-01** | -8.18E-01 | -7.23E-01 |
| C02 | **1.38E+00** | 3.34E-05(1) | 3.63E-05(1) | 2.85E+00 |
| C03 | 1.12E-03(1) | 1.74E+01(1) | **1.00E-04(1)** | 3.38E+01(1) |
| C04 | 4.14E-03(3) | 5.04E-02(4) | **2.50E-05(1)** | 5.41E-02(3) |
| C05 | 2.32E-04(2) | 1.19E-03(2) | 5.45E-05(1) | **3.12E+02** |
| C06 | 3.52E-04(2) | 2.52E-02(2) | 7.95E-05(1) | **6.16E-01** |
| C07 | 4.72E-04 | 1.61E+01 | **0.00E+00** | 1.02E+01 |
| C08 | 2.11E-03 | 2.23E+01 | **0.00E+00** | 1.15E+01 |
| C09 | 1.78E+13 | 1.43E-04(1) | 2.20E-04(1) | **1.67E+13** |
| C10 | **1.05E+12** | 1.06E-04(1) | 4.97E-04(1) | 1.66E+13 |
| C11 | 2.61E-04 | 2.08E+01(1) | **-3.92E-04** | 1.37E+01(1) |
| C12 | **-8.85E+02** | -1.98E-01 | -1.99E-01 | -1.96E-01 |
| C13 | -6.75E+01 | -5.55E+01 | **-6.76E+01** | -6.24E+01 |
| C14 | 1.30E-02 | 9.65E-03(1) | **1.49E-12** | 1.72E+01 |
| C15 | **7.57E+12** | 1.05E-02(1) | 2.59E-01(1) | 4.53E+13 |
| C16 | **1.06E+00** | 5.30E-05(2) | 5.45E-05(2) | 1.08E+00 |
| C17 | 8.09E+02 | 3.45E-05(1) | 3.50E-05(1) | **8.05E+02** |
| C18 | 1.80E+04 | 1.30E-07(1) | 5.70E-07(1) | **1.59E+04** |

**Table 3.37:** Comparison of CSMO against ABC, CHDE and PESO in terms of Median of objective function values obtained with 25 independent runs on CEC2010 Benchmark Problems for 30 dimensions

| Problems | CSMO | ABC | CHDE | PESO |
|---|---|---|---|---|
| C01 | -8.18E-01 | **-8.20E-01** | -7.30E-01 | -6.51E-01 |
| C02 | 3.03E+00 | 4.23E-05(1) | 7.37E-05(1) | 4.15E+00 |
| C03 | 2.61E+01(1) | 4.36E+02(1) | **4.13E+00(1)** | 1.20E+03(1) |
| C04 | **4.27E-02(3)** | 4.39E-01(4) | 4.55E-02(2) | 1.70E+00(3) |
| C05 | 9.78E-04(2) | 4.97E-02(2) | 4.49E-02(2) | **5.67E+02** |
| C06 | 2.04E-03(2) | 8.23E-02(2) | 9.00E-02(2) | **6.16E-01** |
| C07 | 1.37E+01 | 2.35E+01 | **1.33E+01** | 2.57E+01 |
| C08 | 7.99E+01 | **2.49E+01** | 7.29E+01 | 3.04E+02 |
| C09 | 3.36E-04(1) | 1.12E-03(1) | 2.41E-03(1) | **5.65E+13** |
| C10 | 3.06E-04(1) | 1.55E-03(1) | 4.32E-03(1) | **4.78E+13** |
| C11 | 6.91E+00(1) | 2.24E+01(1) | **6.79E+00(1)** | 2.83E+01(1) |
| C12 | **-1.99E-01** | 5.01E-05(1) | 3.19E-01(1) | 4.91E-02(1) |
| C13 | **-6.40E+01** | -5.03E+01 | -6.33E+01 | -5.96E+01 |
| C14 | **1.28E+01** | 1.13E-01(1) | 7.40E+00(1) | 1.17E+04 |
| C15 | **2.60E+13** | 1.07E-01(1) | 1.64E+01(1) | 1.90E+14 |
| C16 | **1.11E+00** | 4.46E-04(2) | 1.45E-04(2) | 1.17E+00 |
| C17 | **1.36E+03** | 4.86E-05(1) | 5.90E-05(1) | 2.25E+03 |
| C18 | **2.63E+04** | 6.56E-06(1) | 5.10E-05(1) | 3.94E+04 |

**Table 3.38:** Comparison of CSMO against ABC, CHDE and PESO in terms of Worst of objective function value obtained with 25 independent runs on CEC2010 Benchmark Problems for 30 dimensions

| Problems | CSMO | ABC | CHDE | PESO |
|---|---|---|---|---|
| C01 | -8.04E-01 | **-8.16E-01** | -4.25E-01 | -5.42E-01 |
| C02 | **4.05E+00** | 8.51E-05(1) | 4.83E-03(1) | 5.06E+00 |
| C03 | **4.53E+02(1)** | 1.89E+03(1) | 5.94E+04(1) | 1.64E+04(1) |
| C04 | **1.22E+00(4)** | 3.10E+00(4) | 3.13E+02(4) | 1.36E+01(3) |
| C05 | 2.22E-03(2) | 9.35E-02(2) | 2.46E-01(2) | **5.93E+02** |
| C06 | 4.22E-03(2) | 1.94E-01(2) | 3.32E-01(2) | **6.16E-01** |
| C07 | 1.03E+02 | **4.83E+01** | 5.58E+08 | 3.00E+02 |
| C08 | 1.51E+04 | **1.83E+02** | 3.63E+08 | 1.10E+04 |
| C09 | 1.26E-03(1) | 7.30E-03(1) | 1.17E-01(1) | **1.10E+14** |
| C10 | 1.53E-03(1) | 9.74E-03(1) | 3.16E-01(1) | **1.30E+14** |
| C11 | 1.38E+02(1) | **2.67E+01(1)** | 8.02E+07(1) | 1.14E+03(1) |
| C12 | **3.23E-01(1)** | 5.70E-01(1) | 6.95E+09(1) | 1.52E+01(1) |
| C13 | -6.19E+01 | -4.53E+01 | -5.65E+01 | **-6.34E+01** |
| C14 | **2.87E+03** | 9.89E-01(1) | 7.37E+01(1) | 2.75E+05 |
| C15 | **2.22E+13** | 6.67E-01(1) | 1.73E+02(1) | 5.62E+14 |
| C16 | **1.17E+00** | 4.80E-03(2) | 7.95E-03(2) | 1.28E+00 |
| C17 | **2.44E+03** | 9.15E-05(1) | 4.00E-04(1) | 4.58E+03 |
| C18 | **3.52E+04** | 2.46E-05(1) | 1.48E-03(2) | 1.07E+05 |

**Table 3.39:** Comparison of CSMO against ABC, CHDE and PESO in terms of Mean of objective function values obtained with feasible runs out of 25 independent runs on CEC2010 Benchmark Problems for 30 dimensions

| Problems | CSMO | ABC | CHDE | PESO |
|---|---|---|---|---|
| C01 | -8.17E-01 | **-8.19E-01** | -7.01E-01 | -6.51E-01 |
| C02 | **2.97E+00** | N.A. | N.A. | 4.09E+00 |
| C03 | N.A. | N.A. | N.A. | N.A. |
| C04 | N.A. | N.A. | N.A. | N.A. |
| C05 | N.A. | N.A. | N.A. | **5.36E+02** |
| C06 | N.A. | N.A. | N.A. | **6.16E-01** |
| C07 | 2.60E+01 | **2.43E+01** | 3.26E+07 | 5.74E+01 |
| C08 | 1.09E+03 | **4.13E+01** | 2.82E+07 | 1.57E+03 |
| C09 | **3.36E+13** | N.A. | N.A. | 5.72E+13 |
| C10 | **2.59E+13** | N.A. | N.A. | 5.11E+13 |
| C11 | 2.61E-04 | N.A. | **-3.85E-04** | N.A. |
| C12 | **-4.19E+02** | -1.64E-01 | -1.88E-01 | 2.46E+00 |
| C13 | **-6.43E+01** | -5.05E+01 | -6.31E+01 | -5.95E+01 |
| C14 | **1.85E+02** | N.A. | 1.07E+13 | 4.03E+04 |
| C15 | **2.32E+13** | N.A. | N.A. | 2.41E+14 |
| C16 | **1.11E+00** | N.A. | N.A. | 1.17E+00 |
| C17 | **1.43E+03** | N.A. | N.A. | 2.35E+03 |
| C18 | **2.65E+04** | N.A. | N.A. | 4.27E+04 |

**Table 3.40:** Comparison of CSMO against ABC, CHDE and PESO in terms of Standard Deviation of objective function values obtained with feasible runs out of 25 independent runs on CEC2010 Benchmark Problems for 30 dimensions

| Problems | CSMO | ABC | CHDE | PESO |
|---|---|---|---|---|
| C01 | 2.80E-03 | **1.80E-03** | 1.12E-01 | 4.84E-02 |
| C02 | **6.29E-01** | N.A. | N.A. | 6.56E-01 |
| C03 | N.A. | N.A. | N.A. | N.A. |
| C04 | N.A. | N.A. | N.A. | N.A. |
| C05 | N.A. | N.A. | N.A. | **8.15E+01** |
| C06 | N.A. | N.A. | N.A. | **2.27E-16** |
| C07 | 3.50E+01 | **5.46E+00** | 1.14E+08 | 6.96E+01 |
| C08 | 3.23E+03 | **4.07E+01** | 7.95E+07 | 3.06E+03 |
| C09 | **2.15E+13** | N.A. | N.A. | 2.64E+13 |
| C10 | **2.33E+13** | N.A. | N.A. | 2.52E+13 |
| C11 | N.A. | N.A. | **1.13E-05** | N.A. |
| C12 | 4.20E+02 | 4.27E-02 | **3.34E-02** | 5.05E+00 |
| C13 | **1.27E+00** | 2.76E+00 | 2.66E+00 | 1.89E+00 |
| C14 | **5.85E+02** | N.A. | 2.55E+13 | 6.67E+04 |
| C15 | **1.10E+13** | N.A. | N.A. | 1.45E+14 |
| C16 | **2.48E-02** | N.A. | N.A. | 4.08E-02 |
| C17 | **3.33E+02** | N.A. | N.A. | 1.03E+03 |
| C18 | **4.57E+03** | N.A. | N.A. | 1.98E+04 |

**Table 3.41:** Summary of pairwise comparison of CSMO against ABC, CHDE and PESO in terms of Best, Median, Worst, Mean and Standard Deviation (Stdev) of objective function values on CEC2010 benchmark problems for 30 dimensions

| CSMO vs. | Criteria | Better | Equal | Worse |
|---|---|---|---|---|
| ABC | Best | 17 | 0 | 1 |
| | Median | 16 | 0 | 2 |
| | Worst | 14 | 0 | 4 |
| | Mean | 2 | 0 | 3 |
| | Stdev | 1 | 0 | 4 |
| | Total | 50 | 0 | 14 |
| CHDE | Best | 8 | 1 | 9 |
| | Median | 14 | 0 | 4 |
| | Worst | 18 | 0 | 0 |
| | Mean | 6 | 0 | 1 |
| | Stdev | 5 | 0 | 1 |
| | Total | 51 | 1 | 15 |
| PESO | Best | 13 | 0 | 5 |
| | Median | 14 | 0 | 4 |
| | Worst | 12 | 0 | 6 |
| | Mean | 13 | 0 | 0 |
| | Stdev | 11 | 0 | 2 |
| | Total | 63 | 0 | 17 |

**Table 3.42:** Wilcoxon Rank sum test based on objective function value with a significance level of $\alpha = 0.05$ for CEC2010 problems for 30 dimensions ('+' indicates CSMO is significantly better, '-' indicates CSMO is significantly worse and '=' indicates there is no significant difference)

| | Pairwise comparison of CSMO vs. | | |
|---|---|---|---|
| Problems | ABC | CHDE | PESO |
| C01 | - | + | + |
| C02 | + | + | + |
| C03 | * | * | * |
| C04 | * | * | * |
| C05 | * | * | - |
| C06 | * | * | - |
| C07 | - | = | + |
| C08 | - | = | + |
| C09 | + | + | + |
| C10 | + | + | + |
| C11 | + | = | + |
| C12 | = | + | + |
| C13 | + | + | + |
| C14 | + | = | + |
| C15 | + | + | + |
| C16 | + | + | + |
| C17 | + | + | + |
| C18 | + | + | + |
| * represents that there is no comparison available | | | |

**Table 3.43:** Average execution time per run (in seconds) on CEC2010 Benchmark Problems for 30 dimensions

| Problems | CSMO | ABC | CHDE | PESO |
|---|---|---|---|---|
| C01 | 13.98664 | 11.241 | 13.85604 | 16.2862 |
| C02 | 5.37496 | 5.9204 | 5.30348 | 6.27464 |
| C03 | 21.386 | 21.62612 | 21.82664 | 22.037 |
| C04 | 23.29676 | 23.873 | 23.15528 | 24.15784 |
| C05 | 5.84628 | 6.40672 | 5.7782 | 6.73036 |
| C06 | 11.7292 | 12.75652 | 11.82968 | 12.77468 |
| C07 | 43.1298 | 42.76172 | 41.29888 | 43.70084 |
| C08 | 39.4982 | 29.24572 | 39.3022 | 38.12704 |
| C09 | 3.42036 | 3.9872 | 3.29032 | 5.52212 |
| C10 | 9.17212 | 10.08184 | 9.29888 | 11.2724 |
| C11 | 47.38996 | 47.8688 | 47.46912 | 48.34676 |
| C12 | 23.08752 | 23.1898 | 22.77252 | 24.19908 |
| C13 | 6.16084 | 5.73776 | 6.45248 | 7.24744 |
| C14 | 36.29884 | 9.06128 | 25.17796 | 33.4856 |
| C15 | 14.20968 | 15.05268 | 29.57508 | 15.2528 |
| C16 | 7.28812 | 7.85596 | 7.93124 | 8.78128 |
| C17 | 3.55688 | 3.9724 | 4.41244 | 5.5216 |
| C18 | 5.92072 | 6.40328 | 8.52552 | 7.33484 |

CHAPTER 4

# TOURNAMENT SELECTION BASED SPIDER MONKEY OPTIMIZATION ALGORITHM FOR UNCONSTRAINED OPTIMIZATION

In this chapter, a modified version of basic SMO [17] namely Tournament selection based Spider Monkey Optimization (TS-SMO) has been proposed for solving unconstrained continuous optimization problems. The objective is to improve the exploration ability of basic SMO by implementing tournament selection based probability scheme in it. Investigation has been made on the performance of the proposed algorithm by testing it over a set of 46 unconstrained benchmark problems and results are compared with the basic version SMO [17]. The chapter is organised as follows: In section 4.1, the motivation behind proposing this new algorithm is discussed. In section 4.2, the proposed algorithm is described. In Section 4.3, details of experimental setup are provided. In section 4.4, experimental results are discussed in order to examine the performance of the proposed algorithm and the chapter is concluded in section 4.5.

## 4.1 MOTIVATION

There are mainly two manipulation phases in basic SMO namely Local Leader Phase and Global Leader Phase, which are responsible for updating the swarm. In Local Leader Phase, all the members of the swarm get equal chance to update themselves, but in Global Leader Phase, members of the swarm get a chance to update themselves on the basis of their probability. The probability scheme used in basic SMO is fitness proportionate, which is similar to roulette wheel selection in Genetic Algorithm [77]. Due to the use of fitness based probability scheme, the members with higher fitness value have better chance to update their position as compared to the members with lower fitness value. Sometimes, even less fit members may contain some very important and useful information, but not given a chance to update due to their low fitness, this useful information is lost. Also, focussing the search on best solutions only due to their high fitness may sometimes lead to premature convergence. In

The content of this chapter contributes to the paper entitled:

Gupta, K., Deep, K., Tournament Selection Based Probability Scheme in Spider Monkey Optimization Algorithm: In Harmony Search Algorithm, Springer, Berlin, Heidelberg, vol. 382, (2016), pp. 239–250.

this chapter, an attempt has been made to overcome these issues by using tournament selection based probability based scheme.

## 4.2 THE PROPOSED TOURNAMENT SELECTION BASED SPIDER MONKEY OPTIMIZATION

### 4.2.1 TOURNAMENT SELECTION BASED PROBABILITY SCHEME

Tournament selection is one the famous selection operators in GA. Unlike, roulette wheel selection, in which only highly fit solutions are preferred, low fit solutions are also given a chance in tournament selection. Instead of fitness proportionate probability scheme in SMO, tournament selection based probability scheme has been used in TS-SMO. This probability scheme will help in increasing the exploration ability of SMO by reducing the biasness towards highly fit solutions and thus handling the problem of premature convergence by increasing diversity in the population.

### 4.2.2 IMPLEMENTATION STRATEGY OF TOURNAMENT SELECTION BASED PROBABILITY SCHEME IN SMO

TS-SMO differs from basic SMO in the use of probability scheme only. The fitness proportionate probability scheme of SMO has been replaced with tournament selection based probability scheme. Probability of a solution is used in the Global Leader Phase to update the position of spider monkeys. Higher the probability, higher the chances of that spider monkey to update its position. In TS-SMO, all the steps for executing the algorithm are same as that of basic SMO except the calculation of probability.

The steps for calculating the tournament selection based probability of solutions in the swarm is provided in Algorithm 4.1. From this algorithm, it can be observed that tournament selection has nothing to do with the magnitude of the fitness of the solutions. While the fitness proportionate probability scheme described in chapter 2 has to deal with the actual fitness of the solution, tournament selection only deals with the ranking of the solution in terms of fitness. The rank of a solution in terms of fitness is a number which indicates the superiority of the solution over other solutions in the population. For example, if the rank of a solution is 5, it means this solution is better than five other solutions in the population. Both the fitness proportionate probability scheme and tournament selection based probability scheme perform almost similar if there is not much variation in the lowest and highest fitness value of the swarm. But if there is much variation, both will behave in a different manner. Pseudocode for TS-SMO is given in Algorithm 4.2.

```
Begin
  Let $a_i$ be the rank of $i^{th}$ spider monkey
    For $i$ = 1 to $N$ Do
        $a_i$ = 0
        For $j$ = 1 to $N$ Do
            If ($fit_i \geq fit_j$) Then
                $a_i = a_i + 1$
            End If
        End For
    End For
    For $i$ = 1 to $N$ Do
        $prob_i = \frac{a_i}{\sum_{i=1}^{n} a_i}$
    End For
End
```

**Algorithm 4.1**: Steps for calculating probability in TS-SMO

## 4.3 EXPERIMENTAL SETUP

### 4.3.1 BENCHMARK PROBLEMS

In order to check the performance of the proposed algorithm TS-SMO, experiments have been performed over a large set of 46 benchmark problems. This benchmark set includes 26 problems considered for experiment in [17] as well as some additional benchmark problems from the literature for an extensive analysis of the performance of the proposed algorithm. This benchmark set is large enough to include problems having objective functions of different characteristics like unimodal, multimodal, separable, non-separable, discontinuous etc. The optimization problems in the benchmark set are broadly divided into two categories namely scalable and non-scalable. Out of these 46 problems, first 30 problems are scalable and rest of the problems are non-scalable. The dimension of all the scalable problems is same and is fixed to be 30. All the problems in the benchmark set are of minimization type. List of problems along with their search range and objective function is provided in the Appendix III. For scalable problems, lower and upper bounds on the decision variables are same for each dimension of the solution. Also, lower and upper bounds of each decision variable are same in non-scalable problems except two cases: problem no.32 and problem no. 36.

```
Begin
    Initialize the swarm using equation (2.1)
    Initialize $LLlt, GLlt, Pr, MG$
    Iteration=0
    Calculate fitness value of the position of each spider monkey in the swarm
    Select Global Leader and Local Leaders by applying Greedy Selection
    While (termination criterion is not satisfied) Do
        //Local Leader Phase (Algorithm 2.1)
        //Calculate Probability of each spider monkey (Algorithm 4.1)
        //Global Leader Phase (Algorithm 2.2)
        //Global Leader Learning Phase (Algorithm 2.3)
        //Local Leader Learning Phase (Algorithm 2.4)
        //Local Leader Decision Phase (Algorithm 2.5)
        //Global Leader Decision Phase (Algorithm 2.6)
        Iteration = Iteration +1
    End While
End
```

**Algorithm 4.2:** Pseudocode for TS-SMO

### 4.3.2 SETTING OF CONTROL PARAMETERS AND TERMINATION CRITERIA

TS-SMO contains one more parameter in addition to the parameters of basic SMO. The parameter associated with tournament selection is the size of the tournament. This size indicates the number of solutions which will participate in the tournament. Tournament of size two has been used in TS-SMO. The reason for taking size two is to keep the tournament selection process simplest. The performance of TS-SMO has been compared with basic SMO. In order to maintain the consistency, same parameter setting has been adopted for both the algorithms.

The setting of parameters is given below:

Swarm size =150

perturbation rate *(Pr)* = linearly increasing ([0.1, 0.4])

maximum number of groups *(MG)* = 5

local leader limit *(LLlt)* = 100

global leader limit *(GLlt)* = 50

Total number of runs = 100

Maximum number of iteration = 4000

Acceptable error = $10^{-5}$

Termination criterion = either maximum number of iterations are performed or acceptable error is achieved.

Here, error is the absolute difference between the global optimal solution and the objective function value of the global leader. In order to make fair comparison between the two algorithms, both the algorithms start with the same initial swarm in every run.

### 4.3.3 PERFORMANCE EVALUATION CRITERIA

In order to access the impact of using tournament selection probability scheme, TS-SMO has been compared with basic SMO using various performance metrics. Comparison of SMO and TS-SMO has been made on the basis of reliability, efficiency and accuracy. Reliability is measured in terms of success rate. The formula for success rate is same as mentioned in eq. (3.4) in subsection 3.3.1 of chapter 3. A run is considered to be successful if error value of the global leader is less than or equal to the acceptable error which is set to be $10^{-5}$ here. If a run is successful for a particular problem, the problem is said to be solved in that particular run. Efficiency is the measured in terms of average number of function evaluations for successful runs and accuracy is the degree of precision in locating the global optima and it is measured in terms of error values obtained in all the runs. For comparison of these two algorithms, success rate, average number of function evaluations of successful runs and best, average and worst of error values obtained in 100 independent runs has been recorded. Results of scalable and non-scalable problems have been discussed separately followed by overall conclusion of the performance of both the algorithms.

Results of both the algorithms have been analyzed in three ways. First, the impact of using Tournament selection based probability scheme in SMO has been studied individually in terms of success rate, average number of function evaluations for successful runs and best, average and worst of error values obtained in 100 runs. For this purpose, two comparison criteria have been defined.

**Criterion 1** is used to compare the reliability and efficiency of both the algorithms. Comparison according to this criterion has been made on the basis of following information.

- Success rate
- Average number of function evaluations for successful runs

Solving a problem is important, so first preference is given to the success rate. The criterion 1 is only applied when atleast one of the algorithm is said to have positive number of successful runs. There can be three cases:

**Case1**: When both the algorithms have different success rate

In this case, algorithm with higher success rate is the winner and is said to perform strictly better than the other.

**Case 2:** When both algorithms have same success rate and different average number of function evaluations for successful runs.

In this case, algorithm with less average number of function evaluations for successful runs is the winner and is said to perform better than the other.

**Case 3:** In extreme case, where both the algorithms have same success rate as well as same number of average function evaluations for successful runs, both the algorithms is considered equivalent.

**Criterion 2** is employed to compare the accuracy of both the algorithms. Comparison has been done on the basis of following:

- Best, average and worst of the error values obtained in 100 runs

Following cases are possible for comparison:

**Case 1**: When an algorithm say A has all the three (best, average and worst) error values less than the other algorithm say B. Then algorithm A is said to be strictly better than algorithm B.

**Case 2**: When best and average error values of A is strictly less than those of B, and the worst error value of A is equal to that of B, then A is said to perform better then B.

**Case 3**: When both A and B have same best, average and worst error values, then both A and B are said to be equivalent.

Second method of analysis is to check the performance of the algorithms statistically. For this purpose, paired t-test at 0.05 level of significance has been used. T-test has been applied to those benchmark problems where the algorithms satisfy case 2 of criterion 1 to see if there is significant difference in the average number of function evaluations of these problems. In other words, t-test has been applied to the average number of function evaluations for successful runs of those problems where both the algorithms have same success rate to see if the difference between the average number of function evaluations for successful runs is significant or not. To apply t-test, we start with null hypothesis that “there is no difference between algorithms”. The alternative hypothesis is that “there is a difference”. Average number of function evaluations of successful runs is taken for each algorithm as a sampling data and the results have been presented for both scalable and non-scalable problems in tabular form. “=” in the table indicates there is no significant difference between the average of function evaluations of two algorithms and “+” and “-” indicates that TS-SMO performs significantly better and worse than SMO respectively.

Third method of analysis is to compare the relative performance of two algorithms on the basis of performance index (PI’s) described in Appendix IV. The weight given to the success rate increases from 0 to 1 while the weight for average number of function evaluations decreases from 1 to 0.

## 4.4 EXPERIMENTAL RESULTS AND DISCUSSION

Tables 4.1-4.4 and Tables 4.5-4.8 present the results for scalable and non-scalable problems respectively. Better entries in the tables appear in bold. N.A. in the tables represents the non-availability of results.

### 4.4.1 DISCUSSION OF RESULTS FOR SCALABLE PROBLEMS

From Table 4.1, it can be observed that out of 30 problems, both SMO and TS-SMO have 100 percent success rate in twenty two problems (No.’s 1, 2, 3, 5, 6, 7, 8, 9, 10, 12, 13, 14, 16, 18, 19, 23, 24, 25, 26, 27, 29, 30). In two problems (No.’s 11 and 22), TS-SMO performs strictly better than SMO according to criterion 1 (case 1). Out of these twenty two problems, where both the algorithms have 100 percent success rate, TS-SMO performs better than SMO in twenty problems (No.’s 1, 2, 6, 7, 8, 9, 10, 12, 13, 14, 16, 18, 19, 23, 24, 25, 26, 27, 29, 30) according to criterion 1 (case 2). Also, it can be observed from Table 4.1 that TS-SMO has better success rate than SMO in two problems (No.’s 11 and 22) though it comes at the cost of more number of function evaluations. Five problems (No.’s 4, 15, 17, 20, and 28) are not solved by any of the two algorithms.

Table 4.2 presents the t-test results for TS-SMO against SMO which clearly shows that TS-SMO performs significantly better than SMO on most of the problems. Only in problem no. 5, TS-SMO has worse performance comparatively. In two problems (No.’s 3 and 29), there is no significant difference between the performances of both the algorithms.

To check the solution quality obtained by both the algorithms, best, average and worst of the error values have been provided in Table 4.3. For best error values, TS-SMO is better than SMO over fifteen problems (No.’s 4, 7, 8, 9, 11, 13, 17, 18, 22, 23, 24, 25, 26, 28, 30) and in two problems (No.’s 15 and 19), both the algorithms have same best error value. For average error values, TS-SMO is better than SMO over fifteen problems (No.’s 2, 4, 6, 8, 9, 10, 17, 18, 21, 24, 25, 27, 28, 29, 30) and there are two problems (No.’s 1 and 19) where both TS-SMO and SMO have same average error. For worst error values, TS-SMO is better than SMO over eleven problems (No.’s 3, 7, 14, 15, 16, 17, 21, 24, 25, 26, 27) and both the algorithms have same worst error value on seven problems (No.’s 1, 2, 9, 10, 13, 19 and 30). There are three problems (No.’s 17, 24 and 25), where TS-SMO is strictly better than SMO according to criterion 2 (case 1). There are two problems (No.’s 9 and 30) where TS-SMO performs better than SMO according to criterion 2 (case 2). Both the algorithms perform identically on problem no. 19 according to criterion 2 (case 3). In three problems (No.’s 5, 12 and 20), SMO performs strictly better than TS-SMO.

From Tables 4.1 and 4.3, it can be concluded that in five problems (No.’s 4, 15, 17, 20, and 28) where both the algorithms fail to solve the problem, average error value obtained by TS-SMO is better than the average error value obtained by SMO except two problems (No.’s

15 and 20). Also, it can be seen for problem no. 19, global optima is obtained by both the algorithms.

Table 4.4 shows that average execution time taken (in seconds) by both the algorithms per run for solving scalable problems of 30 dimensions. From this table, it can be observed that the time taken by both the algorithms is very small and there is not much difference in the execution time of both the algorithms. So, the comparison of algorithms on the basis of execution time has been avoided. So, this table should be considered for information purpose rather than comparison purpose.

Also, in order to analyse the superiority of one algorithm over the another, performance index graph has been drawn for scalable problems in Figure 4.1 from which it is clear that the performance of TS-SMO is better than SMO for all the values of weight w (from 0 to 1). Also, it can be observed that the performance index value for both the algorithms decreases with the increase in the weight given to the success rate.

The convergence graphs for all the scalable problems have been provided in Figures 4.2-4.5. The horizontal axis represents the number of iterations and vertical axis represents the function error value. Some convergence graphs, where the range of error values is very large have been plotted using logarithmic scale. In these graphs, the function error value is labelled as log of error value.

### 4.4.2 DISCUSSION OF RESULTS FOR NON-SCALABLE PROBLEMS

Table 4.5 provides the success rate and average number of function evaluations for successful runs of non-scalable problems. From this table, it can be seen that SMO is strictly better than TS-SMO on problem no. 32 according to criterion 1 (case 1). Both SMO and TS-SMO have 100 percent success rate in twelve problems (No.'s 31, 33, 34, 35, 36, 37, 38, 39, 40, 42, 44, 46). In six problems (No.'s 31, 34, 35, 36, 37 and 40), TS-SMO is better than SMO according to criterion 1 (case 2). In six problems (No.'s 33, 38, 39, 42, 44 and 46), SMO is strictly better than TS-SMO. Both the algorithms failed to solve three problems (No.'s 41, 43 and 45) i.e. both the algorithms have zero percent success rate on these three problems.

Table 4.6 presents the t-test results for TS-SMO against SMO which shows a mixed response. Out of twelve problems (No.'s 31, 33, 34, 35, 36, 37, 38, 39, 40, 42, 44, 46), there are only three problems (No.'s 34, 35 and 40) where TS-SMO performs significantly better than SMO. There are four problems (No.'s 33, 38, 39 and 46), where performance of TS-

SMO is worse and there is no significant difference in the performance of both the algorithms on four problems (No.’s 31, 36, 37 and 44).

Best, average and worst of the error values obtained by both the algorithms have been listed in Table 4.7. For best values, TS-SMO is better than SMO over nine problems (No.’s 31, 33, 34, 36, 37, 38, 40, 44, 46). Out of these nine problems, both the algorithms have same best error values in four problems (No.’s 39, 41, 43 and 45) and SMO is better than TS-SMO over three problems (No.’s 32, 35 and 42). For average error values, TS-SMO is better than SMO over seven problems (No.’s 31, 33, 35, 37, 38, 40 and 44), both have same average error value on problem no. 45 and SMO is better than TS-SMO over eight problems (No.’s 32, 34, 36, 39, 41, 42, 43 and 46). For worst error values, TS-SMO outperforms SMO over six problems (No.’s 31, 34, 35, 38, 39 and 40) and SMO outperforms TS-SMO over seven problems (No.’s 33, 36, 41, 42, 43, 44 and 46). Both the algorithms have same worst error values in three problems (No.’s 32, 37 and 45). In three problems (No.’s 31, 38 and 40), TS-SMO performs strictly better than SMO according to criterion 2 (case 1).

The average execution time taken (in seconds) by both the algorithms per run for solving non-scalable problems is given in Table 4.8. From this table, it can be observed that the time taken by both the algorithms is very small and there is not much difference in the execution time of both the algorithms.

The performance index graph for non-scalable problems has been provided in Figure 4.6. From the graph, it is clear that the performance of SMO is better than TS-SMO for all the values of w (from 0 to 1). Also, it can be observed that the performance index value increases with the increase in the weight given to the success rate. The convergence graphs for all the non-scalable problems have been provided in Figures 4.7-4.8.

From the above discussion on the results of both scalable and non-scalable problems, it can be concluded that though TS-SMO is showing good performance on scalable problems (listed in benchmark set) in terms of reliability, efficiency and accuracy as compared to SMO, its performance is moderate on non-scalable problems.

## 4.5 CONCLUSIONS

In this chapter, a new version of SMO named as Tournament Selection based Spider Monkey Optimization (TS-SMO) has been proposed. The novelty of this version lies in the use of tournament selection based probability scheme instead of fitness proportionate probability scheme which has been used in basic SMO. Maintaining a high level diversity while preserving convergence speed are two contradictory and necessary features of a metaheuristic and the numerical results show that TS-SMO does a good job in balancing both the features by performing well on scalable problems. But it shows only moderate performance on non-scalable problems. In future, other modifications can be proposed to improve the performance of TS-SMO on non-scalable problems also. The question may arise why TS-SMO has been compared only with SMO, not with other metaheuristics as in chapter 3? The answer is that the basic SMO has already been compared with other metaheuristics in [17]. The results demonstrated the competitiveness of SMO in solving unconstrained optimization problems in comparison to other algorithms used for the experiment. The purpose of this chapter is only to see whether the incorporation of tournament based probability scheme has any positive impact on the performance of basic SMO. Therefore, it has been compared with basic SMO only.

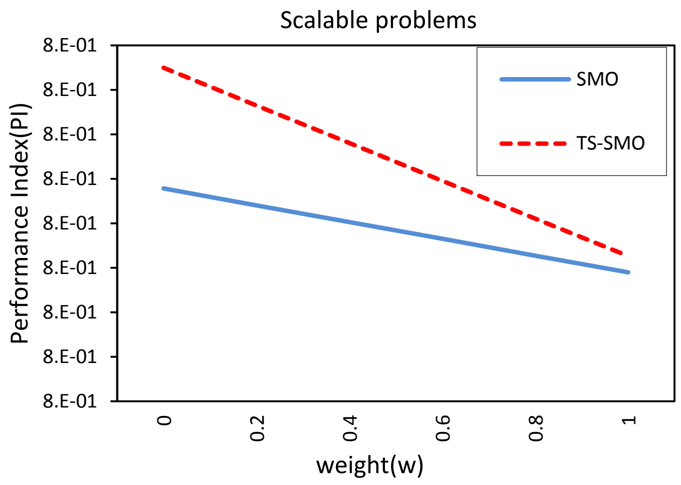

**Figure 4.1**: Performance Index graph for Scalable problems for success rate versus average number of function evaluations for successful runs

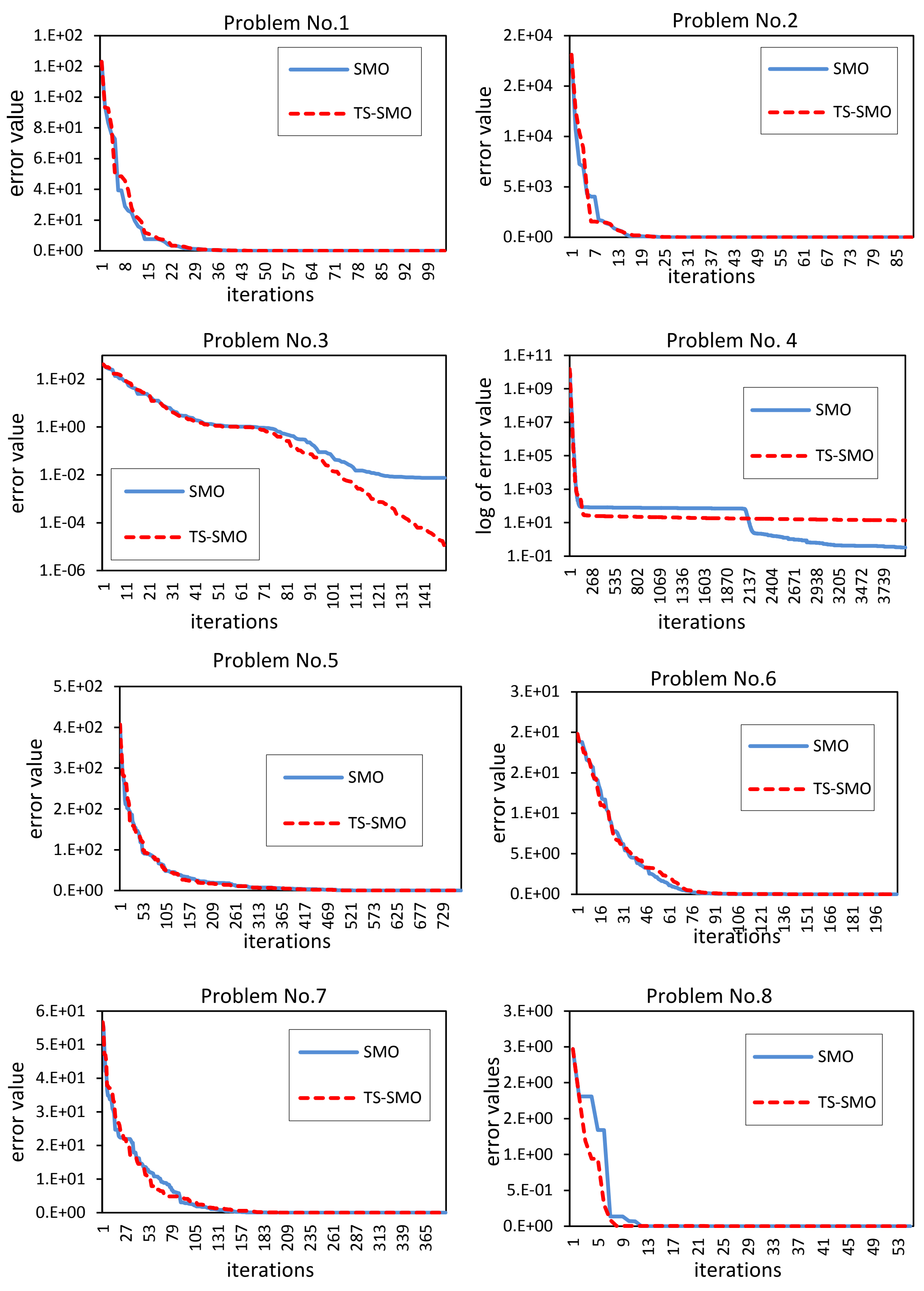

**Figure 4.2**: Convergence graphs for Problem No.1-Problem No.8

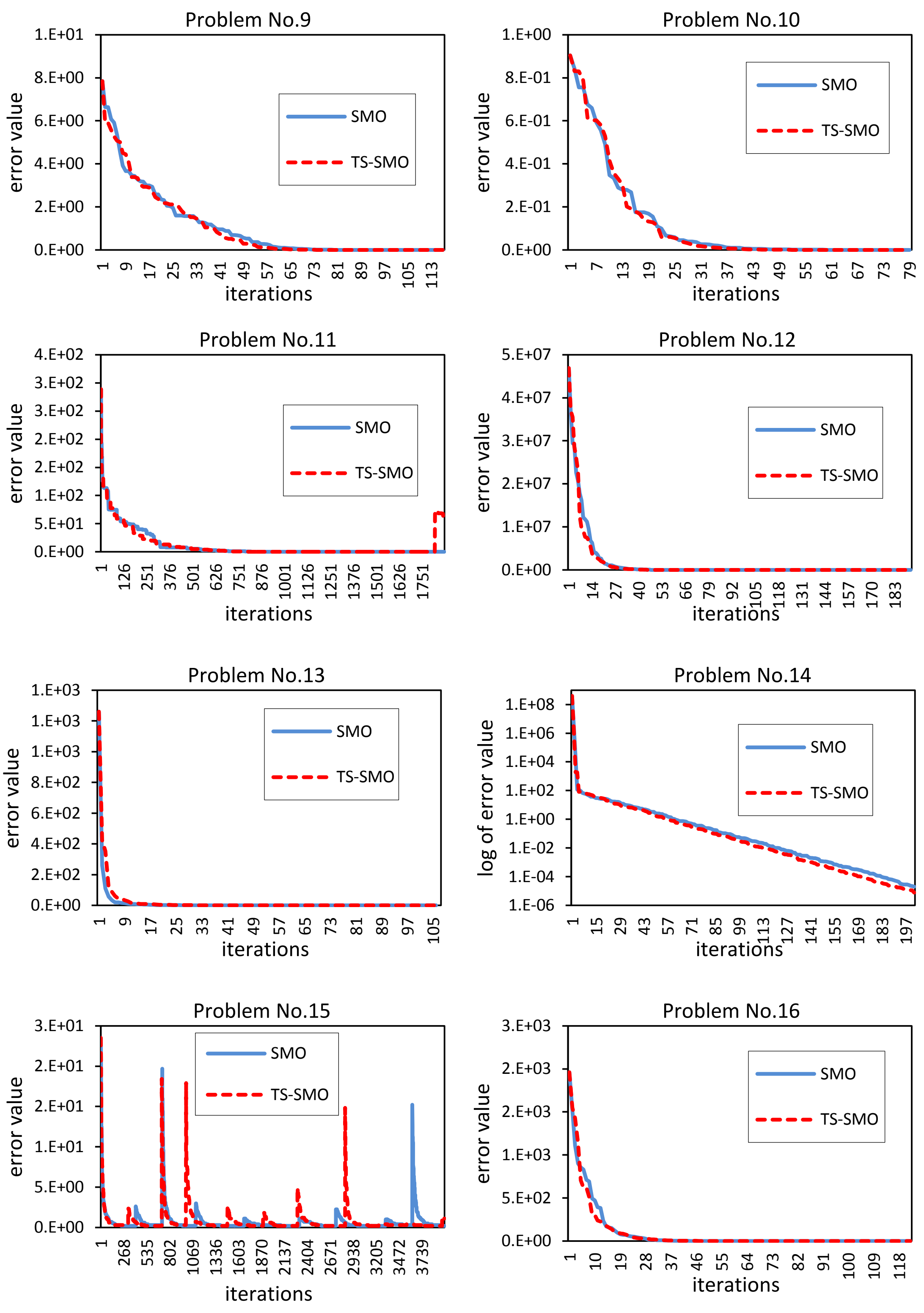

**Figure 4.3**: Convergence graphs for Problem No.9-Problem No.16

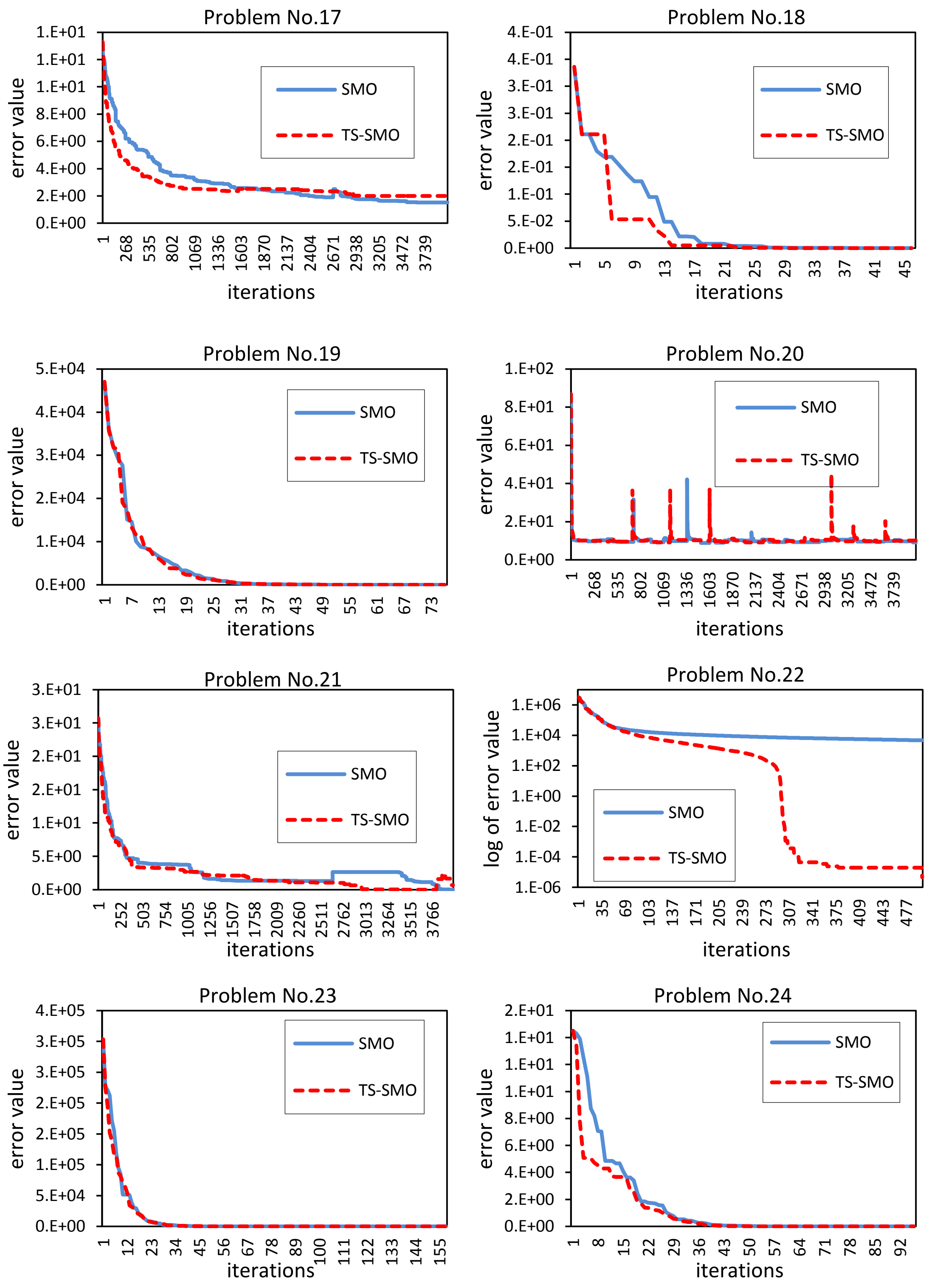

**Figure 4.4:** Convergence graphs for Problem No.17-Problem No.24

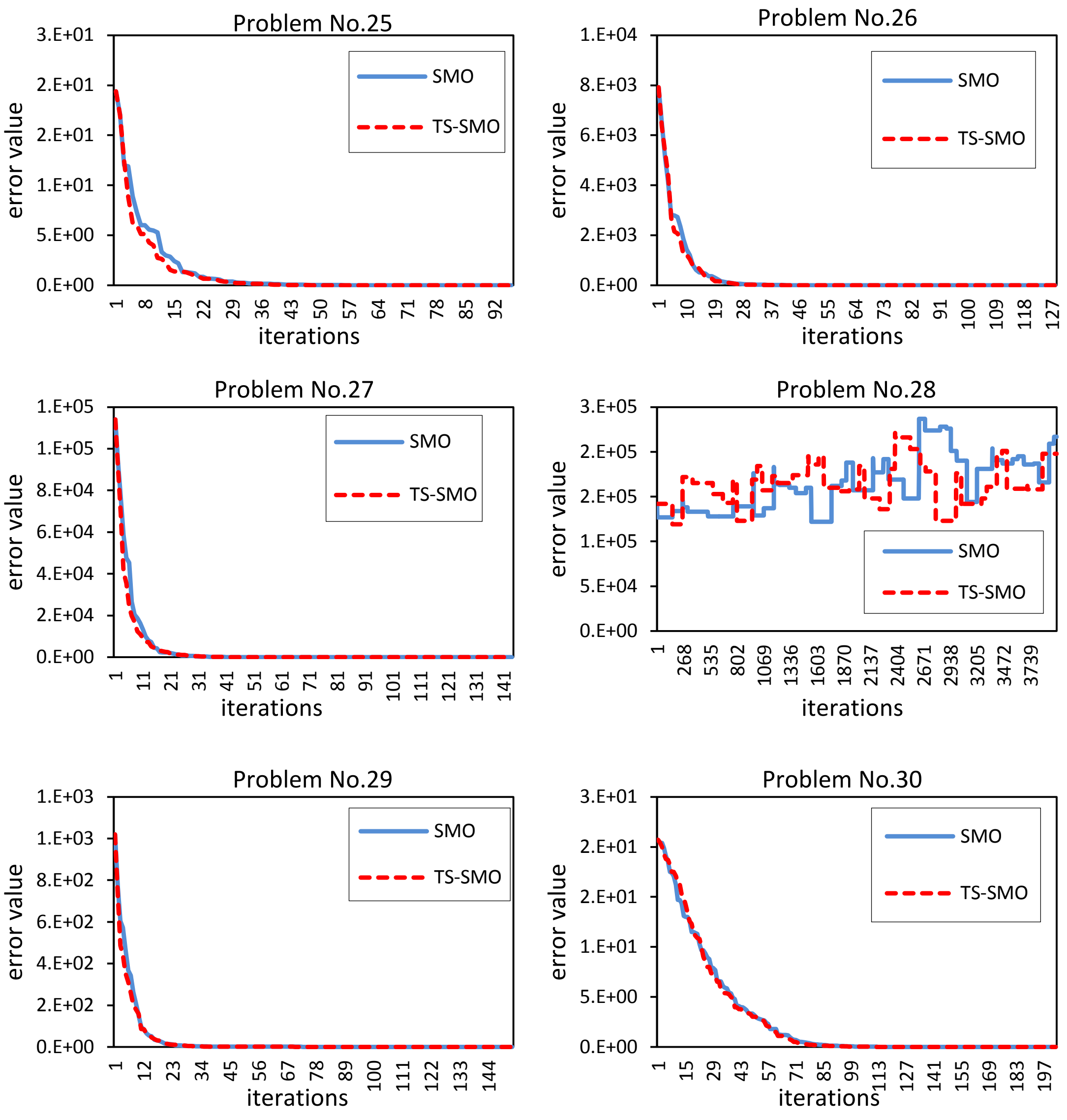

**Figure 4.5:** Convergence graphs for Problem No.25-Problem No.30

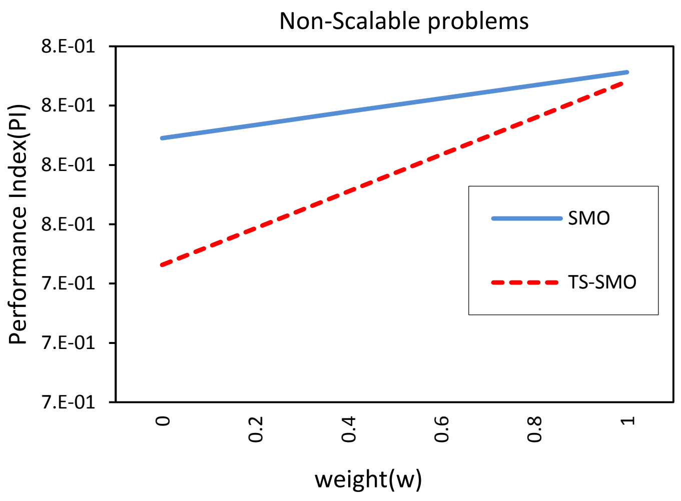

**Figure 4.6:** Performance Index graph for Non-Scalable problems for success rate versus average number of function evaluations for successful runs

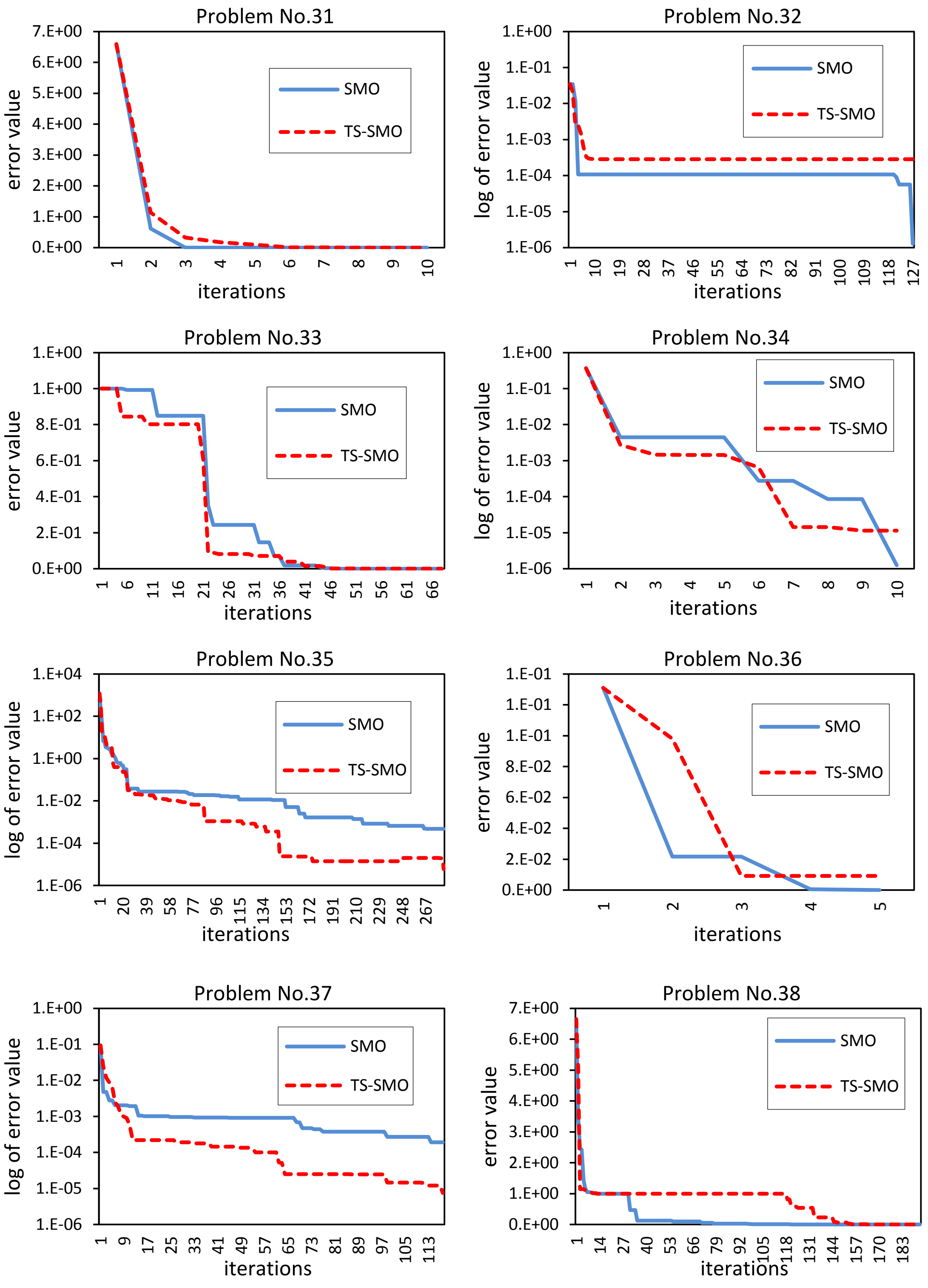

**Figure 4.7:** Convergence graphs for Problem No.31-Problem No.38

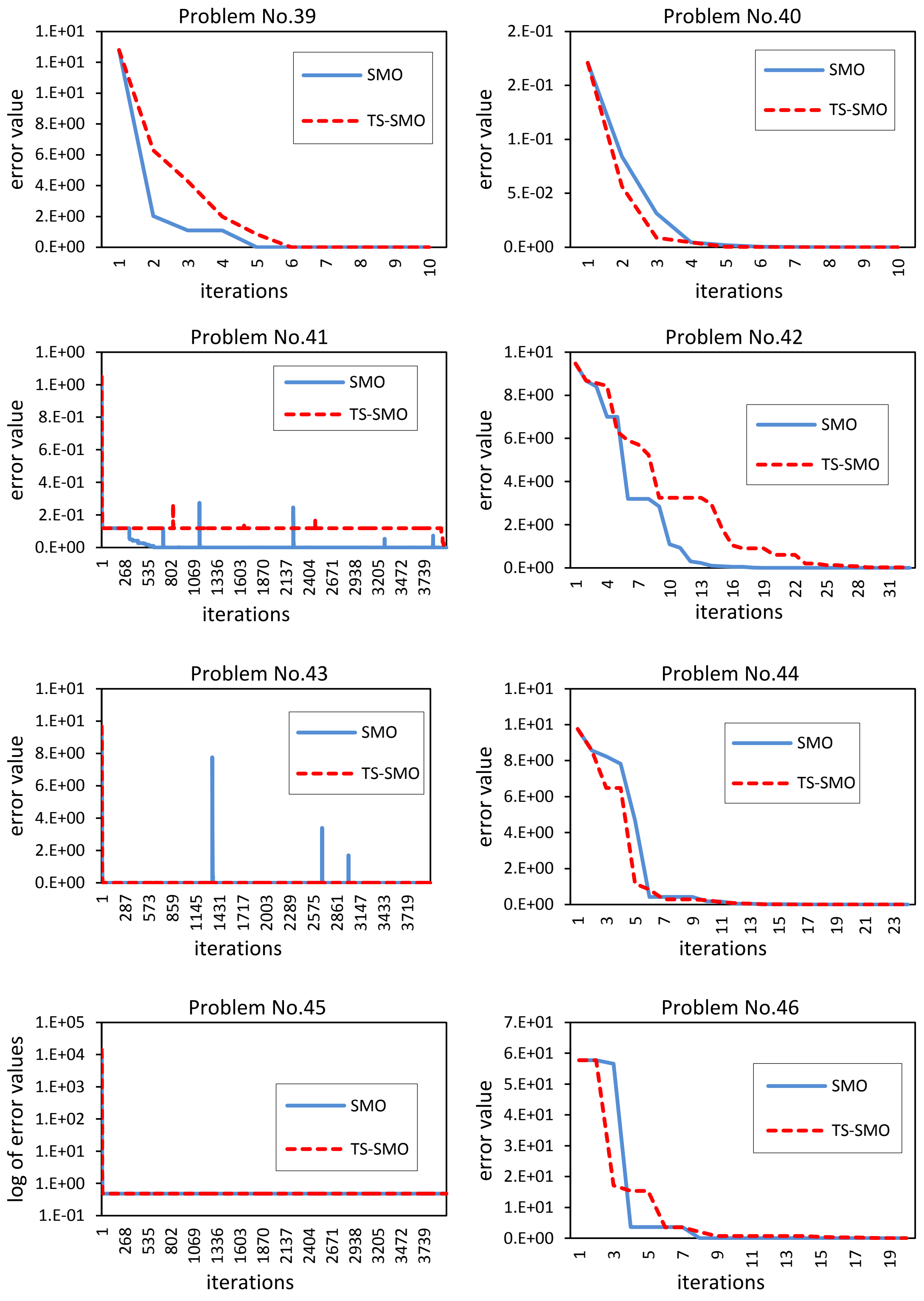

**Figure 4.8:** Convergence graphs for Problem No.39-Problem No.46

**Table 4.1**: Success Rate and Average number of function evaluations for successful runs (Scalable problems-30 dimensions)

| Problem No. | Success Rate | | Average number of function evaluations for successful runs | |
|---|---|---|---|---|
| | SMO | TS-SMO | SMO | TS-SMO |
| 1 | 100 | 100 | 33192 | **32128** |
| 2 | 100 | 100 | 26566 | **25923** |
| 3 | 100 | 100 | **103625** | 108601 |
| 4 | 0 | 0 | N.A. | N.A. |
| 5 | 100 | 100 | **229495** | 254951 |
| 6 | 100 | 100 | 63917 | **61660** |
| 7 | 100 | 100 | 189827 | **149630** |
| 8 | 100 | 100 | 24027 | **19040** |
| 9 | 100 | 100 | 37701 | **34987** |
| 10 | 100 | 100 | 25421 | **24404** |
| 11 | 47 | **61** | **650570** | 682431 |
| 12 | 100 | 100 | 58502 | **56807** |
| 13 | 100 | 100 | 32220 | **31231** |
| 14 | 100 | 100 | 62664 | **60888** |
| 15 | 0 | 0 | N.A. | N.A. |
| 16 | 100 | 100 | 38287 | **37013** |
| 17 | 0 | 0 | N.A. | N.A. |
| 18 | 100 | 100 | 14002 | **13294** |
| 19 | 100 | 100 | 22628 | **22437** |
| 20 | 0 | 0 | N.A. | N.A. |
| 21 | **7** | 2 | **1051267** | 1117825 |
| 22 | 83 | **85** | 354061 | **349233** |
| 23 | 100 | 100 | 48381 | **47167** |
| 24 | 100 | 100 | 30839 | **29457** |
| 25 | 100 | 100 | 30695 | **29586** |
| 26 | 100 | 100 | 39151 | **37892** |
| 27 | 100 | 100 | 44665 | **43394** |
| 28 | 0 | 0 | N.A. | N.A. |
| 29 | 100 | 100 | 106373 | **101093** |
| 30 | 100 | 100 | 63980 | **61938** |

**Table 4.2**: T-test results for problems having identical success rate for SMO and TS-SMO (Scalable problems-30 dimensions)

| Problem No. | Sign | Problem No. | Sign |
|---|---|---|---|
| 1 | + | 14 | + |
| 2 | + | 16 | + |
| 3 | = | 18 | + |
| 5 | - | 19 | + |
| 6 | + | 23 | + |
| 7 | + | 24 | + |
| 8 | + | 25 | + |
| 9 | + | 26 | + |
| 10 | + | 27 | + |
| 12 | + | 29 | = |
| 13 | + | 30 | + |

**Table 4.3**: Best, Average and Worst of error values obtained in 100 runs (Scalable problems-30 dimensions)

| Problem No. | Best | | Average | | Worst | |
|---|---|---|---|---|---|---|
| | SMO | TS-SMO | SMO | TS-SMO | SMO | TS-SMO |
| 1 | **6.01E-06** | 6.33E-06 | 8.72E-06 | 8.72E-06 | 9.99E-06 | 9.99E-06 |
| 2 | **3.12E-06** | 4.25E-06 | 8.03E-06 | **7.98E-06** | 9.99E-06 | 9.99E-06 |
| 3 | **3.70E-06** | 5.80E-06 | **8.52E-06** | 8.73E-06 | 9.99E-06 | **9.98E-06** |
| 4 | 1.57E-02 | **2.36E-03** | 1.63E+01 | **1.62E+01** | **8.08E+01** | 8.30E+01 |
| 5 | **2.17E-06** | 5.40E-06 | **8.42E-06** | 9.05E-06 | **9.97E-06** | 1.00E-05 |
| 6 | **7.58E-06** | 7.86E-06 | 9.43E-06 | **9.35E-06** | **9.98E-06** | 9.99E-06 |
| 7 | 7.68E-06 | **7.67E-06** | **9.36E-06** | 9.47E-06 | 9.99E-06 | **9.98E-06** |
| 8 | 7.54E-09 | **7.12E-11** | 5.11E-06 | **5.09E-06** | **9.98E-06** | 9.99E-06 |
| 9 | 6.42E-06 | **1.31E-07** | 8.92E-06 | **8.49E-06** | 9.99E-06 | 9.99E-06 |
| 10 | **5.15E-06** | 5.36E-06 | 8.85E-06 | **8.68E-06** | 1.00E-05 | 1.00E-05 |
| 11 | 7.36E-06 | **5.72E-06** | **4.17E+00** | 4.73E+00 | **4.62E+01** | 5.80E+01 |
| 12 | **5.62E-06** | 5.72E-06 | **8.74E-06** | 8.75E-06 | **9.98E-06** | 9.99E-06 |
| 13 | 5.87E-06 | **4.78E-06** | **8.71E-06** | 8.79E-06 | 9.99E-06 | 9.99E-06 |
| 14 | **7.44E-06** | 7.78E-06 | **9.30E-06** | 9.37E-06 | 9.99E-06 | **9.98E-06** |
| 15 | 2.00E-01 | 2.00E-01 | **5.85E-01** | 6.02E-01 | 1.05E+01 | **9.47E+00** |
| 16 | **5.36E-06** | 6.02E-06 | **8.63E-06** | 8.79E-06 | 9.99E-06 | **9.98E-06** |
| 17 | 9.38E-03 | **2.72E-04** | 1.49E+00 | **1.38E+00** | 8.67E+00 | **4.34E+00** |
| 18 | 2.74E-06 | **1.38E-06** | 7.60E-06 | **7.09E-06** | **9.97E-06** | 9.98E-06 |
| 19 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 |
| 20 | **7.73E+00** | 8.11E+00 | **9.73E+00** | 9.78E+00 | **1.22E+01** | 1.74E+01 |
| 21 | **7.78E-06** | 9.50E-06 | 1.34E+00 | **1.12E+00** | 4.73E+00 | **2.97E+00** |
| 22 | 7.58E-08 | **2.64E-08** | **1.54E+02** | 5.44E+03 | **1.45E+04** | 5.44E+05 |
| 23 | 6.17E-06 | **5.14E-06** | **8.60E-06** | 8.71E-06 | **9.93E-06** | 1.00E-05 |
| 24 | 6.17E-06 | **5.34E-06** | 8.77E-06 | **8.59E-06** | 9.99E-06 | **9.97E-06** |
| 25 | 5.98E-06 | **4.95E-06** | 8.78E-06 | **8.65E-06** | 1.00E-05 | **9.96E-06** |
| 26 | 5.97E-06 | **5.22E-06** | **8.80E-06** | 8.86E-06 | 9.99E-06 | **9.95E-06** |
| 27 | **5.67E-06** | 6.38E-06 | 8.79E-06 | **8.67E-06** | 1.00E-05 | **9.95E-06** |
| 28 | 1.24E+05 | **1.22E+05** | 1.91E+05 | **1.88E+05** | **2.41E+05** | 2.51E+05 |
| 29 | **3.94E-06** | 4.54E-06 | 8.67E-06 | **8.57E-06** | **9.97E-06** | 9.99E-06 |
| 30 | 7.83E-06 | **7.12E-06** | 9.42E-06 | **9.33E-06** | 9.99E-06 | 9.99E-06 |

**Table 4.4**: Average execution time per run (in seconds) (Scalable problems-30 dimensions)

| Problem No. | SMO | TS-SMO | Problem No. | SMO | TS-SMO |
|---|---|---|---|---|---|
| 1 | 0.06337 | 0.15394 | 16 | 0.07646 | 0.1767 |
| 2 | 0.05144 | 0.12519 | 17 | 106.9177 | 99.91294 |
| 3 | 0.63335 | 1.02872 | 18 | 0.54852 | 0.51132 |
| 4 | 2.20541 | 5.80021 | 19 | 0.2067 | 0.24968 |
| 5 | 0.95737 | 1.81293 | 20 | 56.79807 | 56.42279 |
| 6 | 0.27407 | 0.44748 | 21 | 12.82866 | 13.96821 |
| 7 | 0.74151 | 1.02674 | 22 | 22.34735 | 20.51262 |
| 8 | 1.80296 | 1.50035 | 23 | 0.25422 | 0.33242 |
| 9 | 0.15978 | 0.25221 | 24 | 0.29362 | 0.28435 |
| 10 | 0.04684 | 0.1165 | 25 | 0.29653 | 0.28497 |
| 11 | 1.90559 | 4.60357 | 26 | 1.66475 | 1.51592 |
| 12 | 0.10578 | 0.26914 | 27 | 0.13941 | 0.21441 |
| 13 | 2.10971 | 2.14213 | 28 | 7.6486 | 9.40759 |
| 14 | 0.11883 | 0.2964 | 29 | 1.01688 | 0.9486 |
| 15 | 51.89418 | 56.31783 | 30 | 0.45603 | 0.45293 |

**Table 4.5**: Success Rate and Average number of function evaluations for successful runs (Non-Scalable problems)

| Problem No. | Success Rate | | Average number of function evaluations for successful runs | |
|---|---|---|---|---|
| | SMO | TS-SMO | SMO | TS-SMO |
| 31 | 100 | 100 | 3738 | **3636** |
| 32 | **98** | 93 | **430014** | 465326 |
| 33 | 100 | 100 | **18801** | 19552 |
| 34 | 100 | 100 | 3585 | **3415** |
| 35 | 100 | 100 | 256584 | **211382** |
| 36 | 100 | 100 | 3304 | **3199** |
| 37 | 100 | 100 | 131394 | **131171** |
| 38 | 100 | 100 | **26119** | 33359 |
| 39 | 100 | 100 | **2009** | 2281 |
| 40 | 100 | 100 | 3071 | **2793** |
| 41 | 0 | 0 | N.A. | N.A. |
| 42 | 100 | 100 | **24508** | 45553 |
| 43 | 0 | 0 | N.A. | N.A. |
| 44 | 100 | 100 | **13458** | 13519 |
| 45 | 0 | 0 | N.A. | N.A. |
| 46 | 100 | 100 | **10737** | 12525 |

**Table 4.6**: T-test results for problems having identical success rate for SMO and TS-SMO (Non-Scalable problems)

| Problem no. | sign | Problem no. | sign |
|---|---|---|---|
| 31 | = | 38 | - |
| 33 | - | 39 | - |
| 34 | + | 40 | + |
| 35 | + | 42 | - |
| 36 | = | 44 | = |
| 37 | = | 46 | - |

**Table 4.7**: Best, Average and Worst of error values obtained in 100 runs (Non-Scalable problems)

| Problem No. | Best | | Average | | Worst | |
|---|---|---|---|---|---|---|
| | SMO | TS-SMO | SMO | TS-SMO | SMO | TS-SMO |
| 31 | 8.53E-08 | **1.42E-08** | 4.82E-06 | **4.61E-06** | 9.99E-06 | **9.96E-06** |
| 32 | **6.79E-08** | 8.20E-08 | **1.11E-05** | 2.24E-05 | 2.87E-04 | 2.87E-04 |
| 33 | 1.21E-07 | **5.02E-08** | 4.87E-06 | **4.69E-06** | **9.48E-06** | 9.98E-06 |
| 34 | 1.74E-07 | **2.22E-08** | **4.35E-06** | 4.74E-06 | 9.98E-06 | **9.91E-06** |
| 35 | **1.22E-06** | 1.88E-06 | 7.24E-06 | **6.92E-06** | 1.00E-05 | **9.96E-06** |
| 36 | 9.26E-08 | **3.21E-08** | **4.37E-06** | 4.39E-06 | **9.75E-06** | 9.98E-06 |
| 37 | 2.41E-06 | **1.77E-06** | 7.95E-06 | **7.78E-06** | 9.99E-06 | 9.99E-06 |
| 38 | 1.42E-06 | **6.44E-07** | 6.25E-06 | **6.11E-06** | 9.91E-06 | **9.88E-06** |
| 39 | 3.75E-06 | 3.75E-06 | **4.99E-06** | 5.31E-06 | 9.96E-06 | **9.91E-06** |
| 40 | 1.88E-08 | **1.53E-08** | 4.26E-06 | **3.92E-06** | 9.94E-06 | **9.93E-06** |
| 41 | 3.75E-04 | 3.75E-04 | **3.88E-04** | 2.41E-02 | **1.28E-03** | 1.19E-01 |
| 42 | **1.01E-06** | 1.44E-06 | **5.77E-06** | 6.43E-06 | **9.88E-06** | 9.93E-06 |
| 43 | 5.96E-05 | 5.96E-05 | **5.96E-05** | 6.12E-05 | **5.96E-05** | 2.20E-04 |
| 44 | 1.74E-08 | **3.21E-09** | 4.48E-06 | **3.89E-06** | **9.92E-06** | 9.93E-06 |
| 45 | 4.82E-01 | 4.82E-01 | 4.82E-01 | 4.82E-01 | 4.82E-01 | 4.82E-01 |
| 46 | 5.60E-08 | **2.58E-08** | **3.65E-06** | 4.47E-06 | **9.70E-06** | 9.93E-06 |

**Table 4.8**: Average execution time per run (in seconds) (Non-Scalable problems)

| Problem No. | SMO | TS-SMO | Problem No. | SMO | TS-SMO |
|---|---|---|---|---|---|
| 31 | 0.01315 | 0.01949 | 39 | 0.00136 | 0.00757 |
| 32 | 1.32419 | 2.97114 | 40 | 0.00194 | 0.00934 |
| 33 | 0.06098 | 0.1135 | 41 | 1.20498 | 5.84051 |
| 34 | 0.01496 | 0.02244 | 42 | 0.58038 | 1.19203 |
| 35 | 2.09645 | 2.20515 | 43 | 47.63118 | 49.03274 |
| 36 | 0.00553 | 0.01419 | 44 | 0.70029 | 0.68856 |
| 37 | 1.95449 | 2.17146 | 45 | 14.31415 | 17.63867 |
| 38 | 0.00934 | 0.10548 | 46 | 0.01287 | 0.05063 |

**CHAPTER 5**

# QUADRATIC APPROXIMATION BASED SPIDER MONKEY OPTIMIZATION ALGORITHM FOR UNCONSTRAINED OPTIMIZATION

The present chapter introduces a new version of basic SMO after incorporating Quadratic Approximation (QA) operator in it. The objective of this modification is to improve the local search ability of basic SMO. QA has been applied to make efficient use of the available information about the current global leader and local leaders. The neighbourhood of these solutions have been searched for better solutions using QA operator. Investigation has been made on the performance of the proposed algorithm by testing it over a set of 46 benchmark problems and results are compared with basic SMO. The chapter is organised as follows: In section 5.1, motivation behind using QA in SMO is presented. In section 5.2, the proposed algorithm is presented. In section 5.3, experimental setup has been provided. In section 5.4, experimental results have been discussed. In section 5.5, comparison has been made among SMO, TS-SMO and QASMO. Finally, the chapter has been concluded in section 5.6.

## 5.1 MOTIVATION

The motivation for applying QA in SMO comes from its successful incorporation in various metaheuristic algorithms like CR (Controlled Random Search), GA (Genetic Algorithm), PSO (Particle Swarm Optimization), DE (Differential Evaluation) etc. in past few years. The same operator is used with the name Quadratic Approximation (QA) in some papers and with Quadratic Interpolation (QI) in others available in the literature. In the present chapter, it has been used with the name Quadratic Approximation (QA) operator. QA has been implemented in different metaheuristics in different ways. In Mohan and Shanker [126], QA has been used in the local phase of random search technique to replace worst feasible solution of the population. In Pant et al. [137], QA has been used as a nonlinear crossover operator to

The content of this chapter contributes to the paper entitled:

**Gupta, K**., Deep, K., & Bansal, J. C. (2017). Improving the local search ability of spider monkey optimization algorithm using quadratic approximation for unconstrained optimization. *Computational Intelligence*. (**IF. 0.964)**, *33*(2), 210-240.

produce an offspring from three feasible solutions say two randomly selected particles and best particle of the swarm. In Deep and Das [38], QA has been used as an additional operator in GA. After every cycle of GA operators is completed, QA operator has been used to gain local refinements. In Deep and Bansal [37], QA has been used with two variants of PSO namely PSO-W (PSO with time varying inertia weight) and PSO-C (PSO with constriction factor). In Pant et al. [136], QA named as Quadratic interpolation (QI) has been used to initialize the population in Differential Evolution algorithm. In Liu et al. [109], QI has been used with global version of orthogonal learning based PSO (QIOLPSO-G).

## 5.2 THE PROPOSED QUADRATIC APPROXIMATION BASED SPIDER MONKEY OPTIMIZATION

### 5.2.1 WORKING OF QA OPERATOR

QA has been incorporated in the basic version of SMO with the objective to improve its local search ability. QA provides the point of minima of the quadratic curve passing through three solutions. QA works in the following manner:

First, three solutions say, A $(a_1, a_2, \dots, a_D)$ with the best fitness value, B $(b_1, b_2, \dots, b_D)$ and C $(c_1, c_2, \dots, c_D)$ are randomly chosen such that A, B and C all are distinct. Then a new solution P $(p_1, p_2, \dots, p_D)$, which is the point of minima of the quadratic curve passing through A, B and C is given by:

$$p[j] = \frac{1}{2}\frac{\left(b_j^2-c_j^2\right)f(A)+\left(c_j^2-a_j^2\right)f(B)+(a_j^2-b_j^2)f(C)}{\left(b_j-c_j\right)f(A)+\left(c_j-a_j\right)f(B)+(a_j-b_j)f(C)} \quad \forall\, j = 1,2,\dots,D \tag{5.1}$$

Where, $f$ (A), $f$ (B) and $f$ (C) are the values of the objective function $f$ at A, B and C respectively.

### 5.2.2 IMPLEMENTATION STRATEGY OF QA IN SMO

QA has been implemented in Global Leader Learning Phase and Local Leader Learning Phase of the basic SMO. In Global Leader Learning Phase, the three solutions will include global leader and two randomly selected members from the swarm. In Local Leader Learning Phase, these three solutions will include the local leader of the group and two randomly selected members from the same group. The reason for selecting these two phases for incorporating QA is that in these two phases, we get updated positions of the global leader and local leaders and the probability of getting better solutions in the neighbourhood of good solutions is higher as compared to other solutions. Changes made in the Global Leader Learning Phase and Local Leader Learning Phase have been described in Modified Global Leader Learning Phase and Modified Local Leader Learning Phase respectively.

**Modified Global Leader Learning Phase**: First update the position of the global leader of the swarm. Next, find the position say $SM_{worstglobal}$ of the worst member (solution having the minimum fitness value) of the swarm. Now choose three solutions A, B and C from the swarm, where A= $GL$ (position of the global leader), B and C are the positions of two randomly chosen members of the swarm. Generate a new position using eq. (5.1). If the fitness value of the newly generated position is better than that of $SM_{worstglobal}$, then update the worst position with new one. Pseudocode for this modified phase is given in Algorithm 5.1.

**Modified Local Leader Learning Phase**: First update the position of the local leader of the $k^{th}$ group. Next, find the position say $SM_{worstlocal}$ of the worst member (solution having the minimum fitness value) of the group. Now, choose three solutions A, B and C from the group, where A= $LL_k$ (position of the local leader of the $k^{th}$ group), B and C are positions of randomly chosen members of the group. Generate a new position using eq. (5.1). If the fitness value of the newly generated position is better than that of $SM_{worstglobal}$, then update the worst position with new one. The above mentioned procedure repeats for every group. Algorithm 5.2 provides pseudocode of position update process in this phase.

Pseudocode of QASMO has been provided in Algorithm 5.3.

## 5.3 EXPERIMENTAL SETUP

In order to evaluate the performance of QASMO, the set of benchmark problems, setting of control parameters and termination criteria and performance evaluation criteria is same as mentioned in section 4.3 of chapter 4. First, the impact of incorporating QA in basic SMO has been studied by compared it with basic SMO numerically, statistically and graphically for both scalable and non-scalable problems. Then, comparison has been made among basic SMO, TS-SMO and QASMO to find out the best version of SMO among these three algorithms. This comparison will be fair if we use same parameters for all the three algorithms. This is the reason of choosing the same experimental setup for QASMO which has been used for SMO and TS-SMO in chapter 4.

```
Begin
    Update the position of the global leader in the swarm
    find $SM_{globalworst}$
        For $i$ = 1 to 1000 Do
            Select A= $GL$, B and C are positions of randomly chosen members of the
            swarm such that A, B, C all are distinct
            //generate $P$ using equation (5.1)
            If (fitness($P$) > fitness($SM_{globalworst}$)) Then
                $SM_{globalworst} = P$
                Terminate the loop
            End if
        End For
        If (fitness($P$) > fitness($GL$))
            $GL = P$
        End If
        If (position of global leader is updated from previous position) Then
            $GLC = 0$
        Else
            $GLC = GLC + 1$
        End If
End
```

**Algorithm 5.1:** Modified Global Leader Learning Phase

## 5.4 DISCUSSION OF EXPERIMENTAL RESULTS

In this section, the impact of incorporating QA in basic SMO has been analyzed by comparing the performance of QASMO with basic SMO. Tables 5.1-5.4 present the numerical and statistical results of scalable problems for 30 dimensions. Tables 5.5-5.8 present the numerical and statistical results for non-scalable problems. Better entries in the tables appear in bold.

```
Begin
    For $k$ = 1 to $NG$ Do
        Update the position of the local leader in the swarm
        Find $SM_{localworst}$
        For $i$ = 1 to 1000 Do
            //Select A = $LL_k$ , B and C are positions of randomly chosen members of the groups
            such that A, B, C are all distinct
            //generate $P$ using eq. (5.1)
            If (fitness($P$) > fitness($SM_{localworst}$)) Then
                $SM_{localworst}$= $P$
                Terminate the loop
            End If
            End For
            If (fitness($P$) > fitness($LL_k$)) Then
             $LL_k$=$P$
            End If
            If (position of local leader is updated from previous position) Then
             $LLC_k$ = 0
            Else
                $LLC_k$ = $LLC_k$+ 1
             End If
    End For
 End
```

**Algorithm 5.2**: Modified Local Leader Learning Phase

### 5.4.1 DISCUSSION OF RESULTS FOR SCALABLE PROBLEMS

Table 5.1 presents the success rate and average number of function evaluations for successful runs. From this table, it can be observed that out of thirty problems, SMO and QASMO have 100 percent success rate in twenty two (No.'s 1, 2, 3, 5, 6, 7, 8, 9, 10, 12, 13, 14, 16, 18, 19, 23, 24, 25, 26, 27, 29, 30) and twenty four problems (No.'s 1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 16, 18, 19, 22, 23, 24, 25, 26, 27, 29, 30) respectively. In three problems (No.'s 11, 21 and 22), QASMO performs strictly better than SMO according to criterion 1 (case 1). Out of twenty two problems where both the algorithms have 100 percent success rate, QASMO performs better than SMO in seventeen problems (No.'s 1, 2, 6, 7, 9, 10, 11, 12, 13, 14, 16, 18, 19, 23, 24, 26, 27, 30) according to criterion 1 (case 2). Also, it can be observed from Table 5.1 that QASMO has better success rate than SMO in two problems (No.'s 21 and 22) though it comes at the cost of more number of function evaluations. Five problems (No.'s 4, 15, 17, 20, and 28) are not solved by any of the two algorithms.

```
Begin
    Initialize the swarm using equation (2.1)
    Initialize LLlt, GLlt, Pr, MG
    Iteration = 0
    Calculate fitness value of the position of each spider monkey in the swarm
    Select Global Leader and Local Leaders by applying Greedy Selection
        While (termination criterion is not satisfied) Do
            //Local Leader Phase (Algorithm 2.1)
            //Calculate Probability of each spider monkey using eq. (2.3)
            //Global Leader Phase (Algorithm 2.2)
            //Modified Global Leader Learning Phase (Algorithm 5.1)
            //Modified Local Leader Learning Phase (Algorithm 5.2)
            //Local Leader Decision Phase (Algorithm 2.5)
            //Global Leader Decision Phase (Algorithm 2.6)
            Iteration = Iteration +1
        End While
End
```

**Algorithm 5.3:** Pseudocode for QASMO

Table 5.2 provides t-test results for the average number of function evaluations of successful runs for the problems where both the algorithms identical success rate. T-test results convey that there is a significant difference in the average number of function evaluations in twenty one problems (No.'s 1, 2, 3, 5, 6, 7, 8, 9, 10, 12, 13, 14, 16, 18, 19, 23, 24, 26, 27, 29, 30) out of twenty two problems. QASMO performs significantly better than SMO in terms of average number of function evaluations for successful runs in seventeen problems (No.’s 1, 2, 6, 7, 9, 10, 12, 13, 14, 16, 18, 19, 23, 24, 26, 27, 30). There are four problems (3, 5, 8, 29) where the performance of QASMO is significantly worse than SMO

To check the solution quality obtained by both the algorithms, best, average and worst of the error values obtained in 100 independent runs have been provided in Table 5.3. For best error values, QASMO is better than SMO in eighteen problems (No.’s 3, 4, 5, 7, 9, 11, 13, 15, 16, 18, 20, 21, 22, 24, 25, 26, 27, 28) and in problem no. 19, both the algorithms have same error value. For average error values, QASMO is better than SMO in twenty one problems (No.’s 1, 2, 3, 4, 5, 6, 7, 8, 9,10, 11, 15, 18, 20, 22, 23, 24, 25, 26, 27, 28, 29, 30) and there are three problems(1, 3, 19) where both QASMO and SMO have same average error. For worst error values, QASMO is better than SMO in thirteen problems (No.’s 1, 2, 4, 8, 11, 13, 15, 16, 17, 22, 25, 27, 28) and both the algorithms have same worst error value on eight problems (No.’s 3, 5, 6, 10, 14, 19, 24, 26). QASMO is strictly better than SMO in eight problems (No.’s 4, 11, 15, 22, 25, 26, 27 and 28) according to criterion 2 (case 1). There are three problems (No.’s 3, 5, and 24) where QASMO performs better than SMO according to criterion 2 (case 2). Both the algorithms perform identically on problem no. 19 according to criterion 2 (case 3).

From Tables 5.1 and 5.3, it can be concluded that in five problems (No.’s 4, 15, 17, 20, 28) where both the algorithms fail to solve the problem (zero percent success rate), average error value obtained by QASMO is better than the average error value obtained by SMO except problem no. 17. Also, it can be seen for problem no. 19, global optima is obtained by both the algorithms.

Table 5.4 shows that average execution time taken (in seconds) by both the algorithms per run for solving scalable problems of 30 dimensions. From this table, it can be observed that the time taken by both the algorithms is very small and there is not much difference in the execution time on most of the problems. But there are some problems (No.’s 15, 17, 20, 21, 28), where there is much difference between the execution time taken by both the algorithms. These are the problems where success rate is zero or very low. On these problems, QASMO takes much more time than SMO. The reason for this is the iterative loop inside the Modified Global Leader Learning Phase and Modified Local Leader Learning Phase in QASMO. So, each iteration in QASMO is taking more time than those in SMO.

The performance index graph for scalable problems is provided in Figure 5.1 which shows SMO performs relatively better than QASMO when the weight varies from 0 to 0.6 while QASMO performs better when the weight is greater than or equal to 0.6. Also, it can be concluded from the graph that SMO performs better when priority is given to function evaluations over success rate while QASMO is preferred when priority is given to the success rate over function evaluations. Also, the graph illustrates that the PI value increase when the more weight is given to the success rate as compared to function evaluations. The convergence graphs for all the scalable problems have been provided in Figures 5.2-5.5.

### 5.4.2 DISCUSSION OF RESULTS FOR NON-SCALABLE PROBLEMS

From Table 5.5, it can be seen that success rate of SMO is either identical or better than QASMO. SMO and QASMO have 100 percent success rate in eleven (No.’s 31, 34, 35, 36, 37, 38, 39, 40, 42, 44, 46) and eight (No.’s 31, 34, 35, 36, 38, 39, 40, 46) problems respectively. In four problems (No.’s 32, 37, 42 and 44), SMO is strictly better than QASMO according to criterion 1(case 1). Both the algorithms failed to solve four problems (No.’s 33, 41, 43, and 45) in all the runs. QASMO performs better than SMO over seven problems according to criterion 1(case 2).

Table 5.6 contains t-test results of average number of function evaluations. It can be seen from the table that out of eight problems, where both the algorithms have identical success rate, QASMO performs significantly better than SMO over six problems (No.’s 31, 34, 35, 36, 40, 46).

Best, average and worst of the error values obtained by both the algorithms have been listed in Table 5.7. For best values, QASMO is better than SMO in seven problems (No.’s 31, 33, 34, 35, 37, 38, 40), both the algorithms have same best error values in four problems (39, 41, 43, 45) and SMO is better than QASMO in five problems (No.’s 32, 36, 42, 44, 46). For average values, QASMO is better than SMO in seven problems (No.’s 31, 33, 35, 36, 38, 39, 40), same error value in problem no. 45 and SMO is better than QASMO in eight problems (No.’s 32, 34, 37, 41, 42, 43, 44, 46). For worst error values, QASMO outperforms SMO in six problems (No.’s 31, 34, 35, 39, 40, 46) and SMO outperforms QASMO in eight problems (No.’s 33, 36, 37, 38, 41, 42, 43, 44). Both the algorithms have same worst error values in two problems. In three problems (No.’s 31, 35 and 40), QASMO performs strictly better than SMO according to criterion 2 (case 1).

From Table 5.5 and Table 5.7, it can be concluded that out of four problems (No.'s 33, 41,43 and 45), where both SMO and QASMO has zero percent success rate, average error value obtained by SMO is better than that obtained by QASMO in two problems (No.'s 41and 43). No clear conclusion can be drawn about the performance of two algorithms on non-scalable problems considered in the benchmark set. In some cases, SMO is performing better, while in other, QASMO is better.

Table 5.8 shows that average execution time taken (in seconds) by both the algorithms per run for solving non-scalable problems. From this table, it can be observed that like scalable problems, the time taken by both the algorithms on non-scalable problems is very small and there is not much difference in the execution time on most of the problems. But there are some problems (No.'s 32, 41, 42, 43, 45), where there is much difference between the execution time taken by both the algorithms. These are the problems where success rate is zero or low except problem no. 42. On these problems, QASMO takes much more time than SMO. The reason for this is again the iterative loop inside the Modified Global Leader Learning Phase and Modified Local Leader Learning phase in QASMO. So, each iteration in QASMO is taking more time than in SMO.

The performance index graph for non-scalable problems is provided in Figure 5.6 which shows that the performance of SMO is better than QASMO for all the values of w. Also, it can be observed that the performance index value increases with the increase in the weight given to the success rate. The convergence graphs for all the non-scalable problems have been provided in Figures 5.7-5.8.

From the above discussion on the results of both scalable and non-scalable problems, it can be concluded that though QASMO is showing good performance on scalable problems (listed in benchmark set) in terms of reliability, efficiency and accuracy as compared to SMO, its performance is reasonable on non-scalable problems. Incorporation of quadratic approximation operator has positive impact on the performance of SMO.

## 5.5 COMPARISON AMONG SMO, TS-SMO AND QASMO

In this section, the comparison has been made among basic SMO, TS-SMO and QASMO to find the best version of SMO on the set of benchmark problems given in Appendix III. TS-SMO and QASMO are proposed to improve the exploration and exploitation respectively in basic SMO. Discussion of results in chapters 4 and 5 show that they both are performing better than basic SMO. In this section, a comparison has been made among SMO, TS-SMO and QASMO. Numerical results have been shown in Tables 5.9- 5.16. Table 5.17 provides the summary of comparison and it is clear that QASMO is better than SMO and TS-SMO on most of the scalable problems on all the performance metrics used for evaluating the performance. On non-scalable problems, results are in the favour of all the three.

## 5.6 CONCLUSIONS

In this chapter, a modified version of basic SMO is proposed. Local search ability of SMO is improved by incorporating QA operator in it which is definitely helpful for SMO to explore the surrounding regions of current global and local leaders. To check the robustness of proposed algorithm, its performance has been tested over a benchmark set of 46 problems including both scalable and non-scalable problems. Though results are showing improvement in terms of function evaluations, success and quality of the solution attained on scalable problems, it is performing moderately on non-scalable problems. Also, the comparison has been made among SMO, TS-SMO and QASMO. Results show that QSMO is better than SMO and TS-SMO on scalable problems. For non-scalable problems, the conclusion is not inclined towards a single algorithm. In future, other modifications can be proposed to improve the exploration and exploitation of SMO.

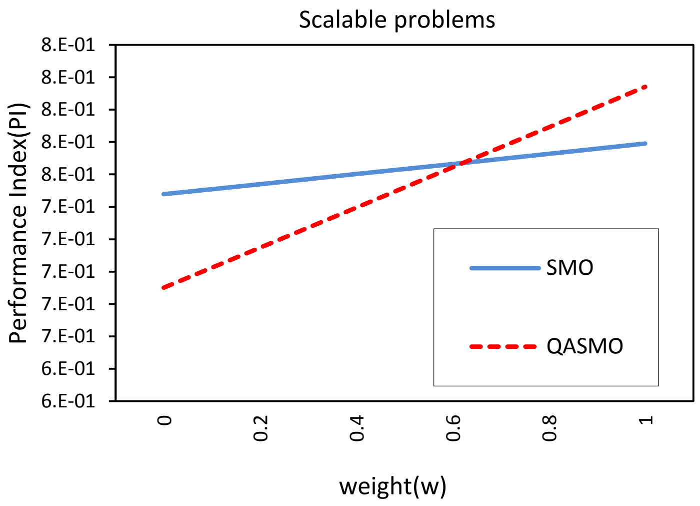

**Figure 5.1**: Performance Index graph for Scalable problems for success rate versus average number of function evaluations for successful runs

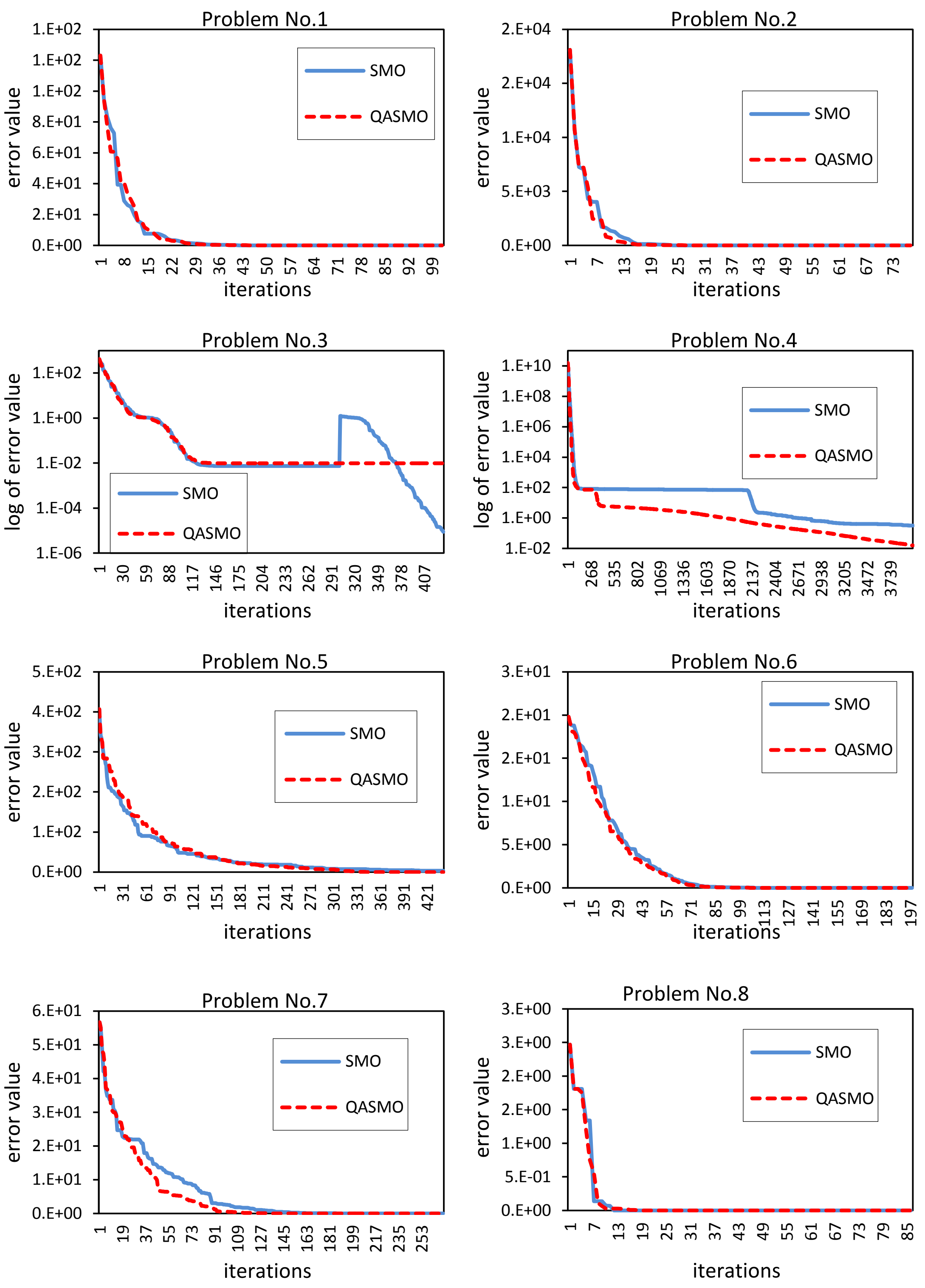

**Figure 5.2**: Convergence graphs for Problem No.1-Problem No.8

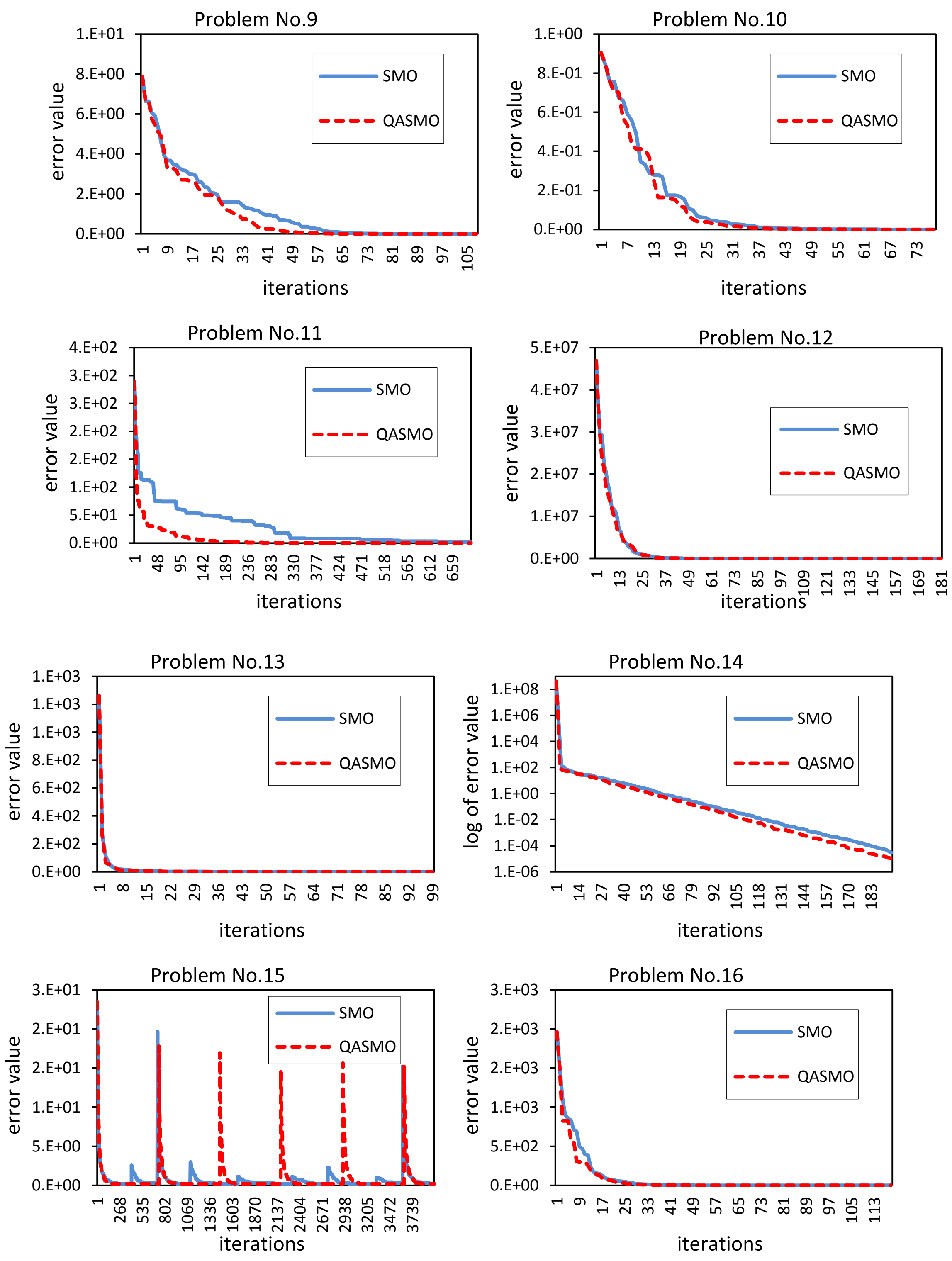

**Figure 5.3:** Convergence graphs for Problem No.9-Problem No.16

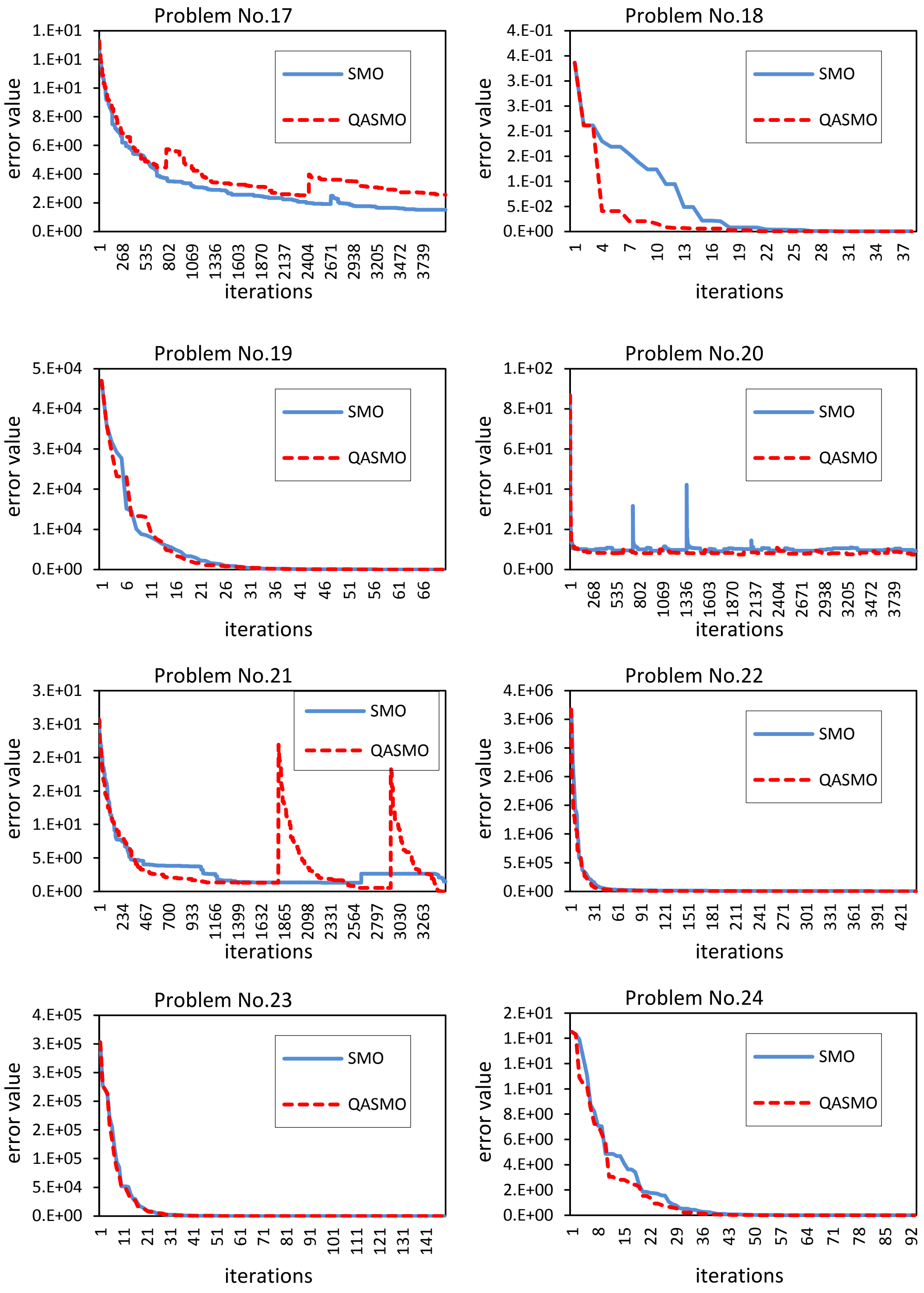

**Figure 5.4:** Convergence graphs for Problem No.17-Problem No.24

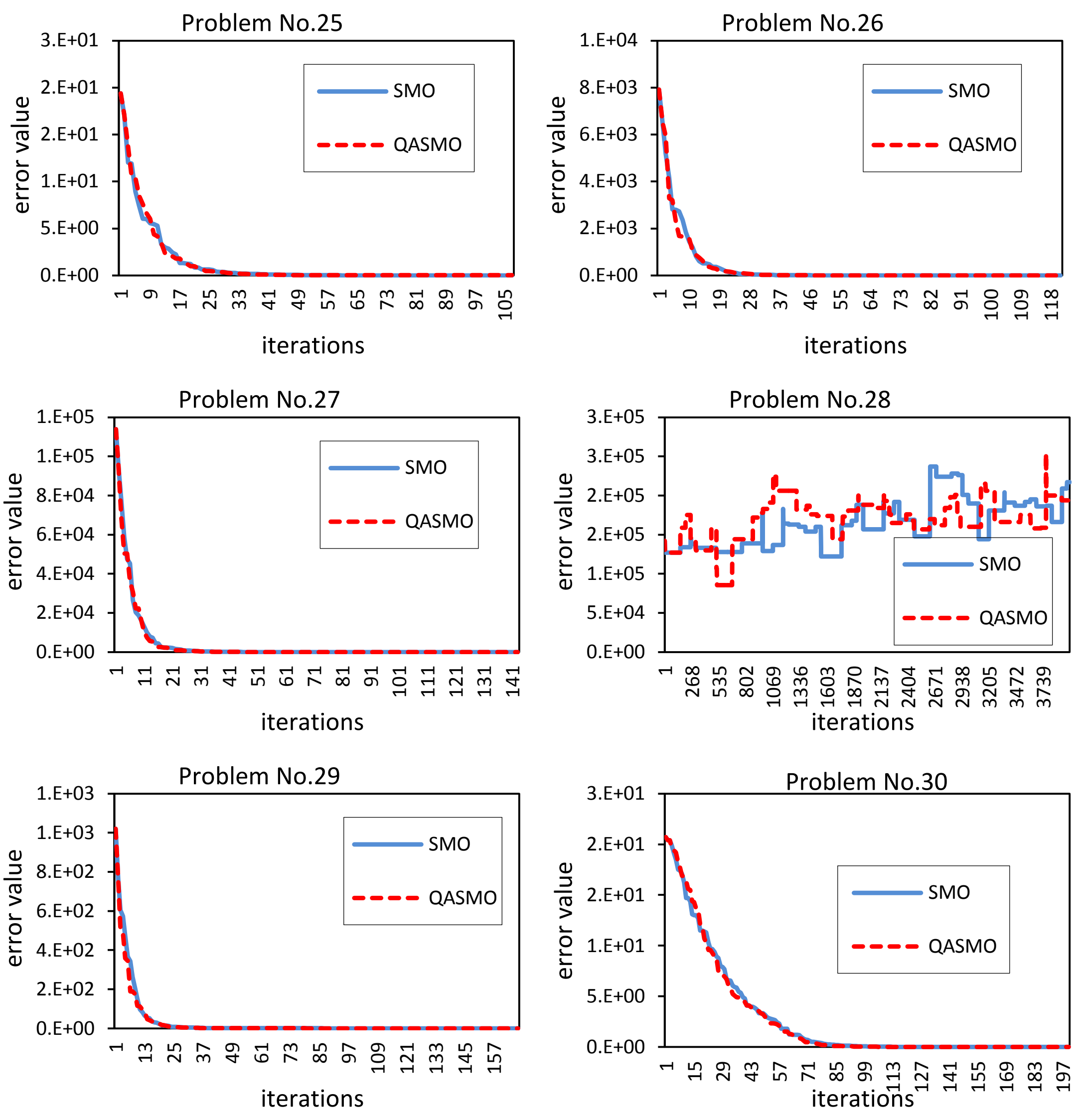

**Figure 5.5:** Convergence graphs for Problem No.25-Problem No.30

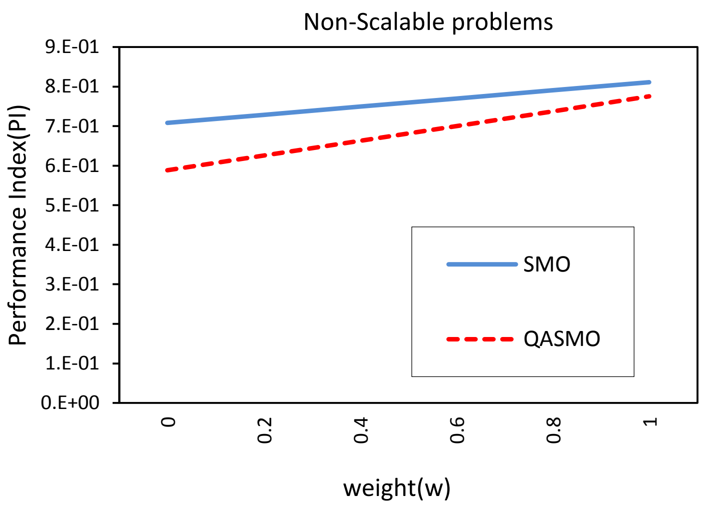

**Figure 5.6**: Performance Index graph for Non-Scalable problems for success rate versus average number of function evaluations for successful runs

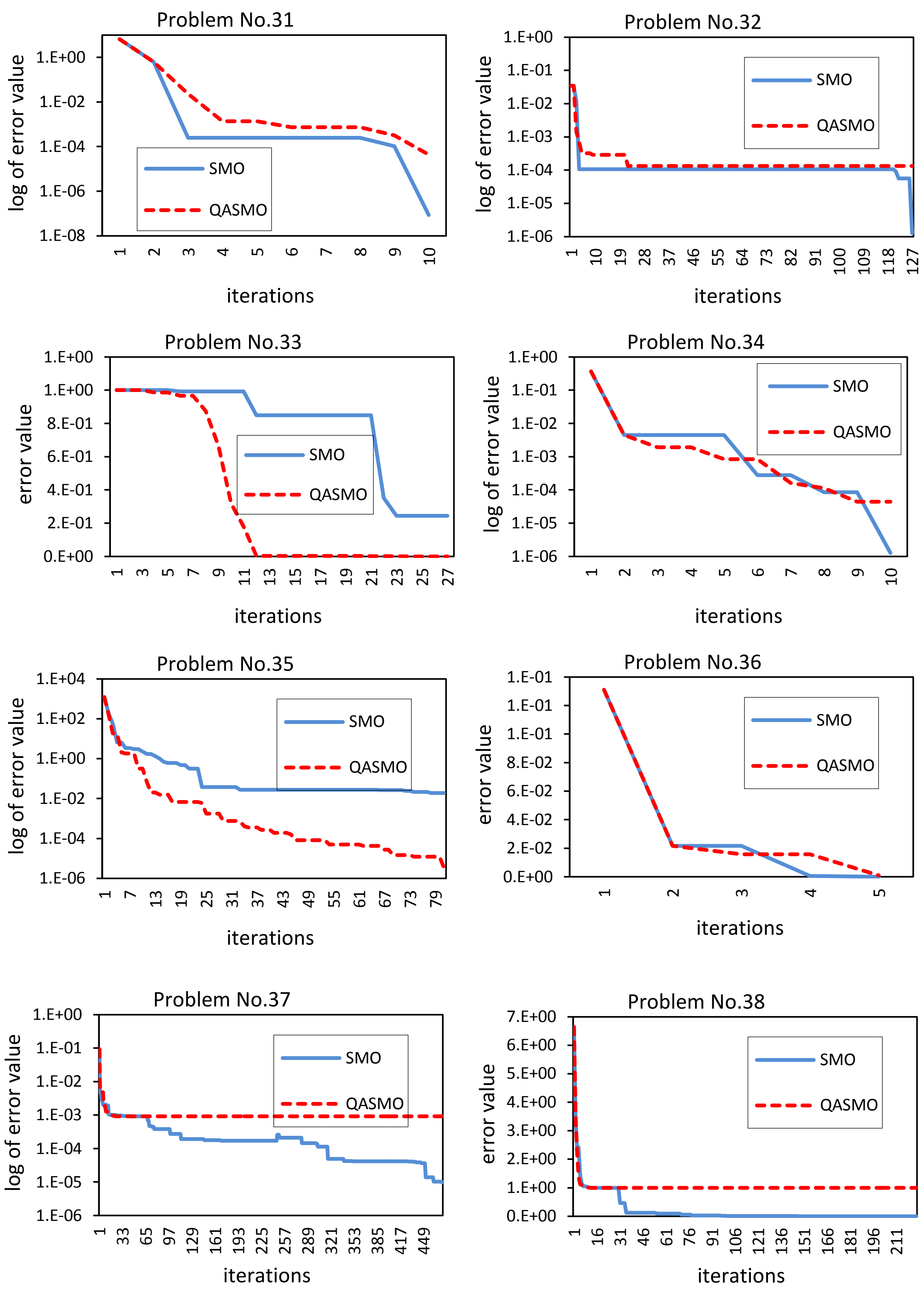

**Figure 5.7:** Convergence graphs for Problem No.31-Problem No.38

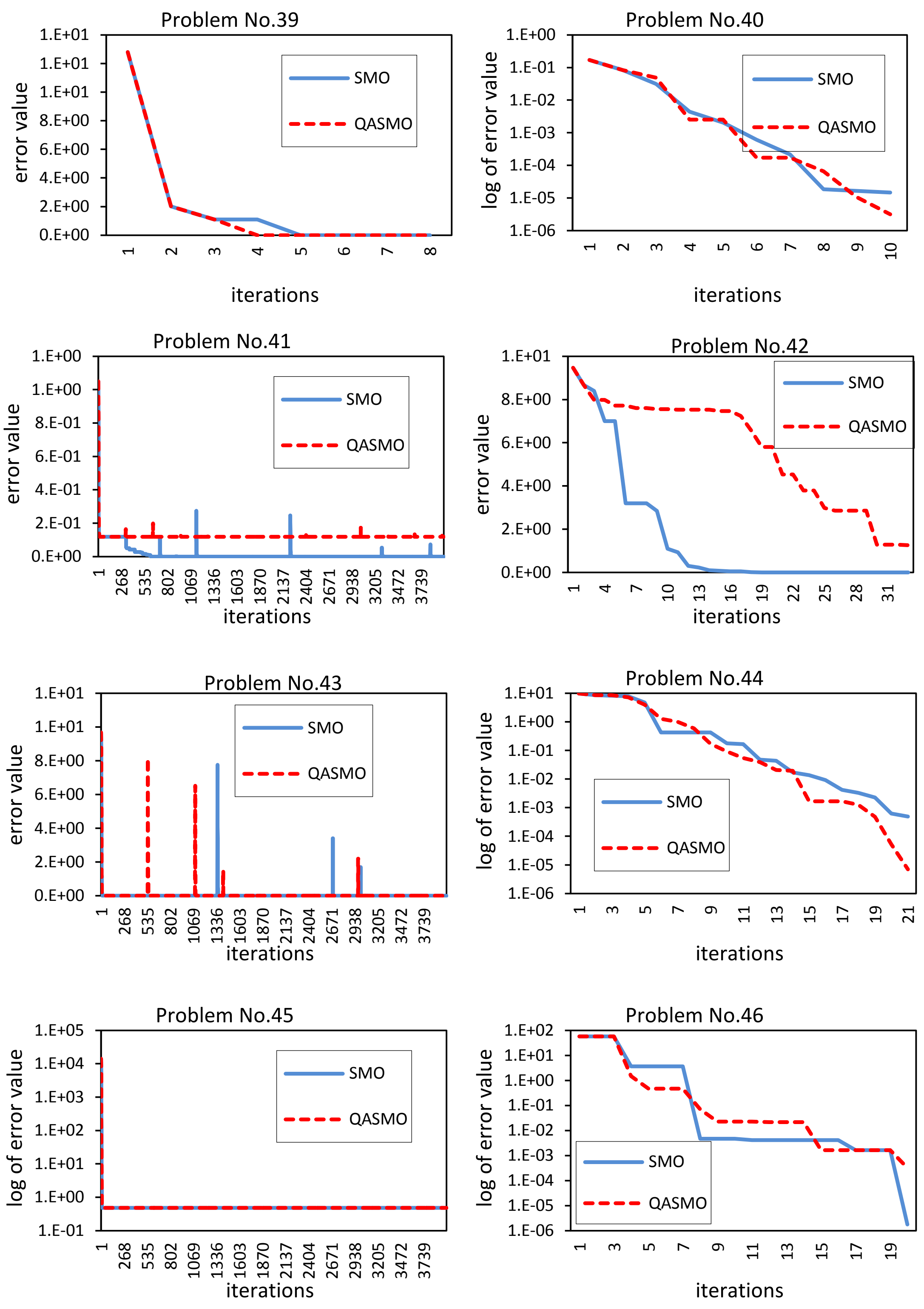

**Figure 5.8:** Convergence graphs for Problem No.39-Problem No.46

**Table 5.1**: Success Rate and Average number of function evaluations of successful runs for SMO and QASMO (Scalable Problems-30 dimensions)

| Problem No. | Success Rate | | Average number of function evaluations for successful runs | |
|---|---|---|---|---|
| | SMO | QASMO | SMO | QASMO |
| 1 | 100 | 100 | 33192 | **30626** |
| 2 | 100 | 100 | 26566 | **23254** |
| 3 | 100 | 100 | **103625** | 407833 |
| 4 | 0 | 0 | N.A. | N.A. |
| 5 | 100 | 100 | **229495** | 1176179 |
| 6 | 100 | 100 | 63917 | **59159** |
| 7 | 100 | 100 | 189827 | **85764** |
| 8 | 100 | 100 | **24027** | 56411 |
| 9 | 100 | 100 | 37701 | **33967** |
| 10 | 100 | 100 | 25421 | **23371** |
| 11 | 47 | **100** | 650570 | **218639** |
| 12 | 100 | 100 | 58502 | **53700** |
| 13 | 100 | 100 | 32220 | **29952** |
| 14 | 100 | 100 | 62664 | **58450** |
| 15 | 0 | 0 | N.A. | N.A. |
| 16 | 100 | 100 | 38287 | **35335** |
| 17 | 0 | 0 | N.A. | N.A. |
| 18 | 100 | 100 | 14002 | **11860** |
| 19 | 100 | 100 | 22628 | **20988** |
| 20 | 0 | 0 | N.A. | N.A. |
| 21 | 7 | **42** | **1051267** | 5794852 |
| 22 | 83 | **100** | **354061** | 764419 |
| 23 | 100 | 100 | 48381 | **44862** |
| 24 | 100 | 100 | 30839 | **27865** |
| 25 | 100 | 100 | **30695** | 30961 |
| 26 | 100 | 100 | 39151 | **36457** |
| 27 | 100 | 100 | 44665 | **41482** |
| 28 | 0 | 0 | N.A. | N.A. |
| 29 | 100 | 100 | **106373** | 563615 |
| 30 | 100 | 100 | 63980 | **59646** |

**Table 5.2**: T-test results for problems having identical success rate for SMO and QASMO (Scalable Problems-30 dimensions)

| Problem No. | sign | Problem No. | sign |
|---|---|---|---|
| 1 | + | 14 | + |
| 2 | + | 16 | + |
| 3 | - | 18 | + |
| 5 | - | 19 | + |
| 6 | + | 23 | + |
| 7 | + | 24 | + |
| 8 | - | 25 | = |
| 9 | + | 26 | + |
| 10 | + | 27 | + |
| 12 | + | 29 | - |
| 13 | + | 30 | + |

**Table 5.3**: Best, average and worst of error values obtained in 100 independent runs (Scalable Problems-30dimensions)

| Problem No. | Best | | Average | | Worst | |
|---|---|---|---|---|---|---|
| | SMO | QASMO | SMO | QASMO | SMO | QASMO |
| 1 | **6.01E-06** | 6.56E-06 | 8.72E-06 | 8.72E-06 | 9.99E-06 | **9.98E-06** |
| 2 | **3.12E-06** | 4.91E-06 | 8.03E-06 | **7.85E-06** | 9.99E-06 | **9.83E-06** |
| 3 | 3.70E-06 | **2.06E-18** | 8.52E-06 | 8.52E-06 | 9.99E-06 | 9.99E-06 |
| 4 | 1.57E-02 | **1.74E-03** | 1.63E+01 | **7.90E+00** | 8.08E+01 | **2.08E+01** |
| 5 | 2.17E-06 | **1.14E-13** | 8.42E-06 | **7.36E-06** | 9.97E-06 | 9.97E-06 |
| 6 | **7.58E-06** | 7.92E-06 | 9.43E-06 | **9.28E-06** | 9.98E-06 | 9.98E-06 |
| 7 | 7.68E-06 | **7.32E-06** | 9.36E-06 | **9.26E-06** | **9.99E-06** | 1.00E-05 |
| 8 | **7.54E-09** | 1.46E-07 | 5.11E-06 | **4.99E-06** | 9.98E-06 | **9.97E-06** |
| 9 | 6.42E-06 | **4.44E-16** | 8.92E-06 | **8.41E-06** | **9.99E-06** | 1.00E-05 |
| 10 | **5.15E-06** | 6.30E-06 | 8.85E-06 | **8.70E-06** | 1.00E-05 | 1.00E-05 |
| 11 | 7.36E-06 | **7.24E-06** | 4.17E+00 | **9.32E-06** | 4.62E+01 | **1.00E-05** |
| 12 | **5.62E-06** | 5.97E-06 | **8.74E-06** | 8.81E-06 | **9.98E-06** | 9.99E-06 |
| 13 | 5.87E-06 | **5.76E-06** | **8.71E-06** | 8.81E-06 | 9.99E-06 | **9.94E-06** |
| 14 | **7.44E-06** | 7.52E-06 | **9.30E-06** | 9.33E-06 | 9.99E-06 | 9.99E-06 |
| 15 | 2.00E-01 | **9.99E-02** | 5.85E-01 | **4.84E-01** | 1.05E+01 | **4.11E+00** |
| 16 | 5.36E-06 | **5.01E-06** | **8.63E-06** | 8.77E-06 | 9.99E-06 | **9.98E-06** |
| 17 | **9.38E-03** | 5.03E-01 | **1.49E+00** | 2.10E+00 | 8.67E+00 | **7.75E+00** |
| 18 | 2.74E-06 | **1.34E-06** | 7.60E-06 | **6.77E-06** | **9.97E-06** | 9.98E-06 |
| 19 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 |
| 20 | 7.73E+00 | **6.86E+00** | 9.73E+00 | **8.59E+00** | **1.22E+01** | 1.43E+01 |
| 21 | 7.78E-06 | **0.00E+00** | **1.34E+00** | 2.10E+00 | **4.73E+00** | 1.61E+01 |
| 22 | 7.58E-08 | **3.63E-08** | 1.54E+02 | **4.89E-06** | 1.45E+04 | **9.89E-06** |
| 23 | **6.17E-06** | 6.21E-06 | 8.60E-06 | **8.44E-06** | **9.93E-06** | 9.98E-06 |
| 24 | 6.17E-06 | **4.38E-06** | 8.77E-06 | **8.74E-06** | 9.99E-06 | 9.99E-06 |
| 25 | 5.98E-06 | **7.29E-08** | 8.78E-06 | **8.50E-06** | 1.00E-05 | **9.98E-06** |
| 26 | 5.97E-06 | **4.77E-06** | 8.80E-06 | **8.68E-06** | 9.99E-06 | 9.99E-06 |
| 27 | 5.67E-06 | **5.47E-06** | 8.79E-06 | **8.51E-06** | 1.00E-05 | **9.99E-06** |
| 28 | 1.24E+05 | **1.20E+05** | 1.91E+05 | **1.86E+05** | 2.41E+05 | **2.40E+05** |
| 29 | **3.94E-06** | 4.80E-06 | 8.67E-06 | **8.66E-06** | **9.97E-06** | 9.99E-06 |
| 30 | **7.83E-06** | 7.89E-06 | 9.42E-06 | **9.33E-06** | **9.99E-06** | 1.00E-05 |

**Table 5.4**: Average execution time per run (in seconds) (Scalable problems-30 dimensions)

| Problem No. | SMO | QASMO | Problem No. | SMO | QASMO |
|---|---|---|---|---|---|
| 1 | 0.06337 | 0.06223 | 16 | 0.07646 | 0.07397 |
| 2 | 0.05144 | 0.04986 | 17 | 106.9177 | 1139.253 |
| 3 | 0.63335 | 2.80771 | 18 | 0.54852 | 0.42603 |
| 4 | 2.20541 | 2.48634 | 19 | 0.2067 | 0.17335 |
| 5 | 0.95737 | 5.78111 | 20 | 56.79807 | 360.9763 |
| 6 | 0.27407 | 0.30021 | 21 | 12.82866 | 51.45583 |
| 7 | 0.74151 | 0.40411 | 22 | 22.34735 | 30.18972 |
| 8 | 1.80296 | 4.59815 | 23 | 0.25422 | 0.20357 |
| 9 | 0.15978 | 0.168 | 24 | 0.29362 | 0.21724 |
| 10 | 0.04684 | 0.04887 | 25 | 0.29653 | 0.23696 |
| 11 | 1.90559 | 0.46502 | 26 | 1.66475 | 1.39955 |
| 12 | 0.10578 | 0.10896 | 27 | 0.13941 | 0.12459 |
| 13 | 2.10971 | 2.08317 | 28 | 7.6486 | 500.9501 |
| 14 | 0.11883 | 0.12553 | 29 | 1.01688 | 6.34944 |
| 15 | 51.89418 | 139.093 | 30 | 0.45603 | 0.41899 |

**Table 5.5**: Success Rate and Average number of function evaluations of successful runs for SMO and QASMO (Non-Scalable problems)

| Problem No. | Success Rate | | Average number of function evaluations of successful runs | |
|---|---|---|---|---|
| | SMO | QASMO | SMO | QASMO |
| 31 | 100 | 100 | 3738 | **3492** |
| 32 | **98** | 49 | **430014** | 4374449 |
| 33 | 0 | 0 | N.A. | N.A. |
| 34 | 100 | 100 | 3585 | **3200** |
| 35 | 100 | 100 | 256584 | **55756** |
| 36 | 100 | 100 | 3304 | **3050** |
| 37 | **100** | 96 | **131394** | 180988 |
| 38 | 100 | 100 | **26119** | 542972 |
| 39 | 100 | 100 | 2009 | **1936** |
| 40 | 100 | 100 | 3071 | **2913** |
| 41 | 0 | 0 | N.A. | N.A. |
| 42 | **100** | 96 | **24508** | 317022 |
| 43 | 0 | 0 | N.A. | N.A. |
| 44 | **100** | 99 | **13458** | 29208 |
| 45 | 0 | 0 | N.A. | N.A. |
| 46 | 100 | 100 | 10737 | **8753** |

**Table 5.6**: T-test results for problems having identical successful rate for SMO and QASMO (Non-Scalable problems)

| Problem No. | sign | Problem No. | sign |
|---|---|---|---|
| 31 | + | 38 | - |
| 34 | + | 39 | = |
| 35 | + | 40 | + |
| 36 | + | 46 | + |

**Table 5.7**: Best, average and worst of error values obtained in 100 independent runs (Non-Scalable problems)

| Problem No. | Best | | Average | | Worst | |
|---|---|---|---|---|---|---|
| | SMO | QASMO | SMO | QASMO | SMO | QASMO |
| 31 | 8.53E-08 | **1.86E-08** | 4.82E-06 | **4.38E-06** | 9.99E-06 | **9.91E-06** |
| 32 | **6.79E-08** | 2.40E-07 | **1.11E-05** | 1.39E-04 | 2.87E-04 | **2.87E-04** |
| 33 | 1.21E-07 | **1.04E-08** | 4.87E-06 | **4.17E-06** | **9.48E-06** | 9.92E-06 |
| 34 | 1.74E-07 | **3.97E-08** | **4.35E-06** | 4.61E-06 | 9.98E-06 | **9.86E-06** |
| 35 | 1.22E-06 | **8.66E-07** | 7.24E-06 | **6.92E-06** | 1.00E-05 | **9.96E-06** |
| 36 | **9.26E-08** | 1.46E-07 | 4.37E-06 | **3.97E-06** | **9.75E-06** | 9.91E-06 |
| 37 | 2.41E-06 | **1.21E-06** | **7.95E-06** | 4.41E-05 | **9.99E-06** | 9.16E-04 |
| 38 | 1.42E-06 | **4.66E-07** | 6.25E-06 | **6.24E-06** | **9.91E-06** | 9.95E-06 |
| 39 | 3.75E-06 | 3.75E-06 | 4.99E-06 | **4.83E-06** | 9.96E-06 | **9.94E-06** |
| 40 | 1.88E-08 | **3.74E-09** | 4.26E-06 | **3.32E-06** | 9.94E-06 | **9.89E-06** |
| 41 | 3.75E-04 | 3.75E-04 | **3.88E-04** | 5.65E-02 | **1.28E-03** | 1.52E-01 |
| 42 | **1.01E-06** | 1.55E-06 | **5.77E-06** | 2.02E-01 | **9.88E-06** | 5.05E+00 |
| 43 | 5.96E-05 | 5.96E-05 | **5.96E-05** | 1.05E-01 | **5.96E-05** | 5.27E+00 |
| 44 | **1.74E-08** | 4.54E-08 | **4.48E-06** | 5.36E-02 | **9.92E-06** | 5.36E+00 |
| 45 | 4.82E-01 | 4.82E-01 | 4.82E-01 | 4.82E-01 | 4.82E-01 | 4.82E-01 |
| 46 | **5.60E-08** | 2.20E-07 | **3.65E-06** | 4.75E-06 | 9.70E-06 | **9.57E-06** |

**Table 5.8:** Average execution time per run (in seconds) (Non-Scalable problems)

| Problem No. | SMO | QASMO | Problem No. | SMO | QASMO |
|---|---|---|---|---|---|
| 31 | 0.01315 | 0.00954 | 39 | 0.00136 | 0.00093 |
| 32 | 1.32419 | 17.22715 | 40 | 0.00194 | 0.00062 |
| 33 | 0.06098 | 0.03591 | 41 | 1.20498 | 68.24806 |
| 34 | 0.01496 | 0.01309 | 42 | 0.58038 | 11.56375 |
| 35 | 2.09645 | 0.43897 | 43 | 47.63118 | 252.6167 |
| 36 | 0.00553 | 0.00502 | 44 | 0.70029 | 4.8148 |
| 37 | 1.95449 | 4.48995 | 45 | 14.31415 | 114.0719 |
| 38 | 0.00934 | 1.06005 | 46 | 0.01287 | 0.01589 |

**Table 5.9**: Success Rate and Average number of function evaluations for successful runs of SMO, TS-SMO and QASMO (Scalable problems-30 dimensions)

| Problem No. | Success Rate | | | Average number of function evaluations for successful runs | | |
|---|---|---|---|---|---|---|
| | SMO | TSMO | QASMO | SMO | TSMO | QASMO |
| 1 | 100 | 100 | 100 | 33192 | 32128 | **30626** |
| 2 | 100 | 100 | 100 | 26566 | 25923 | **23254** |
| 3 | 100 | 100 | 100 | **103625** | 108601 | 407833 |
| 4 | 0 | 0 | 0 | N.A. | N.A. | N.A. |
| 5 | 100 | 100 | 100 | **229495** | 254951 | 1176179 |
| 6 | 100 | 100 | 100 | 63917 | 61660 | **59159** |
| 7 | 100 | 100 | 100 | 189827 | 149630 | **85764** |
| 8 | 100 | 100 | 100 | 24027 | **19040** | 56411 |
| 9 | 100 | 100 | 100 | 37701 | 34987 | **33967** |
| 10 | 100 | 100 | 100 | 25421 | 24404 | **23371** |
| 11 | 47 | 61 | **100** | 650570 | 682431 | **218639** |
| 12 | 100 | 100 | 100 | 58502 | 56807 | **53700** |
| 13 | 100 | 100 | 100 | 32220 | 31231 | **29952** |
| 14 | 100 | 100 | 100 | 62664 | 60888 | **58450** |
| 15 | 0 | 0 | 0 | N.A. | N.A. | N.A. |
| 16 | 100 | 100 | 100 | 38287 | 37013 | **35335** |
| 17 | 0 | 0 | 0 | N.A. | N.A. | N.A. |
| 18 | 100 | 100 | 100 | 14002 | 13294 | **11860** |
| 19 | 100 | 100 | 100 | 22628 | 22437 | **20988** |
| 20 | 0 | 0 | 0 | N.A. | N.A. | N.A. |
| 21 | 7 | 2 | **42** | **1051267** | 1117825 | 5794852 |
| 22 | 83 | 85 | **100** | 354061 | **349233** | 764419 |
| 23 | 100 | 100 | 100 | 48381 | 47167 | **44862** |
| 24 | 100 | 100 | 100 | 30839 | 29457 | **27865** |
| 25 | 100 | 100 | 100 | 30695 | **29586** | 30961 |
| 26 | 100 | 100 | 100 | 39151 | 37892 | **36457** |
| 27 | 100 | 100 | 100 | 44665 | 43394 | **41482** |
| 28 | 0 | 0 | 0 | N.A. | N.A. | N.A. |
| 29 | 100 | 100 | 100 | 106373 | **101093** | 563615 |
| 30 | 100 | 100 | 100 | 63980 | 61938 | **59646** |

**Table 5.10**: Best of error values obtained in 100 independent runs of SMO, TS-SMO and QASMO (Scalable problems-30 dimensions)

| Problem No. | SMO | TS-SMO | QASMO |
|---|---|---|---|
| 1 | **6.01E-06** | 6.33E-06 | 6.56E-06 |
| 2 | **3.12E-06** | 4.25E-06 | 4.91E-06 |
| 3 | 3.70E-06 | 5.80E-06 | **2.06E-18** |
| 4 | 1.57E-02 | 2.36E-03 | **1.74E-03** |
| 5 | 2.17E-06 | 5.40E-06 | **1.14E-13** |
| 6 | **7.58E-06** | 7.86E-06 | 7.92E-06 |
| 7 | 7.68E-06 | 7.67E-06 | **7.32E-06** |
| 8 | 7.54E-09 | **7.12E-11** | 1.46E-07 |
| 9 | 6.42E-06 | 1.31E-07 | **4.44E-16** |
| 10 | **5.15E-06** | 5.36E-06 | 6.30E-06 |
| 11 | 7.36E-06 | **5.72E-06** | 7.24E-06 |
| 12 | **5.62E-06** | 5.72E-06 | 5.97E-06 |
| 13 | 5.87E-06 | **4.78E-06** | 5.76E-06 |
| 14 | **7.44E-06** | 7.78E-06 | 7.52E-06 |
| 15 | 2.00E-01 | 2.00E-01 | **9.99E-02** |
| 16 | 5.36E-06 | 6.02E-06 | **5.01E-06** |
| 17 | 9.38E-03 | **2.72E-04** | 5.03E-01 |
| 18 | 2.74E-06 | 1.38E-06 | **1.34E-06** |
| 19 | 0.00E+00 | 0.00E+00 | 0.00E+00 |
| 20 | 7.73E+00 | 8.11E+00 | **6.86E+00** |
| 21 | 7.78E-06 | 9.50E-06 | **0.00E+00** |
| 22 | 7.58E-08 | **2.64E-08** | 3.63E-08 |
| 23 | 6.17E-06 | **5.14E-06** | 6.21E-06 |
| 24 | 6.17E-06 | 5.34E-06 | **4.38E-06** |
| 25 | 5.98E-06 | 4.95E-06 | **7.29E-08** |
| 26 | 5.97E-06 | 5.22E-06 | **4.77E-06** |
| 27 | 5.67E-06 | 6.38E-06 | **5.47E-06** |
| 28 | 1.24E+05 | 1.22E+05 | **1.20E+05** |
| 29 | **3.94E-06** | 4.54E-06 | 4.80E-06 |
| 30 | 7.83E-06 | **7.12E-06** | 7.89E-06 |

**Table 5.11**: Average of error values obtained in 100 independent runs of SMO, TS-SMO and QASMO (Scalable problems-30 dimensions)

| Problem No. | SMO | TS-SMO | QASMO |
|---|---|---|---|
| 1 | 8.72E-06 | 8.72E-06 | 8.72E-06 |
| 2 | 8.03E-06 | 7.98E-06 | **7.85E-06** |
| 3 | **8.52E-06** | 8.73E-06 | **8.52E-06** |
| 4 | 1.63E+01 | 1.62E+01 | **7.90E+00** |
| 5 | 8.42E-06 | 9.05E-06 | **7.36E-06** |
| 6 | 9.43E-06 | 9.35E-06 | **9.28E-06** |
| 7 | 9.36E-06 | 9.47E-06 | **9.26E-06** |
| 8 | 5.11E-06 | 5.09E-06 | **4.99E-06** |
| 9 | 8.92E-06 | 8.49E-06 | **8.41E-06** |
| 10 | 8.85E-06 | **8.68E-06** | 8.70E-06 |
| 11 | 4.17E+00 | 4.73E+00 | **9.32E-06** |
| 12 | **8.74E-06** | 8.75E-06 | 8.81E-06 |
| 13 | **8.71E-06** | 8.79E-06 | 8.81E-06 |
| 14 | **9.30E-06** | 9.37E-06 | 9.33E-06 |
| 15 | 5.85E-01 | 6.02E-01 | **4.84E-01** |
| 16 | **8.63E-06** | 8.79E-06 | 8.77E-06 |
| 17 | 1.49E+00 | **1.38E+00** | 2.10E+00 |
| 18 | 7.60E-06 | 7.09E-06 | **6.77E-06** |
| 19 | 0.00E+00 | 0.00E+00 | 0.00E+00 |
| 20 | 9.73E+00 | 9.78E+00 | **8.59E+00** |
| 21 | 1.34E+00 | **1.12E+00** | 2.10E+00 |
| 22 | 1.54E+02 | 5.44E+03 | **4.89E-06** |
| 23 | 8.60E-06 | 8.71E-06 | **8.44E-06** |
| 24 | 8.77E-06 | **8.59E-06** | 8.74E-06 |
| 25 | 8.78E-06 | 8.65E-06 | **8.50E-06** |
| 26 | 8.80E-06 | 8.86E-06 | **8.68E-06** |
| 27 | 8.79E-06 | 8.67E-06 | **8.51E-06** |
| 28 | 1.91E+05 | 1.88E+05 | **1.86E+05** |
| 29 | 8.67E-06 | **8.57E-06** | 8.66E-06 |
| 30 | 9.42E-06 | **9.33E-06** | **9.33E-06** |

**Table 5.12**: Worst of error values obtained in 100 independent runs of SMO, TS-SMO and QASMO (Scalable problems-30 dimensions)

| Problem No. | SMO | TS-SMO | QASMO |
|---|---|---|---|
| 1 | 9.99E-06 | 9.99E-06 | **9.98E-06** |
| 2 | 9.99E-06 | 9.99E-06 | **9.83E-06** |
| 3 | 9.99E-06 | **9.98E-06** | 9.99E-06 |
| 4 | 8.08E+01 | 8.30E+01 | **2.08E+01** |
| 5 | **9.97E-06** | 1.00E-05 | **9.97E-06** |
| 6 | **9.98E-06** | 9.99E-06 | **9.98E-06** |
| 7 | 9.99E-06 | **9.98E-06** | 1.00E-05 |
| 8 | 9.98E-06 | 9.99E-06 | **9.97E-06** |
| 9 | **9.99E-06** | **9.99E-06** | 1.00E-05 |
| 10 | 1.00E-05 | 1.00E-05 | 1.00E-05 |
| 11 | 4.62E+01 | 5.80E+01 | **1.00E-05** |
| 12 | **9.98E-06** | 9.99E-06 | 9.99E-06 |
| 13 | 9.99E-06 | 9.99E-06 | **9.94E-06** |
| 14 | 9.99E-06 | **9.98E-06** | 9.99E-06 |
| 15 | 1.05E+01 | 9.47E+00 | **4.11E+00** |
| 16 | 9.99E-06 | **9.98E-06** | **9.98E-06** |
| 17 | 8.67E+00 | **4.34E+00** | 7.75E+00 |
| 18 | **9.97E-06** | 9.98E-06 | 9.98E-06 |
| 19 | 0.00E+00 | 0.00E+00 | 0.00E+00 |
| 20 | **1.22E+01** | 1.74E+01 | 1.43E+01 |
| 21 | 4.73E+00 | **2.97E+00** | 1.61E+01 |
| 22 | 1.45E+04 | 5.44E+05 | **9.89E-06** |
| 23 | **9.93E-06** | 1.00E-05 | 9.98E-06 |
| 24 | 9.99E-06 | **9.97E-06** | 9.99E-06 |
| 25 | 1.00E-05 | **9.96E-06** | 9.98E-06 |
| 26 | 9.99E-06 | **9.95E-06** | 9.99E-06 |
| 27 | 1.00E-05 | **9.95E-06** | 9.99E-06 |
| 28 | 2.41E+05 | 2.51E+05 | **2.40E+05** |
| 29 | **9.97E-06** | 9.99E-06 | 9.99E-06 |
| 30 | **9.99E-06** | **9.99E-06** | 1.00E-05 |

**Table 5.13**: Success Rate and Average number of function evaluations for successful runs of SMO, TS-SMO and QASMO (Non-Scalable problems)

| Problem No. | Success Rate | | | Average number of function evaluations for successful runs | | |
|---|---|---|---|---|---|---|
| | SMO | TS-SMO | QASMO | SMO | TS-SMO | QASMO |
| 31 | 100 | 100 | 100 | 3738 | 3636 | **3492** |
| 32 | **98** | 93 | 49 | **430014** | 465326 | 4374449 |
| 33 | 0 | 0 | 0 | N.A. | N.A. | N.A. |
| 34 | 100 | 100 | 100 | 3585 | 3415 | **3200** |
| 35 | 100 | 100 | 100 | 256584 | 211382 | **55756** |
| 36 | 100 | 100 | 100 | 3304 | 3199 | **3050** |
| 37 | **100** | **100** | 96 | 131394 | **131171** | 180988 |
| 38 | 100 | 100 | 100 | **26119** | 33359 | 542972 |
| 39 | 100 | 100 | 100 | 2009 | 2281 | **1936** |
| 40 | 100 | 100 | 100 | 3071 | **2793** | 2913 |
| 41 | 0 | 0 | 0 | N.A. | N.A. | N.A. |
| 42 | **100** | **100** | 96 | **24508** | 45553 | 317022 |
| 43 | 0 | 0 | 0 | N.A. | N.A. | N.A. |
| 44 | **100** | **100** | 99 | **13458** | 13519 | 29208 |
| 45 | 0 | 0 | 0 | N.A. | N.A. | N.A. |
| 46 | 100 | 100 | 100 | 10737 | 12525 | **8753** |

**Table 5.14**: Best of error values obtained in 100 independent runs of SMO, TS-SMO and QASMO (Non-Scalable problems)

| Problem No. | SMO | TSMO | QASMO |
|---|---|---|---|
| 31 | 8.53E-08 | **1.42E-08** | 1.86E-08 |
| 32 | **6.79E-08** | 8.20E-08 | 2.40E-07 |
| 33 | 1.21E-07 | 5.02E-08 | **1.04E-08** |
| 34 | 1.74E-07 | **2.22E-08** | 3.97E-08 |
| 35 | 1.22E-06 | 1.88E-06 | **8.66E-07** |
| 36 | 9.26E-08 | **3.21E-08** | 1.46E-07 |
| 37 | 2.41E-06 | 1.77E-06 | **1.21E-06** |
| 38 | 1.42E-06 | 6.44E-07 | **4.66E-07** |
| 39 | 3.75E-06 | 3.75E-06 | 3.75E-06 |
| 40 | 1.88E-08 | 1.53E-08 | **3.74E-09** |
| 41 | 3.75E-04 | 3.75E-04 | 3.75E-04 |
| 42 | **1.01E-06** | 1.44E-06 | 1.55E-06 |
| 43 | **5.96E-05** | 5.96E-05 | 5.96E-05 |
| 44 | 1.74E-08 | **3.21E-09** | 4.54E-08 |
| 45 | 4.82E-01 | 4.82E-01 | 4.82E-01 |
| 46 | 5.60E-08 | **2.58E-08** | 2.20E-07 |

**Table 5.15**: Average of error values obtained in 100 independent runs of SMO, TS-SMO and QASMO (Non-Scalable problems)

| Problem No. | SMO | TSMO | QASMO |
|---|---|---|---|
| 31 | 4.82E-06 | 4.61E-06 | **4.38E-06** |
| 32 | **1.11E-05** | 2.24E-05 | 1.39E-04 |
| 33 | 4.87E-06 | 4.69E-06 | **4.17E-06** |
| 34 | **4.35E-06** | 4.74E-06 | 4.61E-06 |
| 35 | 7.24E-06 | **6.92E-06** | **6.92E-06** |
| 36 | 4.37E-06 | 4.39E-06 | **3.97E-06** |
| 37 | 7.95E-06 | **7.78E-06** | 4.41E-05 |
| 38 | 6.25E-06 | **6.11E-06** | 6.24E-06 |
| 39 | 4.99E-06 | 5.31E-06 | **4.83E-06** |
| 40 | 4.26E-06 | 3.92E-06 | **3.32E-06** |
| 41 | **3.88E-04** | 2.41E-02 | 5.65E-02 |
| 42 | **5.77E-06** | 6.43E-06 | 2.02E-01 |
| 43 | **5.96E-05** | 6.12E-05 | 1.05E-01 |
| 44 | 4.48E-06 | **3.89E-06** | 5.36E-02 |
| 45 | 4.82E-01 | 4.82E-01 | 4.82E-01 |
| 46 | **3.65E-06** | 4.47E-06 | 4.75E-06 |

**Table 5.16**: Worst of error values obtained in 100 independent runs of SMO, TS-SMO and QASMO (Non-Scalable problems)

| Problem No. | SMO | TSMO | QASMO |
|---|---|---|---|
| 31 | 9.99E-06 | 9.96E-06 | **9.91E-06** |
| 32 | 2.87E-04 | 2.87E-04 | 2.87E-04 |
| 33 | **9.48E-06** | 9.98E-06 | 9.92E-06 |
| 34 | 9.98E-06 | 9.91E-06 | **9.86E-06** |
| 35 | 1.00E-05 | **9.96E-06** | **9.96E-06** |
| 36 | **9.75E-06** | 9.98E-06 | 9.91E-06 |
| 37 | **9.99E-06** | **9.99E-06** | 9.16E-04 |
| 38 | 9.91E-06 | **9.88E-06** | 9.95E-06 |
| 39 | 9.96E-06 | **9.91E-06** | 9.94E-06 |
| 40 | 9.94E-06 | 9.93E-06 | **9.89E-06** |
| 41 | **1.28E-03** | 1.19E-01 | 1.52E-01 |
| 42 | **9.88E-06** | 9.93E-06 | 5.05E+00 |
| 43 | **5.96E-05** | 2.20E-04 | 5.27E+00 |
| 44 | **9.92E-06** | 9.93E-06 | 5.36E+00 |
| 45 | 4.82E-01 | 4.82E-01 | 4.82E-01 |
| 46 | 9.70E-06 | 9.93E-06 | **9.57E-06** |

**Table 5.17**: Summary of results for Scalable and Non-Scalable problems of SMO, TS-SMO and QASMO

| **Scalable Problems** | | | |
|---|---|---|---|
| Performance metric | SMO | TS-SMO | QASMO |
| Success Rate | 22 | 22 | 24 |
| Highest Success Rate | 0 | 0 | 3 |
| Lowest number of function evaluations for successful runs | 3 | 4 | 18 |
| best error value | 7 | 7 | 15 |
| Average error value | 5 | 5 | 19 |
| Worst error value | 9 | 12 | 12 |
| Total | 46 | 50 | 91 |
| **Non-Scalable Problems** | | | |
| Performance metric | SMO | TS-SMO | QASMO |
| Success Rate | 11 | 11 | 8 |
| Highest Success Rate | 1 | 0 | 0 |
| Lowest number of function evaluations for successful runs | 4 | 2 | 6 |
| best error value | 3 | 5 | 5 |
| Average error value | 6 | 3 | 5 |
| Worst error value | 6 | 2 | 4 |
| Total | 31 | 23 | 28 |

# CHAPTER 6

# QUADRATIC APPROXIMATION BASED CONSTRAINED SPIDER MONKEY OPTIMIZATION ALGORITHM

The present chapter introduces a new version of Constrained Spider Monkey Optimization (CSMO) proposed in chapter 3. This new version is named as Quadratic Approximation Based Constrained Spider Monkey Optimization (QACSMO) algorithm. The motivation behind the proposed algorithm is the improvement in the performance of basic SMO for unconstrained continuous optimization problems after incorporating QA operator in it in chapter 5. Further, the comparison among basic SMO, TS-SMO and QASMO demonstrates the superiority of QASMO over the other two. So, in this chapter an attempt has been made to incorporate QA in constrained version of SMO to study its impact in solving constrained continuous optimization problems. Investigation has been made on the performance of the proposed algorithm by testing it over IEEE CEC2006 and CEC2010 benchmark problems and results are compared with the results of CSMO. The chapter is organised as follows: In section 6.1, the proposed algorithm is presented. In section 6.2, experimental setup has been provided. In section 6.3, experimental results have been discussed. Finally, the chapter has been concluded in section 6.4.

## 6.1 THE PROPOSED QUADRATIC APPROXIMATION BASED CONSTRAINED SPIDER MONKEY OPTIMIZATION

The implementation strategy of QA in CSMO is same as described for QA in SMO in subsection 5.2.2 of chapter 5. QA operator has been incorporated in Global Leader Learning Phase and Local Leader Learning Phase in CSMO with an objective to obtain better solutions in the neighbourhood of the global and local leaders. These two phases have been modified as follows:

**Modified Global Leader Learning Phase**: First, the position of the global leader and the worst member of the swarm are found using Deb’s three feasibility rules mentioned in subsection 3.1.1 in chapter 3. Best member of the swarm is updated as the global leader. Then QA operator mentioned in subsection 5.2.1 in chapter 5 is applied to generate a new solution using three solutions. Among these three solutions, one is the global leader and other two are randomly selected members of the swarm. New solutions are created using QA operator until we get a solution which is better than the worst member say $SM_{globalworst}$ of the swarm. Pseudocode of Modified Global Leader Learning Phase is given in Algorithm 6.1.

```
Begin
    Update the position of the global leader in the swarm
    find $SM_{worstglobal}$
        For $i$ = 1 to 1000 Do
            Select A = $GL$, B and C are positions of randomly chosen members of the
            swarm such that A, B, C all are distinct
            //generate $P$ using equation (5.1)
            Compare the fitness of $P$ and $SM_{globalworst}$ using Deb's three feasibility rules
            If (fitness($P$) > fitness($SM_{globalworst}$)) Then
                $SM_{globalworst} = P$
                Terminate the loop
            End if
        End For
        If (fitness($P$) >fitness($GL$))
            $GL = P$
        End If
        If (position of global leader is updated from previous position) Then
            $GLC = 0$
        Else
            $GLC = GLC + 1$
        End If
End
```

**Algorithm 6.1**: Modified Global Leader Learning Phase

M**odified Local Leader Learning Phase**: In every group, select the local leader and the worst member of the group using Deb's three feasibility rules. Then QA operator mentioned in subsection 5.2.1 in chapter 5 is applied using three solutions. Among these three solutions, one is the local leader and other two are randomly selected members of that group. The process is repeated till we get a solution which is better than worst member say $SM_{localworst}$ of that group. Pseudocode for Modified Local Leader Learning Phase is given in Algorithm 6.2.

```
Begin
    For $k$ = 1 to $NG$
        Update the position of the local leader in the group
        find $SM_{localworst}$
        For $i$ = 1 to 1000 Do
        Select A= $LL_k$ , B and C are positions of randomly chosen members of the
        swarm such that A, B, C all are distinct
        //generate $P$ using equation (5.1)
        Compare the fitness of $P$ and $SM_{localworst}$ using Deb's three feasibility rules
            If (fitness($P$) > fitness($SM_{localworst}$)) Then
                $SM_{localworst}$ = $P$
                Terminate the loop
            End if
        End For
        If (fitness($P$)>fitness($LL_k$)) Then
            $LL_k$= $P$
        End If
        If (position of local leader is updated from previous position) Then
            $LLC_k$ = 0
        Else
            $LLC_k$ = $LLC_k$ + 1
        End If
    End For
End
```

**Algorithm 6.2**: Modified Local Leader Learning Phase

Pseudocode for QACSMO is provided in Algorithm 6.3.

```
Begin
    Initialize the swarm using eq. (2.1)
    Initialize LLlt, GLlt, Pr, MG
    Iteration=0
    Calculate fitness value of the position of each spider monkey in the swarm
    using Algorithm 3.1
    Select Global Leader and Local Leaders by applying Deb's Three feasibility rules
    While (termination criterion is not satisfied) Do
        //Local Leader Phase (Algorithm 3.2)
        //Calculate Probability of spider monkeys (using eq. (2.3))
        //Global Leader Phase (Algorithm 3.3)
        //Modified Global Leader Learning Phase (Algorithm 6.1)
        //Modified Local Leader Learning Phase (Algorithm 6.2)
        //Local Leader Decision Phase (Algorithm 2.5)
        //Global Leader Decision Phase (Algorithm 2.6)
        Iteration = Iteration +1
    End While
End
```

**Algorithm 6.3:** Pseudocode for QACSMO

## 6.2 EXPERIMENTAL SETUP

The performance of the proposed QACSMO has been evaluated on CEC2006 [107] and CEC2010 [112] benchmark problems given in Appendix I and Appendix II respectively and the results have been compared with CSMO proposed in chapter 3. In order to have fair comparison, same benchmark problems and evaluation criteria has been adopted for QACSMO as mentioned for CSMO in subsections 3.3.1 and 3.3.3 of section 3.3 in chapter 3.

## 6.3 DISCUSSION OF EXPERIMENTAL RESULTS

Tables 6.1-6.8 present the results for CEC2006 benchmark problems and Tables 6.9-6.14 and Tables 6.15-6.20 present the results for CEC2010 benchmark problems for 10 dimensions and 30 dimensions respectively. In order to observe whether the results are significantly different or not, Wilcoxon rank sum test at 5% ($\alpha = 0.05$) significance level is performed between QACSMO-CSMO for both CEC2006 and CEC2010 benchmark sets. The null hypothesis assumed for this statistical test is “if there is no difference in the performance of the algorithms” means both the algorithms are equivalent and alternative hypothesis being “there is a difference in the performance of the algorithms”. The test has been applied to the sample containing results of 25 independent runs performed by each algorithm for each benchmark problem. The output of the applied test has been presented in tabular form. If there is no significant difference between the results, ‘=’ sign appears and when there is significant difference between the results, ‘+’ or ‘-’ sign appears based on QACSMO is performing better or worse than CSMO.

### 6.3.1 DISCUSSION OF RESULTS FOR CEC2006 BENCHMARK PROBLEMS

The feasibility rate and success rate of both the algorithms have been provided in Table 6.1. From this table, it can be seen that QACSMO has better performance than CSMO on three problems (g05, g13, g17) in terms of feasibility rate and both the algorithms have same feasibility rate on rest of the problems. In terms of success rate, CSMO has better performance than QACSMO on five problems (g02, g04, g06, g16 and g18) and QACSMO has better performance only on g15. Table 6.2 present the best, median and worst of function error values obtained in 25 runs. Table 6.3 present mean and standard deviation of function error values of feasible runs only. Table 6.4 and Table 6.5 present the best, median, worst, mean and standard deviation of the number of function evaluations for successful runs obtained in 25 runs. The summary of comparison of QACSMO against CSMO has been provided in Table 6.6. From this table, it can be seen that QACSMO is better than CSMO only in terms of feasibility rate. On rest of the performance metrics, CSMO is performing better than QACSMO.

Results of Wilcoxon rank sum test based on function error value is presented in Table 6.7 which shows QACSMO is either equivalent or significantly worse than CSMO on all the problems except problem g08. Table 6.8 presents the average execution time taken by both the algorithms per run. This table shows there is not much difference in the execution time of both the algorithms. So, this table should be considered for information purpose only.

Convergence graphs for problems g01-g24 has been plotted in Figures 6.1-6.3. In the convergence graphs, value on the horizontal axis represents the number of iterations and the vertical axis shows the function error value. The logarithmic graphs have been plotted as the range of the problems was large. Sudden rise in some graphs show that the solution has entered the feasible region.

### 6.3.2 DISCUSSION OF RESULTS FOR CEC2010 BENCHMARK PROBLEMS FOR 10 DIMENSIONS AND 30 DIMENSIONS

Table 6.9 presents the feasibility rate of CEC2010 benchmarks for 10 dimensions only. It shows that CSMO performs better than QACSMO on eight (C02, C03, C04, C09, C12, C16, C17, C18) problems, while QACSMO performs better on two (C06 and C10) problems. Also, from Tables 6.10 and 6.11, it can be seen that CSMO performs better than QACSMO on most of the problems. The summary of comparison of QACSMO against CSMO has been provided in Table 6.12. This table shows that QACSMO performs worse than CSMO on all the performance metrics. Results of  Wilcoxon rank sum test based on objective function value for 10 dimensions problems is presented in Table 6.13. This table shows that QACSMO performs equivalent or significantly worse than CSMO on all the problems. Average execution time of both the algorithms is presented in Table 6.14. From this table, it can be seen that there is not much difference in the execution time of both the algorithms.

Table 6.15 presents feasibility rate of CSMO and QACSMO for CEC2010 benchmark problems for 30 dimensions. CSMO outperforms QACSMO five problems (C11, C12, C16, C17, C18) in terms of feasibility rate. Tables 6.16 and 6.17 show CSMO outperforms QACSMO on most of the problems. The summary of results for 30 dimensions has been provided in Table 6.18. Results of Wilcoxon rank sum test based on objective function value for 30 dimensions problems is presented in Table 6.19 which shows QACSMO is either equivalent or significantly worse than CSMO on all the problems except problem C18.. The average execution time taken by both the algorithms is presented in Table 6.20.

## 6.4 CONCLUSIONS

This chapter proposes a new version of CSMO named as QACSMO after incorporating quadratic approximation operator in it. The performance of QACSMO is evaluated on IEEE CEC2006 and CEC2010 benchmark problems and results are compared with CSMO. Discussion of results demonstrates that incorporation of quadratic approximation operator has negative impact on the performance of CSMO. Since it is an experimental study, the actual reason of the performance deterioration cannot be figured out. But the possible reason can be the decrease in the number of iterations because of the consumption of more number of function evaluations in each iteration in Modified Global leader learning phase and Modified Local Leader Learning Phase of QACSMO. Termination criteria adopted for both the algorithms is the fixed number of function evaluations. Consumption of high number of function evaluations in Modified Global Leader Learning Phase and Modified Local Leader Learning Phase result in less number of iterations for QACSMO than CSMO. Less number of iterations may interrupt the algorithm in the exploration of the search space. So, the conclusion is quadratic approximation operator which improves the performance of SMO for solving unconstrained optimization problems is not able to perform well for solving constrained optimization problems. Though, algorithms are modified with an objective to improve the performance of the algorithm and better results are expected with these modifications. But the presentation of bad results is also necessary to save the user who wants to solve constrained problems using SMO from wasting time. User can think that if quadratic approximation operator is going well with SMO for unconstrained optimization problems, then it will also go well with constrained optimization problems without testing it, but this is not the case. Also, it may be noted that whatever the conclusion has been made, it is made for CEC2006 and CEC2010 benchmark problems. Also, it shows that we cannot conclude if a modification is working well for unconstrained optimization problems, it does not guarantee that it will work for constrained optimization problems also. In future, other modifications will be tried to improve the performance of CSMO.

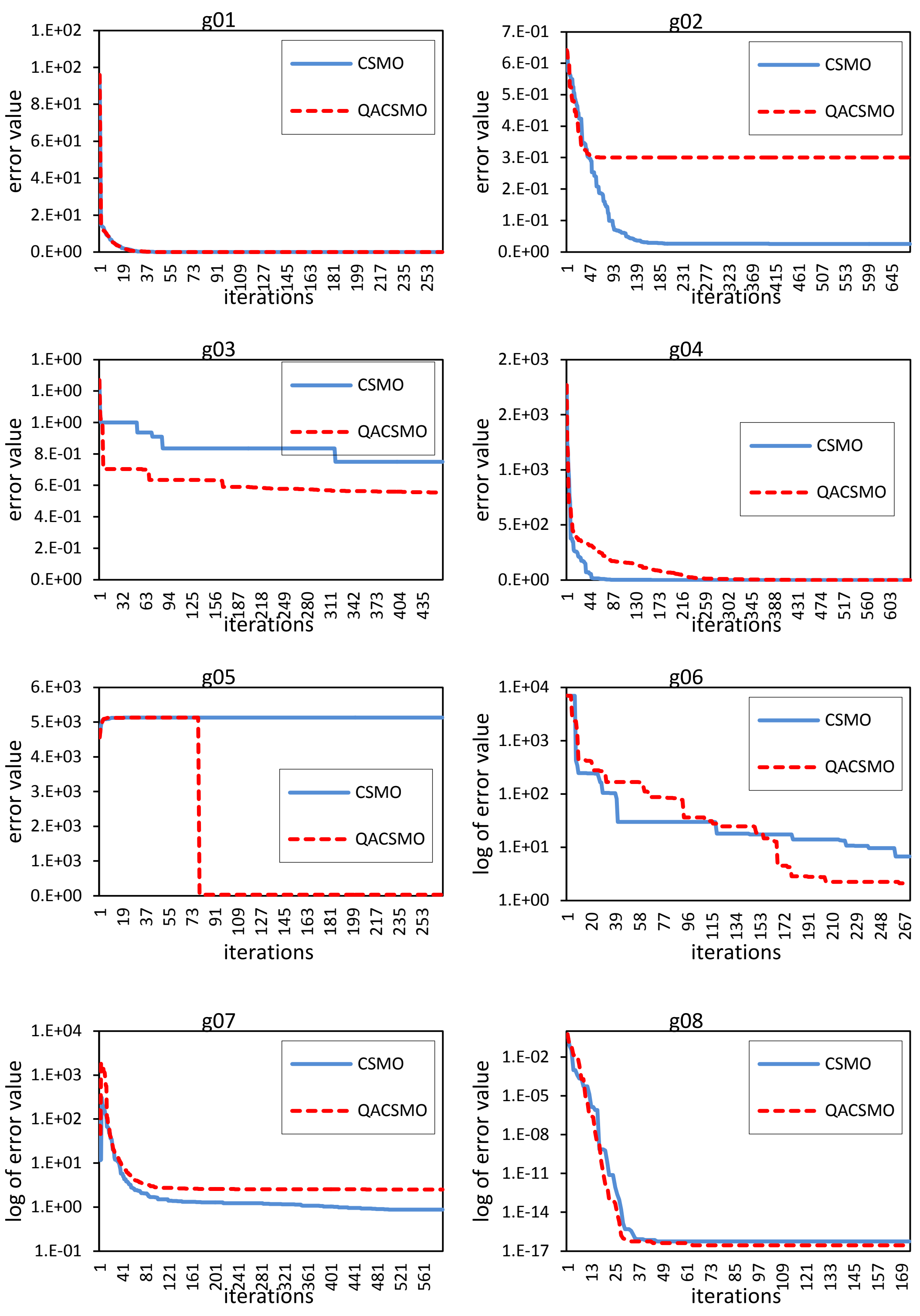

**Figure 6.1:** Convergence graphs of problems g01-g08

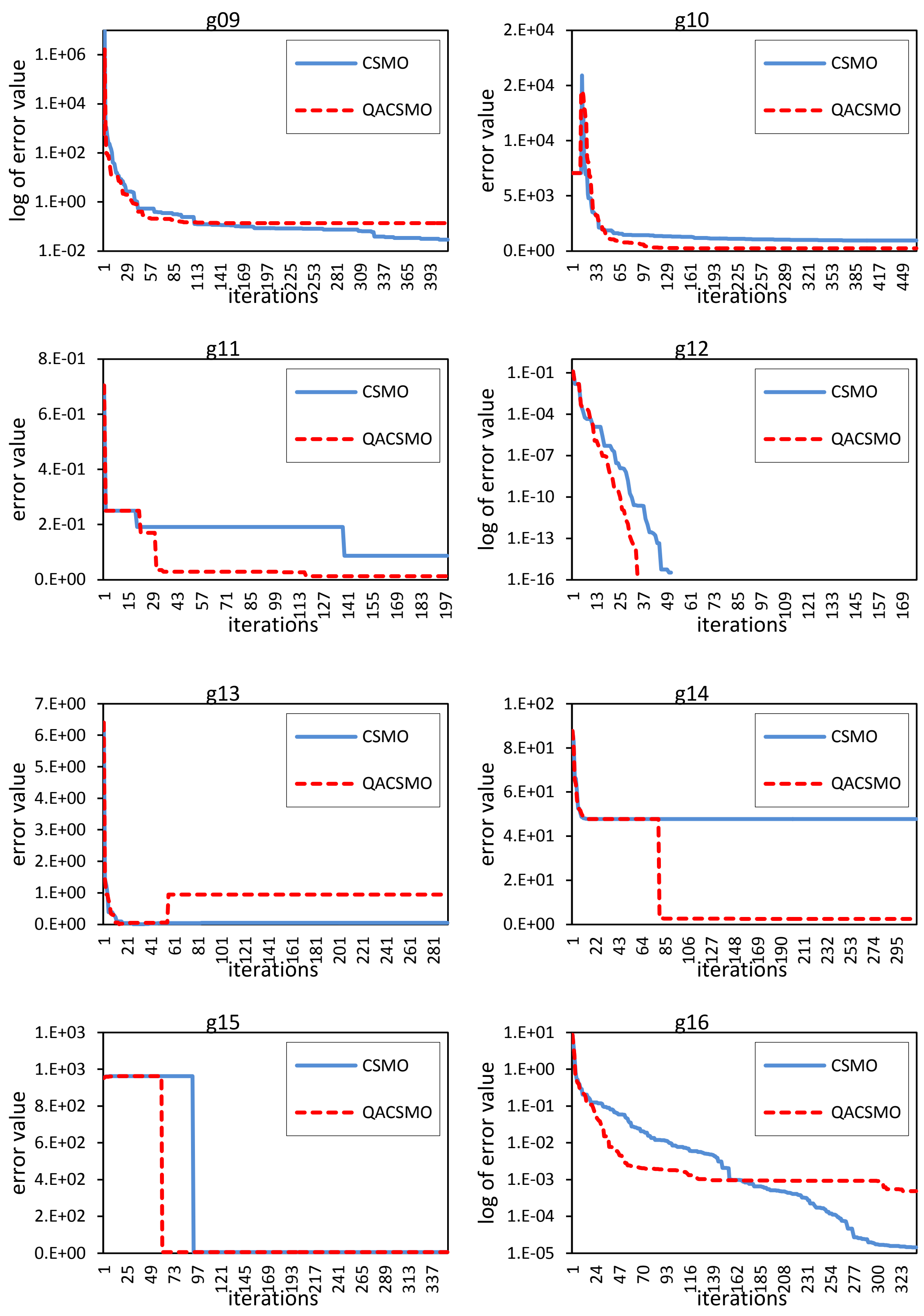

**Figure 6.2:** Convergence graphs of problems g09-g16

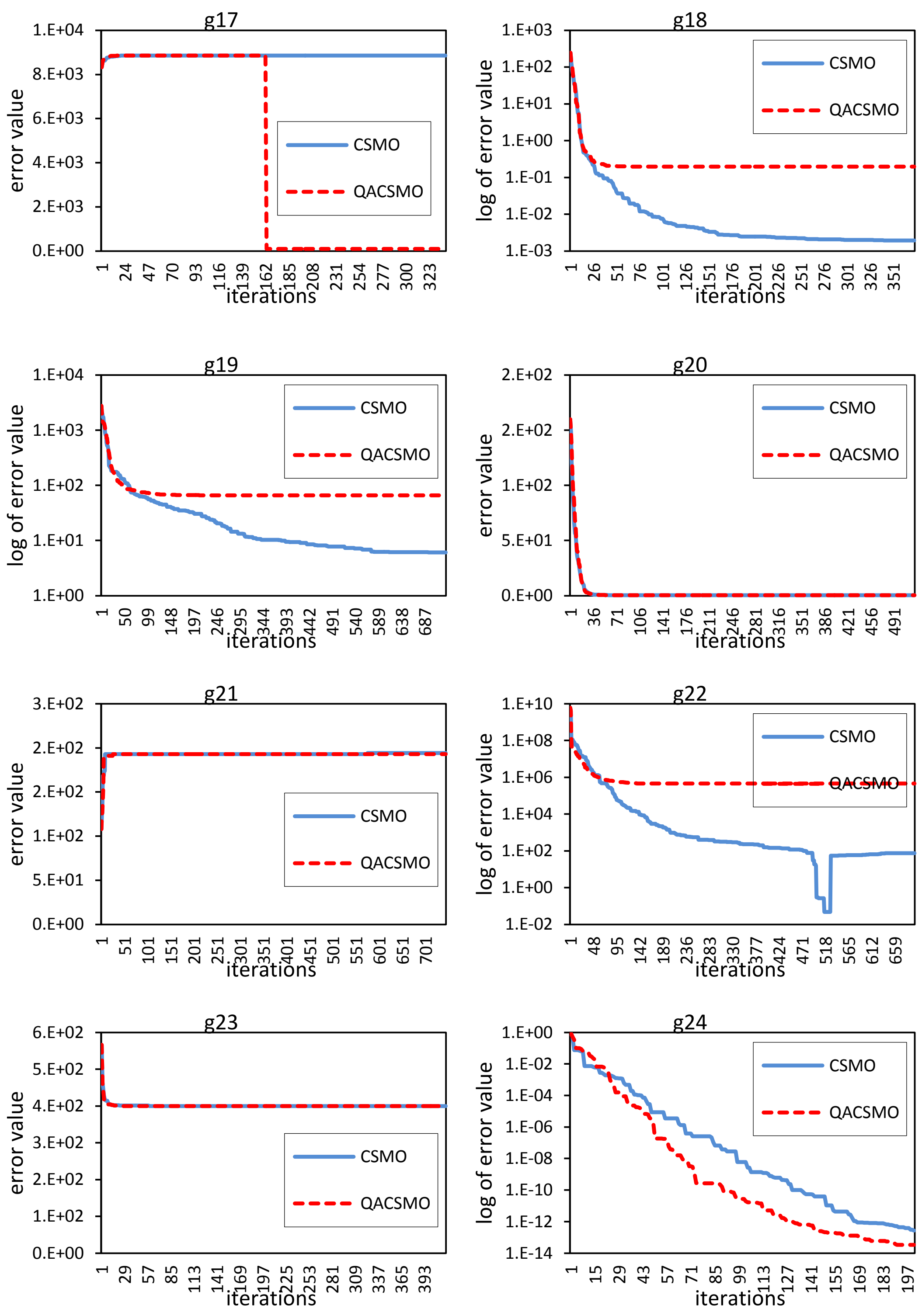

**Figure 6.3:** Convergence graphs of problems g17-g24

**Table 6.1**: Comparison of Feasibility Rate and Success Rate for CEC2006 Benchmark Problems

| Problems | Feasibility Rate | | Success Rate | |
|---|---|---|---|---|
| | CSMO | QACSMO | CSMO | QACSMO |
| g01 | 100 | 100 | 100 | 100 |
| g02 | 100 | 100 | **8** | 0 |
| g03 | 100 | 100 | 0 | 0 |
| g04 | 100 | 100 | **100** | 92 |
| g05 | 80 | **88** | 0 | 0 |
| g06 | 100 | 100 | **96** | 0 |
| g07 | 100 | 100 | 0 | 0 |
| g08 | 100 | 100 | 100 | 100 |
| g09 | 100 | 100 | 0 | 0 |
| g10 | 100 | 100 | 0 | 0 |
| g11 | 100 | 100 | 24 | 24 |
| g12 | 100 | 100 | 100 | 100 |
| g13 | 88 | **100** | 0 | 0 |
| g14 | 100 | 100 | 0 | 0 |
| g15 | 100 | 100 | 0 | 8 |
| g16 | 100 | 100 | **100** | 16 |
| g17 | 96 | **100** | 0 | 0 |
| g18 | 100 | 100 | **20** | 4 |
| g19 | 100 | 100 | 0 | 0 |
| g20 | 0 | 0 | 0 | 0 |
| g21 | 0 | 0 | 0 | 0 |
| g22 | 0 | 0 | 0 | 0 |
| g23 | 16 | 16 | 0 | 0 |
| g24 | 100 | 100 | 100 | 100 |

**Table 6.2**: Comparison of Best, Median and Worst of function error values obtained with 25 independent runs for CEC2006 Benchmark Problems

| Problems | **Best** | | **Median** | | **Worst** | |
|---|---|---|---|---|---|---|
| | CSMO | QACSMO | CSMO | QACSMO | CSMO | QACSMO |
| g01 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 |
| g02 | **2.55E-05** | 3.88E-02 | **1.12E-02** | 1.77E-01 | **4.96E-02** | 3.00E-01 |
| g03 | 2.39E-01 | **2.16E-02** | **4.88E-01** | 8.68E-01 | **8.86E-01** | 9.74E-01 |
| g04 | **0.00E+00** | 3.64E-12 | **0.00E+00** | 6.18E-11 | **0.00E+00** | 2.62E+00 |
| g05 | **4.79E-02** | 8.72E-02 | 5.49E+02 | **2.43E+02** | 5.30E-03(1) | **2.28E-03(1)** |
| g06 | **2.79E-08** | 1.98E-03 | **6.93E-07** | 5.88E-01 | **2.40E-04** | 2.02E+01 |
| g07 | **3.05E-02** | 4.58E-02 | **2.81E-01** | 1.15E+00 | **7.18E-01** | 9.41E+00 |
| g08 | 4.16E-17 | **1.39E-17** | 4.16E-17 | **2.78E-17** | 4.16E-17 | 4.16E-17 |
| g09 | **6.75E-04** | 4.43E-03 | **5.10E-03** | 1.18E-01 | **1.07E-02** | 4.25E-01 |
| g10 | 1.70E+01 | **1.65E+01** | **2.06E+02** | 3.85E+02 | **9.91E+02** | 1.51E+03 |
| g11 | 5.99E-06 | **5.14E-08** | 1.55E-02 | **8.28E-04** | 2.50E-01 | **1.55E-01** |
| g12 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 |
| g13 | 4.91E-01 | **1.54E-01** | **9.09E-01** | 9.32E-01 | 2.14E-04(1) | **1.33E+00** |
| g14 | **5.10E-01** | 2.50E+00 | **4.09E+00** | 4.91E+00 | **6.09E+00** | 9.64E+00 |
| g15 | 1.66E-03 | **1.00E-05** | **6.46E-01** | 1.02E+00 | 1.06E+01 | **1.05E+01** |
| g16 | **6.49E-12** | 1.41E-06 | **6.68E-10** | 7.86E-03 | **1.49E-07** | 2.43E-01 |
| g17 | 5.49E+01 | **2.30E+01** | 1.59E+02 | **1.06E+02** | 8.49E-05(1) | **3.85E+02** |
| g18 | **4.34E-05** | 9.41E-05 | **1.94E-04** | 6.92E-03 | **2.42E-03** | 2.12E-01 |
| g19 | **1.30E+00** | 1.49E+01 | **5.95E+00** | 4.34E+01 | **1.49E+01** | 7.21E+01 |
| g20 | **3.72E-03(6)** | 1.90E-02(9) | **1.84E-02(11)** | 2.70E-02(9) | **3.04E-02(15)** | 3.13E-02(8) |
| g21 | 1.51E-03(2) | **1.38E-03(1)** | **5.44E-03(3)** | 1.23E-02(2) | **1.71E-02(3)** | 8.40E-02(2) |
| g22 | **2.31E01(16)** | 1.61E+00(19) | **5.06E+00(10)** | 1.33E+03(19) | **4.26E+04(19)** | 2.96E+05(18) |
| g23 | 4.00E+02 | **3.33E+02** | **7.36E-04** | 7.52E-02(1) | **8.32E-03(4)** | 4.00E-01(3) |
| g24 | **1.24E-14** | 3.29E-14 | **1.24E-14** | 3.29E-14 | **1.24E-14** | 5.68E-14 |

**Table 6.3**: Comparison of Mean and Standard Deviation of function error values obtained with feasible runs out of 25 independent runs for CEC2006 Benchmark Problems

| Problems | Mean | | Stdev | |
|---|---|---|---|---|
| | CSMO | QACSMO | CSMO | QACSMO |
| g01 | 0.00E+00 | 0.00E+00 | 0 | 0 |
| g02 | **1.54E-02** | 1.63E-01 | **0.015199** | 0.070354 |
| g03 | **4.91E-01** | 7.85E-01 | **0.144542** | 0.218356 |
| g04 | **0.00E+00** | 1.16E-01 | **0** | 0.524738 |
| g05 | 3.68E+02 | **3.24E+02** | 392.0755 | **328.2717** |
| g06 | **1.43E-05** | 2.05E+00 | **0.000048** | 4.16957 |
| g07 | **3.01E-01** | 1.85E+00 | **0.208882** | 2.201276 |
| g08 | 4.16E-17 | **2.83E-17** | **6.29E-33** | 4.86E-18 |
| g09 | 5.27E-03 | **1.35E-01** | **0.002991** | 0.108808 |
| g10 | **2.69E+02** | 5.04E+02 | **219.6704** | 438.4481 |
| g11 | 8.49E-02 | **2.68E-02** | 0.103121 | **0.050794** |
| g12 | 0.00E+00 | 0.00E+00 | 0 | 0 |
| g13 | 8.58E-01 | **8.35E-01** | **0.146289** | 0.277501 |
| g14 | **3.90E+00** | 4.90E+00 | **1.431023** | 1.649259 |
| g15 | **2.98E+00** | 3.16E+00 | 3.6463 | **3.433578** |
| g16 | **9.79E-09** | 3.54E-02 | **3.13E-08** | 0.06049 |
| g17 | 2.21E+02 | **1.94E+02** | 139.4337 | **126.7075** |
| g18 | **3.47E-04** | 5.10E-02 | **0.000474** | 0.084862 |
| g19 | **6.33E+00** | 4.24E+01 | **3.771037** | 16.4541 |
| g20 | N.A. | N.A. | N.A. | N.A. |
| g21 | N.A. | N.A. | N.A. | N.A. |
| g22 | N.A. | N.A. | N.A. | N.A. |
| g23 | **4.00E+02** | 4.35E+02 | **0** | 147.5678 |
| g24 | **1.24E-14** | 3.50E-14 | **3.22E-30** | 6.59E-15 |

**Table 6.4**: Comparison of Best, Median and Worst of the number of function evaluations for successful runs obtained with 25 independent runs for CEC2006 Benchmark Problems

| Problems | Best | | Median | | Worst | |
|---|---|---|---|---|---|---|
| | CSMO | QACSMO | CSMO | QACSMO | CSMO | QACSMO |
| g01 | 8950 | **8610** | **10750** | 12307 | **28150** | 40675 |
| g02 | 217736 | N.A. | 255548.5 | N.A. | 293361 | N.A. |
| g03 | N.A. | N.A. | N.A. | N.A. | N.A. | N.A. |
| g04 | 17550 | **13474** | **21850** | 26405 | **30150** | 141786 |
| g05 | N.A. | N.A. | N.A. | N.A. | N.A. | N.A. |
| g06 | 165267 | N.A. | 201324 | N.A. | 326597 | N.A. |
| g07 | N.A. | N.A. | N.A. | N.A. | N.A. | N.A. |
| g08 | 650 | **459** | 950 | **780** | 1250 | **970** |
| g09 | N.A. | N.A. | N.A. | N.A. | N.A. | N.A. |
| g10 | N.A. | N.A. | N.A. | N.A. | N.A. | N.A. |
| g11 | 274204 | **9271** | 359310 | **53018** | 476918 | **229429** |
| g12 | 350 | **255** | 1050 | **907** | 1650 | **1454** |
| g13 | N.A. | N.A. | N.A. | N.A. | N.A. | N.A. |
| g14 | N.A. | N.A. | N.A. | N.A. | N.A. | N.A. |
| g15 | N.A. | 7607 | N.A. | 8997 | N.A. | 10388 |
| g16 | 12850 | **8098** | 26050 | **12448** | 80724 | **32107** |
| g17 | N.A. | N.A. | N.A. | N.A. | N.A. | N.A. |
| g18 | **219036** | 364579 | 372280 | **364579** | 476428 | **364579** |
| g19 | N.A. | N.A. | N.A. | N.A. | N.A. | N.A. |
| g20 | N.A. | N.A. | N.A. | N.A. | N.A. | N.A. |
| g21 | N.A. | N.A. | N.A. | N.A. | N.A. | N.A. |
| g22 | N.A. | N.A. | N.A. | N.A. | N.A. | N.A. |
| g23 | N.A. | N.A. | N.A. | N.A. | N.A. | N.A. |
| g24 | 2950 | **2711** | 4250 | **3467** | 5850 | **4049** |

**Table 6.5**: Comparison of Mean and Standard Deviation of the number of function evaluations for successful runs obtained with 25 independent runs for CEC2006 Benchmark Problems

| Problems | Mean | | Stdev | |
|---|---|---|---|---|
| | CSMO | QACSMO | CSMO | QACSMO |
| g01 | **13714** | 17251 | **6026** | 9680 |
| g02 | 255548 | N.A. | 53474 | N.A. |
| g03 | N.A. | N.A. | N.A. | N.A. |
| g04 | **22690** | 44916 | **2973** | 39397 |
| g05 | N.A. | N.A. | N.A. | N.A. |
| g06 | 216177 | N.A. | 42756 | N.A. |
| g07 | N.A. | N.A. | N.A. | N.A. |
| g08 | 934 | **786** | 146 | **121** |
| g09 | N.A. | N.A. | N.A. | N.A. |
| g10 | N.A. | N.A. | N.A. | N.A. |
| g11 | 374996 | **82512** | **80233** | 85497 |
| g12 | 1026 | **893** | 274 | **266** |
| g13 | N.A. | N.A. | N.A. | N.A. |
| g14 | N.A. | N.A. | N.A. | N.A. |
| g15 | N.A. | 8997 | N.A. | 1966 |
| g16 | 34930 | **16275** | 20422 | **10845** |
| g17 | N.A. | N.A. | N.A. | N.A. |
| g18 | **348277** | 364579 | 109348 | N.A. |
| g19 | N.A. | N.A. | N.A. | N.A. |
| g20 | N.A. | N.A. | N.A. | N.A. |
| g21 | N.A. | N.A. | N.A. | N.A. |
| g22 | N.A. | N.A. | N.A. | N.A. |
| g23 | N.A. | N.A. | N.A. | N.A. |
| g24 | 4258 | **3421** | 796 | **372** |

**Table 6.6**: Summary of comparison of QACSMO against CSMO for CEC2006 Benchmark Problems

| | Criteria | Better | Equal | Worse |
|---|---|---|---|---|
| QACSMO vs. CSMO | Feasibility rate | 3 | 21 | 0 |
| | Success rate | 1 | 19 | 4 |
| | Function error values | | | |
| | Criteria | Better | Equal | Worse |
| | Best | 9 | 2 | 13 |
| | Median | 4 | 2 | 18 |
| | Worst | 5 | 3 | 16 |
| | Mean | 6 | 2 | 13 |
| | Stdev | 4 | 2 | 15 |
| | Number of function evaluations for successful runs | | | |
| | Criteria | Better | Equal | Worse |
| | Best | 7 | 0 | 1 |
| | Median | 6 | 0 | 2 |
| | Worst | 5 | 0 | 2 |
| | Mean | 5 | 0 | 3 |
| | Stdev | 3 | 0 | 3 |

**Table 6.7:** Wilcoxon Rank sum test based on function error values with a significance level of $\alpha = 0.05$ for CEC2006 Benchmark Problems ('+' indicates QACSMO is significantly better, '-' indicates QACSMO is significantly worse and '=' indicates there is no significant difference)

| Problems | Sign | Problems | Sign |
|---|---|---|---|
| g01 | = | g13 | = |
| g02 | - | g14 | = |
| g03 | - | g15 | = |
| g04 | - | g16 | - |
| g05 | = | g17 | = |
| g06 | - | g18 | - |
| g07 | - | g19 | - |
| g08 | + | g20 | - |
| g09 | - | g21 | = |
| g10 | = | g22 | - |
| g11 | = | g23 | - |
| g12 | = | g24 | - |

**Table 6.8**: Average time per run (in seconds) for CEC2006 Benchmark Problems

| Problems | CSMO | QACSMO | Problems | CSMO | QACSMO |
|---|---|---|---|---|---|
| g01 | 0.41768 | 0.97928 | g13 | 1.35976 | 0.42004 |
| g02 | 6.40724 | 2.74136 | g14 | 0.34888 | 0.65752 |
| g03 | 0.35684 | 0.655 | g15 | 1.65564 | 0.3144 |
| g04 | 0.21896 | 0.37756 | g16 | 1.88088 | 0.61992 |
| g05 | 0.4378 | 0.54464 | g17 | 2.35924 | 0.68696 |
| g06 | 0.2148 | 0.17312 | g18 | 9.66048 | 1.0168 |
| g07 | 0.36836 | 0.69132 | g19 | 4.32468 | 1.37668 |
| g08 | 1.03548 | 0.24876 | g20 | 1.12376 | 1.97204 |
| g09 | 1.30812 | 0.54604 | g21 | 1.50956 | 0.75236 |
| g10 | 0.31108 | 0.53376 | g22 | 2.59196 | 1.52636 |
| g11 | 0.13656 | 0.17436 | g23 | 0.34364 | 0.62092 |
| g12 | 3.90484 | 4.55596 | g24 | 3.0542 | 0.32632 |

**Table 6.9**: Feasibility Rate for CEC2010 Benchmark Problems (10 dimensions)

| Problems | CSMO | QACSMO | Problems | CSMO | QACSMO |
|---|---|---|---|---|---|
| C01 | 100 | 100 | C10 | 8 | 20 |
| C02 | **100** | 44 | C11 | 0 | 0 |
| C03 | **100** | 0 | C12 | **100** | 72 |
| C04 | 16 | 0 | C13 | 100 | 100 |
| C05 | 0 | 0 | C14 | 100 | 100 |
| C06 | 0 | 16 | C15 | 100 | 100 |
| C07 | 100 | 100 | C16 | **100** | 80 |
| C08 | 100 | 100 | C17 | **100** | 56 |
| C09 | 8 | 4 | C18 | **100** | 84 |

**Table 6.10**: Best, Median and Worst of objective function values obtained with 25 independent runs for CEC2010 Benchmark Problems (10 dimensions)

| Problems | Best | | Median | | Worst | |
|---|---|---|---|---|---|---|
| | CSMO | QACSMO | CSMO | QACSMO | CSMO | QACSMO |
| C01 | -7.47E-01 | -7.47E-01 | -7.47E-01 | -7.47E-01 | **-7.47E-01** | -6.18E-01 |
| C02 | **4.96E-01** | 8.31E-01 | **2.10E+00** | 4.17E-05(1) | **3.55E+00** | 1.20E-04(1) |
| C03 | **1.29E+08** | 2.16E-04(1) | **2.73E+12** | 4.39E-04(1) | **5.22E+14** | 1.04E-03(1) |
| C04 | **1.55E-03** | 1.53E-03(2) | **9.02E-04(2)** | 7.23E-03(3) | **2.28E+00(2)** | 4.10E+00(3) |
| C05 | **3.08E-04(2)** | 6.00E-03(2) | **5.92E-03(2)** | 7.33E-02(2) | **1.73E-02(2)** | 1.67E-01(2) |
| C06 | 6.02E-04(2) | **3.05E-01** | **9.14E-03(2)** | 6.35E-02(2) | **2.31E-02(2)** | 2.18E-01(2) |
| C07 | **6.37E-02** | 1.86E-01 | **1.57E+00** | 2.12E+00 | **7.20E+01** | 8.16E+01 |
| C08 | **4.12E-05** | 4.90E-02 | **1.06E+01** | 1.10E+01 | 1.02E+03 | **7.71E+02** |
| C09 | **1.35E+12** | 1.43E+13 | **3.90E-04(1)** | 6.58E-03(1) | **5.79E-03(1)** | 2.16E-02(1) |
| C10 | **3.15E+11** | 4.00E+11 | **5.29E-04(1)** | 4.42E-03(1) | **2.88E-03(1)** | 3.46E-02(1) |
| C11 | **7.24E-04(1)** | 8.27E-02(1) | **8.53E-01(1)** | 1.35E+00(1) | 1.89E+01(1) | **4.74E+00(1)** |
| C12 | **-5.70E+02** | -3.31E+01 | **-2.62E+02** | 1.56E-01 | **2.14E+01** | 3.04E+02(1) |
| C13 | -6.84E+01 | -6.84E+01 | **-6.74E+01** | -6.35E+01 | **-6.23E+01** | -6.12E+01 |
| C14 | **4.95E-03** | 4.81E-01 | **5.43E-01** | 1.11E+02 | **7.42E+00** | 3.88E+06 |
| C15 | **8.23E+10** | 7.69E+11 | **1.15E+12** | 1.22E+13 | **8.53E+12** | 5.07E+14 |
| C16 | **8.33E-01** | 9.92E-01 | **1.02E+00** | 1.03E+00 | **1.05E+00** | 1.96E-04(2) |
| C17 | 8.19E+01 | **6.18E+01** | **3.04E+02** | 6.76E+02 | **1.01E+03** | 2.58E-04(1) |
| C18 | **2.32E+03** | 2.41E+03 | **6.78E+03** | 7.92E+03 | **1.70E+04** | 2.83E-04(2) |

**Table 6.11**: Mean and Standard Deviation of objective function values obtained with feasible runs out of 25 independent runs for CEC2010 Benchmark Problems (10 dimensions)

| Problems | Mean | | Stdev | |
|---|---|---|---|---|
| | CSMO | QACSMO | CSMO | QACSMO |
| C01 | **-7.47E-01** | -7.35E-01 | **0.00E+00** | 3.08E-02 |
| C02 | **2.22E+00** | 3.48E+00 | **9.03E-01** | 1.45E+00 |
| C03 | **4.82E+13** | N.A. | **1.16E+14** | N.A. |
| C04 | **7.00E+00** | N.A. | **8.08E+00** | N.A. |
| C05 | N.A. | N.A. | N.A. | N.A. |
| C06 | N.A. | **3.05E-01** | N.A. | 0.00E+00 |
| C07 | **9.63E+00** | 1.22E+01 | **1.88E+01** | 2.42E+01 |
| C08 | **6.27E+01** | 1.05E+02 | 2.03E+02 | **1.94E+02** |
| C09 | 1.50E+13 | **1.43E+13** | **1.93E+13** | N.A. |
| C10 | **4.68E+11** | 2.77E+12 | **2.16E+11** | 3.40E+12 |
| C11 | N.A. | N.A. | N.A. | N.A. |
| C12 | **-2.94E+02** | -1.38E+00 | 2.75E+02 | **8.06E+00** |
| C13 | **-6.66E+01** | -6.46E+01 | **2.21E+00** | 2.59E+00 |
| C14 | **1.34E+00** | 6.37E+05 | **1.95E+00** | 1.45E+06 |
| C15 | **1.90E+12** | 4.54E+13 | **2.27E+12** | 1.03E+14 |
| C16 | **9.96E-01** | 1.03E+00 | **5.27E-02** | 1.93E-02 |
| C17 | 3.83E+02 | **3.44E+02** | 2.33E+02 | **2.06E+02** |
| C18 | **7.48E+03** | 8.49E+03 | **3.54E+03** | 5.93E+03 |

**Table 6.12**: Summary of comparison of QACSMO against CSMO for CEC2010 Benchmark Problems (10 dimensions)

| | Criteria | Better | Equal | Worse |
|---|---|---|---|---|
| QACSMO vs. CSMO | Feasibility rate | 2 | 8 | 8 |
| | Function error values | | | |
| | Criteria | Better | Equal | Worse |
| | Best | 2 | 14 | 2 |
| | Median | 0 | 1 | 17 |
| | Worst | 2 | 0 | 16 |
| | Mean | 2 | 0 | 11 |
| | Stdev | 3 | 0 | 9 |

**Table 6.13:** Wilcoxon Rank sum test based on objective function value with a significance level of $\alpha = 0.05$ for CEC2010 Benchmark Problems (10 dimensions) ('+' indicates CSMO is significantly better, '-' indicates QACSMO is significantly worse and '=' indicates there is no significant difference)

| Problems | Sign | Problems | Sign |
|---|---|---|---|
| C01 | = | C10 | - |
| C02 | - | C11 | = |
| C03 | - | C12 | - |
| C04 | = | C13 | - |
| C05 | - | C14 | - |
| C06 | - | C15 | - |
| C07 | = | C16 | = |
| C08 | - | C17 | - |
| C09 | - | C18 | = |

**Table 6.14**: Average time per run (in seconds) for CEC2010 Benchmark Problems (10 dimensions)

| Problems | CSMO | QACSMO | Problems | CSMO | QACSMO |
|---|---|---|---|---|---|
| C01 | 1.57704 | 0.6316 | C10 | 0.6866 | 0.77816 |
| C02 | 0.62848 | 0.78756 | C11 | 4.61672 | 0.63064 |
| C03 | 2.99672 | 0.3194 | C12 | 2.44728 | 0.53316 |
| C04 | 2.26852 | 0.54132 | C13 | 0.69888 | 0.78688 |
| C05 | 0.66564 | 0.6808 | C14 | 2.34368 | 0.92004 |
| C06 | 0.9182 | 0.98948 | C15 | 1.22948 | 1.18256 |
| C07 | 4.56004 | 0.54628 | C16 | 0.8308 | 0.85876 |
| C08 | 3.45228 | 0.83192 | C17 | 0.41632 | 0.4794 |
| C09 | 0.39936 | 0.47128 | C18 | 0.67808 | 0.685 |

**Table 6.15:** Feasibility Rate for CEC2010 Benchmark Problems (30 dimensions)

| Problems | CSMO | QACSMO | Problems | CSMO | QACSMO |
|---|---|---|---|---|---|
| C01 | 100 | 100 | C10 | 12 | **28** |
| C02 | 100 | 100 | C11 | **4** | 0 |
| C03 | 0 | 0 | C12 | **92** | 52 |
| C04 | 0 | 0 | C13 | 100 | 100 |
| C05 | 0 | **4** | C14 | 100 | 100 |
| C06 | 0 | **16** | C15 | 100 | 100 |
| C07 | 100 | 100 | C16 | **100** | 84 |
| C08 | 100 | 100 | C17 | **100** | 64 |
| C09 | 16 | 16 | C18 | **100** | 96 |

**Table 6.16**: Best, Median and Worst of objective function values obtained with 25 independent runs for CEC2010 Benchmark Problems (30 dimensions)

| Problems | Best | | Median | | Worst | |
|---|---|---|---|---|---|---|
| | CSMO | QACSMO | CSMO | QACSMO | CSMO | QACSMO |
| C01 | -8.18E-01 | -8.18E-01 | **-8.18E-01** | -8.07E-01 | **-8.04E-01** | -6.61E-01 |
| C02 | **1.38E+00** | 2.93E+00 | **3.03E+00** | 4.58E+00 | **4.05E+00** | 5.48E+00 |
| C03 | **1.12E-03(1)** | 1.25E-02(1) | **2.61E+01(1)** | 3.96E+01(1) | 4.53E+02(1) | **1.94E+02(1)** |
| C04 | **4.14E-03(3)** | 1.71E-02(3) | **4.27E-02(3)** | 1.34E-01(3) | **1.22E+00(4)** | 1.33E+00(4) |
| C05 | 2.32E-04(2) | **2.64E+02** | **9.78E-04(2)** | 6.15E-03(1) | **2.22E-03(2)** | 3.05E-02(2) |
| C06 | 3.52E-04(2) | **1.44E+00** | **2.04E-03(2)** | 1.07E-02(2) | **4.22E-03(2)** | 4.49E-02(2) |
| C07 | **4.72E-04** | 2.29E-01 | **1.37E+01** | 1.49E+01 | 1.03E+02 | **9.04E+01** |
| C08 | **2.11E-03** | 1.19E-01 | **7.99E+01** | 1.99E+02 | 1.51E+04 | **6.44E+03** |
| C09 | 1.78E+13 | **5.96E+12** | **3.36E-04(1)** | 3.91E-03(1) | **1.26E-03(1)** | 1.79E-02(1) |
| C10 | **1.05E+12** | 3.60E+12 | **3.06E-04(1)** | 1.68E-03(1) | **1.53E-03(1)** | 1.37E-02(1) |
| C11 | **2.61E-04** | 5.83E-02(1) | 6.91E+00(1) | **4.03E+00(1)** | 1.38E+02(1) | **6.87E+01(1)** |
| C12 | **-8.85E+02** | -2.03E+02 | **-1.99E-01** | 1.29E+01 | **3.23E-01(1)** | 2.27E+00(1) |
| C13 | **-6.75E+01** | -6.43E+01 | **-6.40E+01** | -5.91E+01 | **-6.19E+01** | -5.71E+01 |
| C14 | **1.30E-02** | 7.15E-02 | **1.28E+01** | 7.01E+01 | **2.87E+03** | 5.02E+07 |
| C15 | 7.57E+12 | **1.28E+12** | **2.60E+13** | 5.14E+13 | **2.22E+13** | 2.32E+14 |
| C16 | 1.06E+00 | 1.06E+00 | **1.11E+00** | 1.12E+00 | **1.17E+00** | 1.15E-04(2) |
| C17 | 8.09E+02 | **5.28E+02** | **1.36E+03** | 1.40E+03 | **2.44E+03** | 1.66E-04(1) |
| C18 | 1.80E+04 | **1.39E+04** | 2.63E+04 | **2.20E+04** | **3.52E+04** | 1.13E-05(1) |

**Table 6.17**: Mean and Standard Deviation of objective function values obtained with 25 independent runs for CEC2010 Benchmark Problems (30 dimensions)

| Problems | Mean | | Stdev | |
|---|---|---|---|---|
| | CSMO | QACSMO | CSMO | QACSMO |
| C01 | **-8.17E-01** | -7.91E-01 | **2.80E-03** | 3.86E-02 |
| C02 | **2.97E+00** | 4.47E+00 | **6.29E-01** | 7.04E-01 |
| C03 | N.A. | N.A. | N.A. | N.A. |
| C04 | N.A. | N.A. | N.A. | N.A. |
| C05 | N.A. | **2.64E+02** | N.A. | N.A. |
| C06 | N.A. | **1.44E+00** | N.A. | **0.00E+00** |
| C07 | 2.60E+01 | **2.14E+01** | 3.50E+01 | **2.61E+01** |
| C08 | **1.09E+03** | 1.10E+03 | 3.23E+03 | **1.81E+03** |
| C09 | 3.36E+13 | **2.44E+13** | **2.15E+13** | 3.07E+13 |
| C10 | 2.59E+13 | **1.46E+13** | 2.33E+13 | **7.44E+12** |
| C11 | **2.61E-04** | N.A. | N.A. | N.A. |
| C12 | **-4.19E+02** | -1.16E+01 | 4.20E+02 | **5.77E+01** |
| C13 | **-6.43E+01** | -6.01E+01 | **1.27E+00** | 1.95E+00 |
| C14 | **1.85E+02** | 2.16E+06 | **5.85E+02** | 1.00E+07 |
| C15 | **2.32E+13** | 6.37E+13 | **1.10E+13** | 5.76E+13 |
| C16 | **1.11E+00** | 1.12E+00 | **2.48E-02** | 3.29E-02 |
| C17 | 1.43E+03 | **1.07E+03** | **3.33E+02** | 3.77E+02 |
| C18 | 2.65E+04 | **2.53E+04** | **4.57E+03** | 1.29E+04 |

**Table 6.18**: Summary of comparison of QACSMO against CSMO for CEC2010 Benchmark Problems (10 dimensions)

| | Criteria | Better | Equal | Worse |
|---|---|---|---|---|
| QACSMO vs. CSMO | Feasibility rate | 3 | 10 | 5 |
| | Function error values | | | |
| | Criteria | Better | Equal | Worse |
| | Best | 6 | 2 | 10 |
| | Median | 2 | 0 | 16 |
| | Worst | 4 | 0 | 14 |
| | Mean | 5 | 0 | 8 |
| | Stdev | 4 | 0 | 9 |

**Table 6.19:** Wilcoxon Rank sum test based on objective function value with a significance level of $\alpha = 0.05$ for CEC2010 Benchmark Problems (30 dimensions) ('+' indicates CSMO is significantly better, '-' indicates QACSMO is significantly worse and '=' indicates there is no significant difference)

| Problems | Sign | Problems | Sign |
|---|---|---|---|
| C01 | - | C10 | - |
| C02 | - | C11 | = |
| C03 | = | C12 | - |
| C04 | = | C13 | - |
| C05 | - | C14 | - |
| C06 | - | C15 | - |
| C07 | = | C16 | = |
| C08 | = | C17 | - |
| C09 | - | C18 | + |

**Table 6.20**: Average time per run (in seconds) for CEC2010 problems (30 dimensions)

| Problems | CSMO | QACSMO | Problems | CSMO | QACSMO |
|---|---|---|---|---|---|
| C01 | 13.98664 | 5.1734 | C10 | 9.17212 | 10.52392 |
| C02 | 5.37496 | 7.01812 | C11 | 47.38996 | 9.16544 |
| C03 | 21.386 | 1.97136 | C12 | 23.08752 | 4.614 |
| C04 | 23.29676 | 4.02116 | C13 | 6.16084 | 6.48596 |
| C05 | 5.84628 | 5.97224 | C14 | 36.29884 | 7.97916 |
| C06 | 11.7292 | 12.37252 | C15 | 14.20968 | 14.20876 |
| C07 | 43.1298 | 4.36836 | C16 | 7.28812 | 7.4746 |
| C08 | 39.4982 | 10.51492 | C17 | 3.55688 | 4.2032 |
| C09 | 3.42036 | 4.13772 | C18 | 5.92072 | 5.96916 |

# CHAPTER 7

# APPLICATION OF SPIDER MONKEY OPTIMIZATION TO SOLVE LENNARD-JONES PROBLEM

Determination of Molecular confirmation is one of the most challenging problems of computational chemistry which can be modelled as a global optimization problem. A molecular conformation problem deals with finding the global minimum of a suitable potential energy function, which depends on relative positions of atoms. Minimization of this energy provides maximum stability for atomic clusters [47]. This energy is the sum of several factors including energy caused by the interaction of two non-bonding atoms. Vander waals interaction is a contributing factor in the energy of interaction between two non-bonding atoms. Vander waals interaction is characterized by the Lennard Jones (L-J) potential function. The main obstacle in solving Lennard-Jones (L-J) problem is the non-linearity and non-convexity of the objective function and exponentially increasing number of local minima with increase in the number of dimensions. Despite these difficulties, solution of this problem is very important to facilitate drug design, synthesis and utilization of pharmaceutical products. So, in the past few years, this problem has attracted several researchers from the field of global optimization to apply their algorithms to solve this problem. In Wille and Vennik [178], it has been shown that complexity of determining global minimum energy of the L-J cluster makes this problem fall in the category of NP-hard problems. This observation has been proved as a motivation for applying metaheuristics to this problem because of their success in solving NP-hard problems in the past few decades. In Deep et.al. [40], L-J problem is solved by applying different variants of Real coded genetic algorithms. In Deep and Madhuri [36], an attempt has been made to solve L-J problem using a variant of PSO named as Globally adaptive inertia weight PSO. In this chapter, L-J problem is solved using SMO, TS-SMO and QASMO. To the best of author’s knowledge, this is the first attempt to solve L-J optimization problem using SMO. Search space for the different clusters has been taken from Deep and Madhuri [36]. The chapter is organised as follows: In section 7.1, a brief introduction to L-J problem is provided. In section 7.2, experimental setup is provided. In section 7.3, the experimental results have been discussed. The chapter is concluded with future scope in section 7.4.

## 7.1 LENNARD-JONES (L-J) PROBLEM

A system containing more than one atom, whose Van der Waals interaction can be  described by L-J potential is called a L-J cluster. L-J problem deals with finding the relative position of atoms in a cluster in the three dimensional Euclidean space in such a way that that potential energy is minimum. Given a cluster of n atoms, L-J problem can be defined mathematically as below [36]:

$$V= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \left(\frac{1}{r_{ij}^{12}} - \frac{2}{r_{ij}^{6}}\right) \tag{7.1}$$

Where $r_{ij}$ is the Euclidean distance between two distinct atoms i and j. Each atom is characterized by a 3-dimensional vector say (a,b,c). So, dimension of each solution of L-J problem will be 3n, where n is the number of atoms in the cluster. Here target is to minimize V. The Potential repels two atoms when they come too close to each other and this behaviour requires a special treatment in molecular dynamics simulation.

## 7.2 EXPERIMENTAL SETUP

Setting for the experiment is given below:

Swarm size = 150

Perturbation rate (*Pr)* = linearly increasing ([0.1, 0.4])

Maximum number of groups (*MG*) =5

local leader limit = 100

global leader limit = 50

Total number of runs = 100

Maximum number of iterations = 4000

Acceptable error = 1.0e-05

Stopping criterion = maximum number of iterations or acceptable error (whichever is achieved earlier)

Tournament size for TS-SMO = 2

Here, error is the absolute difference between the optimal solution and the objective function value of the global leader. In order to make fair comparison between the two algorithms, both the algorithms start with the same initial swarm.

The search space for different number of atoms has been provided in Table 7.1. The problem has been solved using SMO, TS-SMO and QASMO. For comparing the results, success rate, average number of function evaluations for successful runs, best, average and worst of function error values has been recorded.

**Table 7.1**: Dimension and search space for different number of atoms

| Number of atoms | dimension | search space |
|---|---|---|
| 3 | 9 | $[-0.52, 0.45]^{9}$ |
| 4 | 12 | $[-0.52, 0.62]^{12}$ |
| 5 | 15 | $[-0.75, 0.75]^{15}$ |
| 6 | 18 | $[-0.75, 0.75]^{18}$ |
| 7 | 21 | $[-0.96, 0.87]^{21}$ |
| 8 | 24 | $[-0.9, 1.022]^{24}$ |
| 9 | 27 | $[-2, 2]^{27}$ |
| 10 | 30 | $[-2, 2]^{30}$ |

## 7.3 DISCUSSION OF EXPERIMENTAL RESULTS

The numerical and statistical results of the experiment for SMO, TS-SMO and QASMO have been provided in Tables 7.2-7.7. Best entries in the table appear in bold.

Table 7.2 provides the success rate and average number of function evaluations for successful runs. From this table, it can be seen that QASMO has highest success rate among all the three algorithms. It has 100 percent success rate in all the clusters except for the cluster of 9 atoms, while SMO and TS-SMO have 100 percent success rate in two clusters (atoms 3 and 6) and three clusters (atoms 3, 4 and 6) respectively. Also, for average number of function evaluations, QASMO has better performance in six clusters (atoms 3, 4, 5, 6, 7 and 8), while in two clusters (atoms9 and 10), TS-SMO has the better performance.

Tables 7.3-7.5 provide the best, average and worst of function error values of SMO, TS-SMO and QASMO obtained in 100 independent runs. From Table 7.3, it can be seen that SMO has best value in three clusters (atoms 3, 8 and 10) and QASMO has best value in five clusters (atoms 4, 5, 6, 7 and 9). In case of average error values (Table 7.4), SMO has best value for one cluster (atoms 3), TS-SMO for one cluster (atoms 6) and QASMO for 6 clusters (atoms 4, 5, 7, 8, 9, 10).

By concluding the results from Tables 7.2-7.5, it is observed that QASMO has best performance among all the three algorithms in terms of success rate and function error values. Further to see if there is significant difference in the average number of function evaluations for successful runs where all the three algorithms have same success rate (atoms 3 and 6), paired t-test at significance level 0.05 has been applied. The null hypothesis says "there is a not a difference" and alternate hypothesis says "there is a difference". Pairwise t-test has been applied between QASMO vs. SMO and QASMO vs. TS-SMO because QASMO has best performance among all the three. T-test results have been provided in Table 7.6. "+" sign in the cells indicates there is a significant difference between the performance of the two algorithms. Average execution time taken by all the three algorithms has been provided in Table 7.7. It can be observed from the table that the time taken by QASMO is less as compared to other two algorithms on most of the clusters.

PI graph for success rate and average number of function evaluations has been given in Figure 7.1. It can be seen from it that performance of QASMO is better than SMO and TS-SMO for all the values of w. Also, it can be observed that the performance index value increases with the increase in the weight given to success rate.

From the above discussion of results, we can conclude that QSMO is performing better than SMO and TS-SMO in solving Lennard jones problem.

## 7.4 CONCLUSIONS

In this chapter, Lennard-Jones potential problem for 3 to 10 atoms cluster is solved using SMO, TS-SMO and QASMO and the results are compared. Results are best for QASMO for solving L-J problem among the three versions of SMO. In literature, L-J problem has been solved using clusters of large number of atoms. But in this chapter, the clusters of very small number of atoms have been used for the experiment and the results have not been compared with any other metaheuristic algorithm used for solving L-J problem. The reason is the aim of this chapter is to introduce SMO as an optimization tool to the researchers working in the area of chemistry. Moreover, in order to compare the results with any other metaheuristic algorithms, the parameter setting of all the algorithms used for comparison should be same. The performance of metaheuristic algorithms depends heavily on the setting of these control parameters and these parameters can be fine-tuned to perform well on a particular optimization problem. The process of fine tuning the parameters of an optimization algorithm is called meta optimization and such an experiment is avoided here. Because the aim of this chapter is not to find the best algorithm to solve L-J problem, but to see the performance of SMO in solving this problem. Such an experiment will be helpful in analyzing the performance of SMO so that in future, it can be modified in a way to solve L-J problem more efficiently. Also, in future, L-J problem for large number of atoms will be solved using improved versions of SMO and results will be compared with other metaheuristic algorithms also,

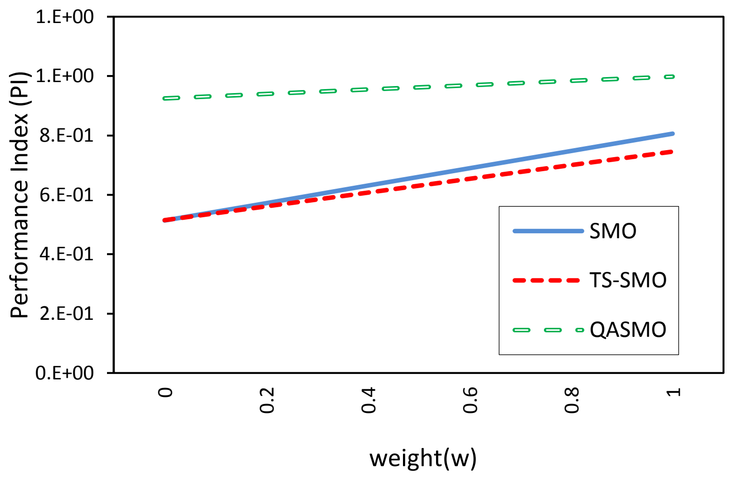

**Figure 7.1:** Performance Index graphs for clusters of atoms

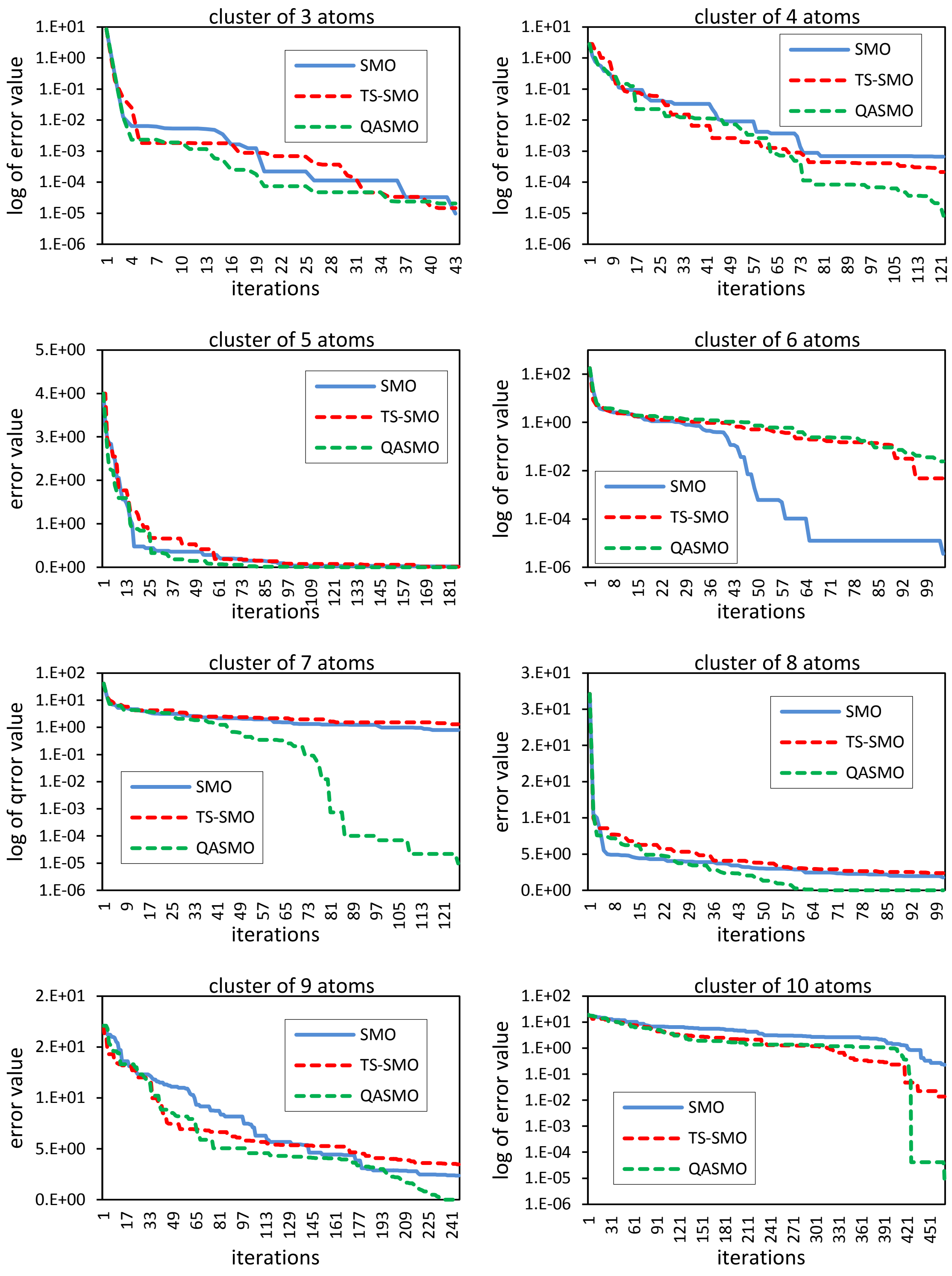

**Figure 7.2**: Convergence graphs for clusters of atoms

**Table 7.2**: Success Rate and Average number of function evaluations for successful runs for SMO, TS-SMO and QASMO (Lennard-Jones problem)

| Number of atoms | Success Rate | | | Average number of function evaluations for successful runs | | |
|---|---|---|---|---|---|---|
| | SMO | TS-SMO | QASMO | SMO | TS-SMO | QASMO |
| 3 | 100 | 100 | 100 | 31486 | 21860 | **22154** |
| 4 | 98 | **100** | **100** | 189065 | 137827 | **36624** |
| 5 | 18 | 21 | **100** | 1177457 | 450678 | **50135** |
| 6 | 100 | 100 | 100 | 144074 | 123027 | **57802** |
| 7 | 97 | 95 | **100** | 385681 | 403569 | **62017** |
| 8 | 94 | 80 | **100** | 378044 | 483973 | **140117** |
| 9 | 58 | 47 | **99** | 974105 | **744380** | 1075422 |
| 10 | 80 | 54 | **100** | 764331 | **709669** | 852578 |

**Table 7.3**: Best of error values obtained in 100 independent runs (Lennard-Jones problem)

| Number of atoms | SMO | TS-SMO | QASMO |
|---|---|---|---|
| 3 | **1.56E-07** | 9.55E-07 | 6.01E-07 |
| 4 | 3.07E-06 | 3.90E-06 | **2.33E-06** |
| 5 | 4.87E-06 | 3.27E-06 | **1.96E-06** |
| 6 | 7.02E-08 | 1.05E-07 | **5.03E-08** |
| 7 | 2.79E-07 | 4.26E-07 | **2.04E-07** |
| 8 | **5.35E-08** | 2.41E-07 | 1.40E-07 |
| 9 | 2.26E-07 | 2.16E-07 | **7.75E-09** |
| 10 | **9.62E-08** | 6.52E-07 | 1.60E-07 |

**Table 7.4**: Average of error values obtained in 100 independent runs (Lennard-Jones problem)

| Number of atoms | SMO | TS-SMO | QASMO |
|---|---|---|---|
| 3 | **5.93E-06** | 6.31E-06 | 6.22E-06 |
| 4 | 9.85E-06 | 8.06E-06 | **7.57E-06** |
| 5 | 6.34E-03 | 2.29E-03 | **8.02E-06** |
| 6 | 5.12E-06 | **4.88E-06** | 6.50E-06 |
| 7 | 9.52E-06 | 2.33E-03 | **6.49E-06** |
| 8 | 1.34E-02 | 9.37E-03 | **5.03E-06** |
| 9 | 1.86E-01 | 1.80E-01 | **9.24E-03** |
| 10 | 1.24E-01 | 3.33E-01 | **4.86E-06** |

**Table 7.5**: Worst of error values obtained in 100 independent runs (Lennard-Jones problem)

| Number of atoms | SMO | TS-SMO | QASMO |
|---|---|---|---|
| 3 | **9.92E-06** | 9.99E-06 | 9.98E-06 |
| 4 | 1.61E-04 | **9.96E-06** | 1.00E-05 |
| 5 | 1.43E-01 | 3.38E-02 | **9.98E-06** |
| 6 | **9.91E-06** | 9.93E-06 | 1.00E-05 |
| 7 | 2.39E-04 | 1.95E-01 | **1.00E-05** |
| 8 | 9.59E-01 | 3.78E-01 | **9.96E-06** |
| 9 | 2.93E+00 | 1.43E+00 | **9.24E-01** |
| 10 | 4.11E+00 | 1.41E+01 | **9.98E-06** |

**Table 7.6**: T-test results for problems having identical success rate for SMO, TS-SMO and QASMO

| Number of atoms | QASMO vs. SMO | QASMO vs. TS-SMO |
|---|---|---|
| 3 | + | + |
| 6 | + | + |

**Table 7.7**: Average execution time per run (in seconds)

| No. of atoms | SMO | TS-SMO | QASMO |
|---|---|---|---|
| 3 | 0.62409 | 0.46143 | **0.44426** |
| 4 | 7.36452 | 5.191 | **1.43584** |
| 5 | 76.16252 | 70.32987 | **3.26166** |
| 6 | 13.9424 | 10.93795 | **5.55685** |
| 7 | 52.12204 | 54.60527 | **8.4648** |
| 8 | 66.66463 | 105.2772 | **25.43765** |
| 9 | 224.8 | **216.8756** | 250.1304 |
| 10 | **215.5962** | 257.3145 | 254.4484 |

# CHAPTER 8

# APPLICATION OF CONSTRAINED SPIDER MONKEY OPTIMIZATION TO SOLVE PORTFOLIO OPTIMIZATION PROBLEM

Portfolio optimization problem has attracted the attention of researchers since ages because of its practical application. This problem is constrained in nature and deals with answering the question what amount of wealth should be invested in a particular asset. In this chapter, an attempt has been made to solve portfolio optimization problem using CSMO proposed in chapter 3. The objective behind this work is the application of CSMO for solving a real world optimization problem. For the experiment purpose, basic mean variance optimization model is considered. To the best of author's knowledge, this is the first attempt to apply SMO for solving a problem in finance.

The chapter is organised as follows: In section 8.1, a brief introduction to portfolio optimization problem is provided. In section 8.2, mean-variance optimization model has been discussed. In section 8.3, experimental setup is provided. In section 8.4, the experimental results have been discussed. The chapter is concluded in section 8.5.

## 8.1 PORTFOLIO OPTIMIZATION PROBLEM

A portfolio is a collection of two or more risky/riskless assets held by an institution or an individual. Suppose a user wants to invest money in n assets. Then its portfolio is represented by n-tuple $\left(x_1, x_{2,} \dots, x_n\right)$, where $x_i$ denotes the amount of fund to be invested in the $i^{th}$ asset. Each of the assets in a portfolio has a return and risk associated with them. Portfolio optimization problem deals with maximizing the profitable returns while minimizing the associated risk of the portfolio. Markowitz [115] was the first to develop an optimization model based on this idea. Since then, various optimization models which are variations of basic Markowitz's model have been developed. Different optimization methods like stochastic optimization method, fuzzy optimization method and robust optimization methods have been applied to solve portfolio optimization problem using these optimization models. Various metaheuristics have been applied to solve different models of portfolio optimization problem [27; 34; 50; 131; 177; 190].

## 8.2 MEAN-VARIANCE MODEL

Markowitz mean-variance model [69] is the basic model for solving portfolio optimization problem. Mathematical formulation for the mean-variance model is given below:

Let return of the $i^{th}$ asset is denoted by a random variable say $R_i$, $x_i$ is the amount of fund to be invested in $i^{th}$ asset.

Asset return is the amount of return which can be calculated for a given period of time. Mathematically it may be defined as

Return = (closing price of current period – closing price of previous period + dividend collect during the period)/ (closing price of previous period)

$$r_{it} = \frac{(p_{it}) - (p_{it-1}) + (d_{it})}{(p_{it-1})}$$

Where $p_{it}$ is the closing price of the asset during the period t

$d_{it}$ is the dividend collected during the period

The aim is to maximize the expected return on the portfolio and minimize the risk.

$$r(x_1, x_2, \dots, x_n) = E[\textstyle\sum_{i=1}^{n} R_i x_i] = \sum_{i=1}^{n} E[R_i] x_i = \sum_{i=1}^{n} r_i x_i\,, \tag{8.1}$$

Where $r_i$ is the expected return on the $i^{th}$ asset and $r_i = E[R_i]$

$$r_i = E[R_i] = \tfrac{1}{T}\textstyle\sum_{t=1}^{T} r_{it} \tag{8.2}$$

The covariance $\sigma_{ij}$ between the asset returns $R_i$ and $R_j$ can be expressed as follows:

$$\sigma_{ij} = E\big[(R_i - E[R_i])(R_j - E[R_j])\big] = \tfrac{1}{T}\textstyle\sum_{t=1}^{T}(r_{it} - r_i)\,(r_{jt} - r_j) \tag{8.3}$$

The portfolio risk is characterized by the variance of returns on that portfolio. The variance of return on a portfolio is then expressed as follows:

$$v(x_1, x_2, \dots, x_n) = \textstyle\sum_{i=1}^{n}\sum_{j=1}^{n} \sigma_{ij} x_i x_j \tag{8.4}$$

The mathematical formulation of the Markowitz's mean variance optimization model is given in model $M$ (8.1).

$$\mathrm{M}(8.1) \qquad \min f(\boldsymbol{x}) = \sum_{i=1}^{n}\sum_{j=1}^{n} \sigma_{ij} x_i x_j$$

Subject to

$$\textstyle\sum_{i=1}^{n} r_i x_i = r_0 \tag{8.5}$$

$$\textstyle\sum_{i=1}^{n} x_i = 1 \tag{8.6}$$

$$x_i \geq 0, \quad i = 1,2,\dots,n \tag{8.7}$$

From the model of the problem M (8.1), it can be seen that it is a constrained optimization problem with equality constraints only. Objective function $f$ is actually the risk $v(x_1, x_2, \dots, x_n)$. $r_0$ in constraint (8.4) denotes the amount of return desired by the investor. The constraint (8.4) makes sure that the expected portfolio return should be equal to the amount of return desired by the investor. Constraint (8.5) represents the capital budget constraint on the assets. Constraint (8.6) makes sure that the value of proportion to be invested in an asset should be non-negative. From these constraints, it can be concluded that the value of $r_0$ cannot be chosen arbitrarily. Though a high portfolio return is always desirable, but aspiring it to be very high can make the problem infeasible. The value of $r_0$ lies between $r_{min}$ and $r_{max}$ . Here, $r_{min}$ is the portfolio return corresponding to the minimum risk. This value can be obtained by solving the problem M (8.1) after removing the constraint represented by the eq. (8.5). $r_{max}$ is the maximum feasible value of $r_0$ and it is given by the maximum mean return among the mean return of all the assets.

## 8.3 EXPERIMENTAL SETUP

### 8.3.1 PARAMETER SETTING AND TERMINATION CRITERIA

Swarm size = 50

perturbation rate *(Pr)* = linearly increasing ([0.1, 0.4])

Maximum number of groups *(MG)* = 5

local leader limit = 1500

global leader limit = 50

Total number of runs = 25

Stopping criterion = 20000 function evaluations

## 8.3.2 OPTIMIZATION MODEL AND INPUT DATA

The mean variance model described in section 8.2 has been taken for the experiment. In order to understand the working of this portfolio optimization model, the data of a real world problem has been taken for illustration purpose. The retail industry has been chosen for experiment because it contributes a big percentage in the gross domestic income. The 11 retail companies, listed on National Stock Exchange (NSE) have been selected as assets to construct portfolios. The list of these companies has been provided in Table 8.1. Our sample data includes the closing prices of these 11 assets from April 1, 2015 to March 31, 2016. The reason behind choosing these particular 11 companies is that our sample data has been extracted from Capitaline and these were the only companies listed on NSE during the financial year 2015-16 whose data was available.

Average monthly returns of these 11 assets are provided in Table 8.2. The expected return, variance and covariance for these assets have been provided in Tables 8.3-Table 8.4 respectively.

Using the optimization model M (8.1) and entries in Tables 8.2-8.4 as input data, the optimization model M (8.2) is formulated:

M(8.2) min f(x) =

$$1.28733x_1x_1+0.32512x_1x_2+0.96732x_1x_3+0.30506x_1x_4+0.44927x_1x_5+0.39730x_1x_6+0.33890$$
$$x_1x_7+0.20332x_1x_8+0.22915x_1x_9+0.42335x_1x_{10}+0.31637*x_1x_{11}+0.84620*x_2x_2+0.08426*$$
$$x_2x_3+0.04178x_2x_4+0.16766x_2x_5+0.32539x_2x_6+0.02563x_2x_7+0.27929x_2x_8-0.20748x_2x_9-$$
$$0.03959x_2x_{10}+0.07751x_2x_{11}+$$
$$1.24506x_3x_3+0.53418x_3x_4+0.51028x_3x_5+0.09637x_3x_6+0.27569x_3x_7+0.11821x_3x_8+0.2392$$
$$6x_3x_9+0.19023x_3x_{10}-0.04089x_3x_{11}+$$
$$0.51375x_4x_4+0.41437x_4x_5+0.13632x_4x_6+0.03128x_4x_7+0.10967x_4x_8+0.09818x_4x_9+0.2444$$
$$5x_4x_{10}+0.00358x_4x_{11}+0.45578x_5x_5+0.24846x_5x_6+0.07455x_5x_7+0.26494x_5x_8+0.07977x_5x_9$$
$$+0.28799x_5x_{10}+0.11858x_5x_{11}+0.64313x_6x_6+0.04808x_6x_7+0.02623x_6x_8+0.00280x_6x_9+0.35$$
$$585x_6x_{10}+0.42780x_6x_{11}+0.14967x_7x_7+0.07379x_7x_8+0.15905x_7x_9+0.03754x_7x_{10}+0.09046$$
$$x_7x_{11}+0.84003x_8x_8+0.00414*x_8x_9+0.40828x_8x_{10}+0.05099x_8x_{11}+0.31191x_9x_9+0.06612$$

$$x_9x_{10}+0.12249x_9x_{11}+0.97290x_{10}x_{10}+0.30167x_{10}x_{11}+0.44687x_{11}x_{11}$$

Such that

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} + x_{11} = 1 \qquad (8.8)$$

$$0.1617x_1 + 0.2972x_2 + 0.4546x_3 + 0.1723x_4 + 0.1189x_5 + 0.0486x_6$$

$$-0.0329x_7 + 0.3958x_8 + 0.0515x_9 + 0.2553x_{10} - 0.0561x_{11} = r_0 \qquad (8.9)$$

$$x_i \geq 0, \quad i = 1,2,\dots,n \qquad (8.10)$$

## 8.4 DISCUSSION OF EXPERIMENTAL RESULTS

In order to solve the model M (8.2), the expected value of return i.e. $r_0$ should be assigned. In section 8.2, it has been mentioned that value of $r_0$ lies between $r_{min}$ and $r_{max}$. So, the above problem has been solved in two parts.

In solution phase I, the range of $r_0$ is determined. $r_{min}$ is calculated by omitting the constraint represented by eq.(8.6) from model M(8.2) and solving the remaining model using CSMO by using the parameter setting and termination criterion described in subsection 8.3.1 of section 8.3. The computational result has been provided in Table 8.5 and based on this result, the value of $r_{min}$ can be calculated using the eq. (8.6). $r_{min}$ is 0.114457. From Table 8.3, it can be seen that $r_{max}$ is 0.4546. Thus, we have obtained the range in which $r_0$ lies.

In solution phase II, the objective function value i.e. risk has been minimized for different values of $r_0$ between $r_{min}$ and $r_{max}$ . Ten uniform random numbers have been generated in $[r_{min}, r_{max}]$. These ten random numbers give ten different values of $r_0$. By using these 10 values of $r_0$ in eq. (8.6) one by one, 10 different portfolios have been generated by solving the model M (8.2) using CSMO and the results have been summarized in Table 8.6. This table contains the expected portfolio return, the proportion of fund to be invested in a particular asset and the associated risk. It can be observed that the portfolio risk level increases with an increase in the expected portfolio return. This relationship always holds in portfolio optimization problem. The average execution time taken by CSMO per run (in seconds) is given in table 8.7. From the table, it can be seen that the execution time for generating these 10 portfolios is very small.

The efficient frontier of the obtained portfolios has been shown in Fig.8.1. X-axis represents the different values of expected returns i.e. $r_0$ and Y-axis represents the associated risk presented in Table 8.7. It can also be seen that as the level of expected return increases, the level of risk also increases.

## 8.5 CONCLUSIONS

CSMO has been applied to solve Markowitz's mean-variance model. The above explained procedure to solve the portfolio optimization problem can be helpful in two ways. First, if the investor has a particular choice for the expected return in advance without having the concern for the associated risk, then the optimal portfolios can be generated directly using the solution phase II. If the investor does not have a particular choice and want to see different possible portfolio returns with associated risks, then the problem can be solved using the above method in which various portfolios can be generated and the investor can choose any portfolio according to his/her choice. In Markowitz's mean variance model, variance has been taken as a measure for risk. In future, other optimization models based on different risk measures can be considered for experiment.

In this chapter, the results of CSMO have not been compared with any other metaheuristic algorithms. The reason which is explained in the forthcoming lines depends upon the case when the investor does not have a predefined choice of portfolio return. It can be seen from the solution procedure explained in section 8.4 that the final solution is obtained after solution phase II and the input for the solution phase II is generated from the results of solution phase I. So, we can't compare different algorithms here because the results generated by different algorithms after solution phase I will be different. Consequently, different algorithms will have different input values for solution phase II and it is not fair to compare the results if the input values are different. But if an investor has a particular expected return in mind, then comparison can be made among different algorithms. But this case has been avoided here because it can be seen as biasness towards the selection of input value.

Also, the model considered for solving portfolio optimization model is the Markowitz's mean variance model which is the most basic model which has limitations also [69]. There are various other advanced models which overcome the limitations of this basic portfolio optimization model with better risk measures for solving portfolio optimization problems [27; 34; 50; 131; 177; 190]. But there are few reasons for choosing this optimization model for experiment in comparison to various other advanced versions of portfolio optimization models. Portfolio optimization problem is one of most prominent optimization problem with varying complexities depending upon the portfolio optimization model under consideration. So, application of SMO for solving it will introduce it to the researchers working in the field of finance for using this algorithm for different types of financial optimization problems. Moreover, in order to develop SMO as a powerful tool for solving portfolio optimization problems, it is necessary to study its behaviour on the basic portfolio optimization models. This study will help in recognizing the strengths and limitations of SMO in solving these problems. These limitations can be overcome and better versions of SMO can be designed for solving different types of financial problems.

**Table 8.1: List of retail companies (assets)**

| **Company** | **NSE Name** | **Allocation of funds** |
|---|---|---|
| Aditya Birla Fashion & Retail Ltd | ABFRL | $x_1$ |
| Cantabil Retail India Ltd | CANTABIL | $x_2$ |
| Future Enterprises-DVR | FELDVR | $x_3$ |
| Future Enterprises Ltd | FEL | $x_4$ |
| Future Lifestyle Fashions Ltd | FLFL | $x_5$ |
| Provogue (India) Ltd | PROVOGE | $x_6$ |
| Shoppers Stop Ltd | SHOPERSTOP | $x_7$ |
| Store One Retail India Ltd | SORILINFRA | $x_8$ |
| Trent Ltd | TRENT | $x_9$ |
| V2 Retail Ltd | V2RETAIL | $x_{10}$ |
| V-Mart Retail Ltd | VMART | $x_{11}$ |

**Table 8.2:** Monthly Return of assets for the period April 01, 2015 to March 31, 2016

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ABFRL | -0.00895 | 2.59500 | -0.03682 | 1.02261 | -0.77381 | 0.73950 | -0.06800 | 0.07895 | 0.27727 | -0.14550 | -2.48429 | 0.74450 |
| CANTABIL | 1.46316 | 0.66400 | -0.60455 | 0.52739 | 0.12667 | -0.29100 | 1.12400 | -0.62474 | 0.07773 | 2.20550 | -1.16619 | 0.06400 |
| FELDVR | 0.43105 | 3.17950 | -0.33682 | 1.04043 | -0.19762 | 0.00000 | -0.07500 | 1.91947 | -0.46091 | -0.26800 | -0.86095 | 1.08400 |
| FEL | 0.87053 | 0.34900 | -0.71182 | 0.74696 | -0.22476 | 0.34000 | -0.10600 | 1.89842 | -0.48773 | -0.23600 | -0.65095 | 0.28000 |
| FLFL | 0.74316 | 0.45600 | -0.67682 | 0.85826 | -1.06905 | 0.20200 | -0.11800 | 1.18526 | -0.07545 | 0.11800 | -0.82048 | 0.62350 |
| PROVOGE | 1.07947 | -0.10900 | -0.46591 | 0.97478 | -0.44667 | -0.32400 | 0.20850 | -0.18632 | 1.66682 | -0.20700 | -1.47667 | -0.13100 |
| SHOPERSTOP | -0.78000 | 0.72700 | -0.06591 | 0.07522 | -0.18857 | -0.10150 | 0.16700 | 0.08895 | 0.15409 | -0.08700 | -0.67952 | 0.29550 |
| SORILINFRA | 0.05158 | -0.14600 | 1.38955 | 1.10696 | -1.29905 | -0.21300 | 0.65100 | 0.94158 | -0.41909 | 1.99850 | -0.47190 | 1.15950 |
| TRENT | -1.31632 | 0.27200 | -0.23000 | 0.36609 | 0.11190 | 0.28400 | 0.34650 | 0.79263 | 0.38591 | -0.56400 | -0.39381 | 0.56350 |
| V2RETAIL | 0.74737 | -0.55200 | 1.73182 | 2.21478 | -0.36762 | 0.27400 | -0.45200 | 0.60053 | 0.10727 | -0.52050 | -1.44905 | 0.72950 |
| VMART | -0.12263 | -0.49450 | -0.02455 | 0.26043 | -0.40476 | 0.19500 | -0.12600 | -0.24947 | 1.61273 | -0.27700 | -1.40762 | 0.36500 |

**Table 8.3:** Expected return of assets

| **Company** | **Average Monthly Return** |
|---|---|
| ABFRL | 0.16171 |
| CANTABIL | 0.29716 |
| FELDVR | 0.45460 |
| FEL | 0.17230 |
| FLFL | 0.11887 |
| PROVOGE | 0.04858 |
| SHOPERSTOP | -0.03290 |
| SORILINFRA | 0.39580 |
| TRENT | 0.05153 |
| V2RETAIL | 0.25534 |
| VMART | -0.05611 |

**Table 8.4:** Variance and Covariance Matrix

| | ABFRL | CANTABIL | FELDVR | FEL | FLFL | PROVOGE | SHOPERSTOP | SORILINFRA | TRENT | V2RETAIL | VMART |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ABFRL | 1.28733 | 0.32512 | 0.96732 | 0.30506 | 0.44927 | 0.39730 | 0.33890 | 0.20332 | 0.22915 | 0.42335 | 0.31637 |
| CANTABIL | 0.32512 | 0.84620 | 0.08426 | 0.04178 | 0.16766 | 0.32540 | 0.02563 | 0.27929 | -0.20748 | -0.03959 | 0.07751 |
| FELDVR | 0.96732 | 0.08426 | 1.24506 | 0.53418 | 0.51028 | 0.09637 | 0.27569 | 0.11821 | 0.23926 | 0.19023 | -0.04089 |
| FEL | 0.30506 | 0.04178 | 0.53418 | 0.51375 | 0.41437 | 0.13632 | 0.03128 | 0.10967 | 0.09818 | 0.24445 | 0.00358 |
| FLFL | 0.44927 | 0.16766 | 0.51028 | 0.41437 | 0.45578 | 0.24846 | 0.07455 | 0.26494 | 0.07977 | 0.28799 | 0.11858 |
| PROVOGE | 0.39730 | 0.32540 | 0.09637 | 0.13632 | 0.24846 | 0.64313 | 0.04808 | 0.02623 | 0.00280 | 0.35585 | 0.42780 |
| SHOPERSTOP | 0.33890 | 0.02563 | 0.27569 | 0.03128 | 0.07455 | 0.04808 | 0.14967 | 0.07379 | 0.15905 | 0.03754 | 0.09046 |
| SORILINFRA | 0.20332 | 0.27929 | 0.11821 | 0.10967 | 0.26494 | 0.02623 | 0.07379 | 0.84003 | 0.00414 | 0.40828 | 0.05100 |
| TRENT | 0.22915 | -0.20748 | 0.23926 | 0.09818 | 0.07977 | 0.00280 | 0.15905 | 0.00414 | 0.31191 | 0.06612 | 0.12249 |
| V2RETAIL | 0.42335 | -0.03959 | 0.19023 | 0.24445 | 0.28799 | 0.35585 | 0.03754 | 0.40828 | 0.06612 | 0.97290 | 0.30167 |
| VMART | 0.31637 | 0.07751 | -0.04089 | 0.00358 | 0.11858 | 0.42780 | 0.09046 | 0.05100 | 0.12249 | 0.30167 | 0.44687 |

**Table 8.5: Result of portfolio selection using variance**

| risk | allocation | value |
|---|---|---|
| 0.12251 | $x_1$ | 0 |
| | $x_2$ | 0.03842 |
| | $x_3$ | 0 |
| | $x_4$ | 0.22077 |
| | $x_5$ | 0.02689 |
| | $x_6$ | 0.03302 |
| | $x_7$ | 0.28191 |
| | $x_8$ | 0.09185 |
| | $x_9$ | 0.03031 |
| | $x_{10}$ | 0.1512 |
| | $x_{11}$ | 0.12555 |

**Table 8.6:** Portfolio set for different values of $r_0$

| | Portfolio return($r_0$) | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ | $x_7$ | $x_8$ | $x_9$ | $x_{10}$ | $x_{11}$ | risk |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Portfolio 1 | 0.114457 | 0 | 0.038417 | 0 | 0.220772 | 0.026886 | 0.033022 | 0.281905 | 0.091854 | 0.03031 | 0.151196 | 0.125553 | 0.12251 |
| Portfolio 2 | 0.133034 | 0.089545 | 0.07102 | 0.042973 | 0.079982 | 0.120765 | 0.095751 | 0.120367 | 0.068455 | 0.145773 | 0.076248 | 0.089052 | 0.17600 |
| Portfolio 3 | 0.325644 | 0.042152 | 0.096549 | 0.073242 | 0.108008 | 0.091606 | 0.082267 | 0.095386 | 0.126766 | 0.131162 | 0.073082 | 0.079719 | 0.18100 |
| Portfolio 4 | 0.274968 | 0.040766 | 0.163346 | 0.210123 | 0.087351 | 0.078322 | 0.0306 | 0.028971 | 0.171482 | 0.040897 | 0.121281 | 0.026785 | 0.27100 |
| Portfolio 5 | 0.421342 | 0.025446 | 0.185471 | 0.252474 | 0.057057 | 0.037203 | 0.028429 | 0.010679 | 0.273721 | 0.020414 | 0.094761 | 0.014278 | 0.30800 |
| Portfolio 6 | 0.170313 | 0.031218 | 0.186193 | 0.26909 | 0.046313 | 0.037022 | 0.024592 | 0.00688 | 0.257036 | 0.023046 | 0.106966 | 0.011569 | 0.32700 |
| Portfolio 7 | 0.322086 | 0.018167 | 0.162767 | 0.296027 | 0.055422 | 0.024654 | 0.028249 | 0.018962 | 0.289006 | 0.013181 | 0.084079 | 0.009408 | 0.32900 |
| Portfolio 8 | 0.414408 | 0.002333 | 0.042986 | 0.503342 | 0.006629 | 0.003965 | 0.001755 | 0.001675 | 0.425676 | 0.001352 | 0.008998 | 0.001209 | 0.53500 |
| Portfolio 9 | 0.335115 | 0.002656 | 0.030453 | 0.562387 | 0.002418 | 0.001518 | 0.00113 | 0.001068 | 0.388639 | 0.001028 | 0.007039 | 0.001603 | 0.58000 |
| Portfolio 10 | 0.4546 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1.24000 |

**Table 8.7:** Average execution time taken by CSMO per run (in seconds)

| **Portfolio** | **Execution time** |
|---|---|
| Portfolio 1 | 0.14 |
| Portfolio 2 | 0.13624 |
| Portfolio 3 | 0.13688 |
| Portfolio 4 | 0.14124 |
| Portfolio 5 | 0.13688 |
| Portfolio 6 | 0.13816 |
| Portfolio 7 | 0.13812 |
| Portfolio 8 | 0.14008 |
| Portfolio 9 | 0.14064 |
| Portfolio 10 | 0.14376 |

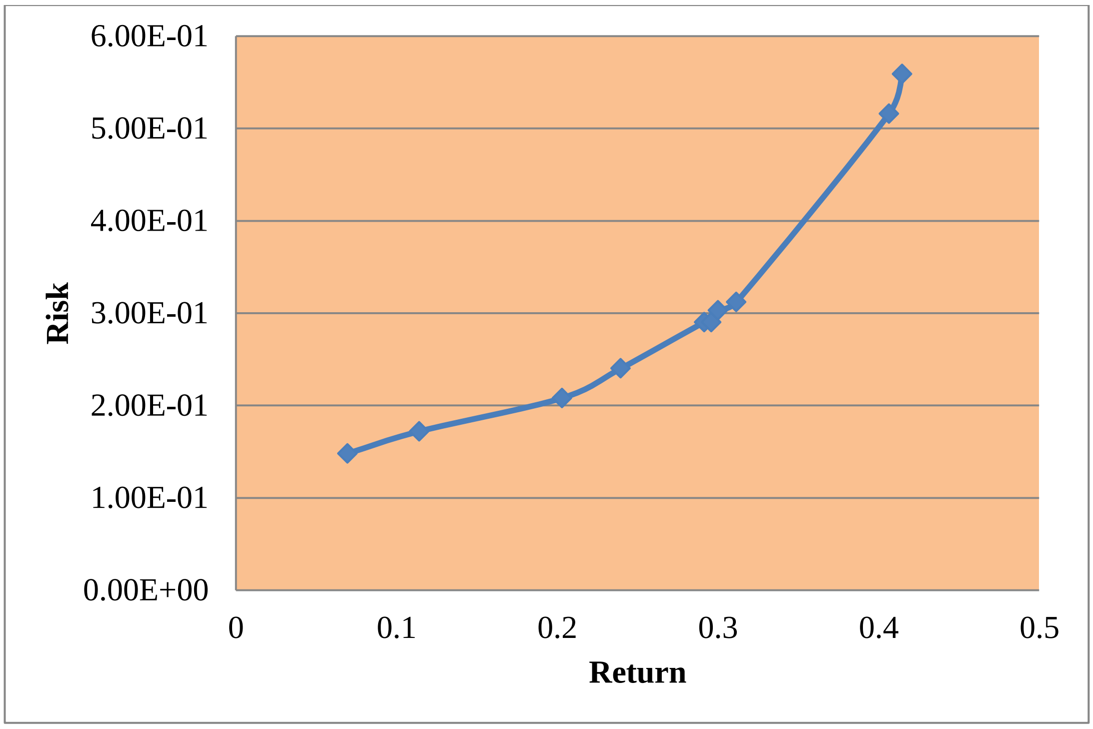

**Figure 8.1:** Efficient Frontier

# CHAPTER 9

# CONCLUSIONS AND FUTURE SCOPE

This chapter forms the concluding part of the thesis as well as proposes possible future directions to continue the research in the present work. The chapter has been divided into two sections: Section 9.1 presents the conclusions obtained from the overall research work done in the present thesis and section 9.2 suggests the possible future directions based on those conclusions.

## 9.1 CONCLUSIONS

The primary goal of the thesis is to propose new versions of basic SMO for solving constrained optimization problems, to improve the performance of basic SMO for unconstrained optimization problems, and to apply the proposed versions for solving real world unconstrained and constrained optimization problems. This thesis proposes four version of SMO including two versions for unconstrained optimization problems and other two for constrained optimization problems.

The basic SMO had been designed for solving unconstrained optimization problems. In order to extend its ability for solving constrained optimization problems, a new version of SMO called Constrained Spider Monkey Optimization (CSMO) algorithm has been proposed which uses Deb’s technique as a constraint handling mechanism. CSMO has been designed with minimal modifications in basic SMO in such a way that the original structure of SMO remains intact. CSMO has been investigated over the constrained benchmark problems of IEEE CEC sessions 2006 and 2010. For comparison, three state-of-the-art algorithms namely ABC, DE and PSO have been used. The results have been discussed in various aspects. Rigours analysis has been done for both the benchmark sets separately. The results have been presented numerically and graphically. Also, a statistical test is employed to validate the significance of results. It has been concluded that CSMO has outperformed all the other three algorithms on both the benchmark sets. Such an outcome of results demonstrates that CSMO is a good global optimizer for constrained optimization problems.

Two new versions of basic SMO have been proposed to improve its performance for unconstrained optimization problems. The first version namely TS-SMO is proposed by replacing the fitness based probability scheme in basic SMO with tournament selection based probability scheme with an objective to make the exploration ability of basic SMO stronger. In fitness based probability scheme, the selection of the solution to be updated is always biased towards the highly fit individuals thus leading to the problem of premature convergence. But, in tournament selection based probability scheme this problem is avoided. The results have been discussed to check the reliability, efficiency and accuracy of the proposed algorithm and it has been concluded that the proposed modification has a positive impact on the performance of SMO. The second version namely QASMO has been designed by implementing quadratic approximation operator in it. The objective of this modification is to improve the exploitation ability of basic SMO. QA has been incorporated in global leader learning phase and local leader learning phase as a local search method. The discussion of results demonstrates that incorporation of QA has a positive impact on the performance of basic SMO. Also, a comparison has been made among SMO, TS-SMO and QASMO and it has been concluded that QASMO is outperforming the other two algorithms on the benchmark set of problems under consideration.

Based on the conclusion made for unconstrained optimization problems, QA has been incorporated in CSMO also to study its impact for solving constrained optimization problems. The discussion of results shows that incorporation of QA has negative impact on the performance of CSMO. From this, we can conclude that we cannot generalize the results for different types of optimization problems.

Below is the overall conclusion of the proposed algorithms over the set of benchmark functions under consideration:

**CSMO**: A good constrained optimizer for scalable as well as non-scalable problems.

**TS-SMO**: A good optimizer with better exploration ability than basic SMO for unconstrained optimization problems.

**QASMO**: a good optimizer with better exploitation ability than basic SMO for unconstrained optimization problems.

**QACSMO:** not a good optimizer for constrained optimization problems.

Real life applications of the proposed algorithms have been carried out. SMO, TS-SMO and QASMO have been applied to solve Lennard-Jones problem, which is a real world unconstrained optimization problem. This problem deals with finding the position of atoms in a cluster so that the overall potential energy is minimum. For the experiment purpose, clusters of atoms ranging from 3 to 10 atoms have been considered. Comparison has been made among SMO, TS-SMO and QASMO and it has been concluded that QASMO performs better than SMO and TS-SMO. CSMO has been applied to solve portfolio optimization problem which is a real world constrained optimization problem. Markowitz model has been considered for portfolio optimization problem.

## 9.2 FUTURE SCOPE

Research is an ever evolving process and the work presented in this thesis can be expanded further. The possible future directions are given below:

### 9.2.1 ON CSMO

- Deb’s technique has been used as a constraint handling mechanism. Different constraint handling techniques can be employed.
- It has problems while solving benchmark problems with equality constraint, modifications can be made to improve its performance for solving these type of problems.

### 9.2.2 ON TS-SMO

- A different probability scheme can be employed for the better selection of solutions.

### 9.2.3 ON QASMO

- A different local search operator can be employed to improve the exploitation ability of SMO.

### 9.2.4 ON QACSMO

- Different ways can be employed to improve the performance.

In addition to the aforementioned future directions, following aspects of SMO can also be explored.

- Till yet, SMO has been applied for solving single objection optimization problems only. New versions of SMO can be developed for solving multi-objective optimization problems.

- No theoretical study has been performed on SMO so far. Work can be done in this direction.

- SMO can be hybridized with other metaheuristic algorithms like GA, PSO, ABC, DE etc.

- So far, modifications in SMO have been proposed without finding its limitations. So, limitations of SMO are needed to be pointed out so that work can be done on those limitations to improve the performance of SMO.

- SMO can be applied to solve various real world optimization problems arising in industry, science, engineering, economics etc.

Although SMO has great potential, it was clear to the scientific community that some modifications to the original structure are still necessary in order to significantly improve its performance. And also SMO can be used as an evolutionary framework into which different traditional or modern heuristic algorithmic components are integrated. SMO can be also applied for optimization in dynamic and uncertain environments. In order to improve the performance of SMO in terms of convergence, new neighbour production mechanisms can be proposed.

# BIBLIOGRAPHY

[1] Agarwal, N., & Jain, S. C. (2017). Fast Convergent Spider Monkey Optimization Algorithm. In *Proceedings of Sixth International Conference on Soft Computing for Problem Solving* (pp. 42-51). Springer, Singapore.

[2] Agrawal, A., Farswan, P., Agrawal, V., Tiwari, D. C., & Bansal, J. C. (2017). On the Hybridization of Spider Monkey Optimization and Genetic Algorithms. In *Proceedings of Sixth International Conference on Soft Computing for Problem Solving* (pp. 185-196). Springer, Singapore.

[3] Al-Azza, A. A., Al-Jodah, A. A., & Harackiewicz, F. J. (2016). Spider Monkey Optimization: A Novel Technique for Antenna Optimization. *IEEE Antennas and Wireless Propagation Letters*, *15*, 1016-1019.

[4] Ali, A. F. (2017). An Improved Spider Monkey Optimization for Solving a Convex Economic Dispatch Problem. In *Nature-Inspired Computing and Optimization* (pp. 425-448). Springer International Publishing.

[5] Ali, L., & Sabat, S. L. (2012). Particle swarm optimization based universal solver for global optimization. *Journal of Bioinformatics and Intelligent Control*, *1*(1), 95-105.

[6] Ali, M. M. (2007). Differential evolution with preferential crossover. *European Journal of Operational Research*, *181*(3), 1137-1147.

[7] Ali, M. M., & Kaelo, P. (2008). Improved particle swarm algorithms for global optimization. *Applied mathematics and computation*, *196*(2), 578-593.

[8] Ali, M. M., & Törn, A. (2004). Population set-based global optimization algorithms: some modifications and numerical studies. *Computers & Operations Research*, *31*(10), 1703-1725.

[9] Ali, M., Ahn, C. W., & Pant, M. (2014). Multi-level image thresholding by synergetic differential evolution. *Applied Soft Computing*, *17*, 1-11.

[10] Ali, M., Pant, M., & Nagar, A. (2010, September). Two local search strategies for differential evolution. In *Bio-Inspired Computing: Theories and Applications (BIC-TA), 2010 IEEE Fifth International Conference on* (pp. 1429-1435). IEEE.

[11] Al-Obaidi, A. T. S., & Abdullah, H. S. (2017). Camel Herds Algorithm: a New Swarm Intelligent Algorithm to Solve Optimization Problems. *International Journal on Perceptive and Cognitive Computing*, *3*(1).

[12] Arias-Montano, A., Coello, C. A. C., & Mezura-Montes, E. (2012). Multiobjective evolutionary algorithms in aeronautical and aerospace engineering. *IEEE Transactions on Evolutionary Computation*, *16*(5), 662-694.

[13] Arora, V., Sood, P., & Keshari, K. U. (2015, December). A comparison of HPSOWM, krill herd and Spider Monkey optimization algorithms. In *Recent Advances in Engineering & Computational Sciences (RAECS), 2015 2nd International Conference on* (pp. 1-5). IEEE.

[14] Asafuddoula, M., Ray, T., & Sarker, R. (2015). A differential evolution algorithm with constraint sequencing: An efficient approach for problems with inequality constraints. *Applied Soft Computing*, *36*, 101-113.

[15] Askarzadeh, A. (2016). A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. *Computers & Structures*, *169*, 1-12.

[16] Bansal, J. C., Sharma, H., Arya, K. V., & Nagar, A. (2013). Memetic search in artificial bee colony algorithm. *Soft Computing*, *17*(10), 1911-1928.

[17] Bansal, J. C., Sharma, H., Jadon, S. S., & Clerc, M. (2014). Spider monkey optimization algorithm for numerical optimization. *Memetic computing*, *6*(1), 31-47.

[18] Bansal, J. C., Sharma, H., Nagar, A., & Arya, K. V. (2013). Balanced artificial bee colony algorithm. *International Journal of Artificial Intelligence and Soft Computing*, *3*(3), 222-243.

[19] Bedi, P., Bansal, R., & Sehgal, P. (2012). Multimodal biometric authentication using PSO based watermarking. Procedia Technology, 4, 612-618.

[20] Bedi, P., Bansal, R., & Sehgal, P. (2013). Using PSO in a spatial domain based image hiding scheme with distortion tolerance. Computers & Electrical Engineering, 39(2), 640-654.

[21] Bhardwaj, A., & Tiwari, A. (2015). Breast cancer diagnosis using genetically optimized neural network model. *Expert Systems with Applications*, *42*(10), 4611-4620.

[22] Bhardwaj, A., Tiwari, A., Chandarana, D., & Babel, D. (2014, October). A genetically optimized neural network for classification of breast cancer disease. In *Biomedical Engineering and Informatics (BMEI), 2014 7th International Conference on* (pp. 693-698). IEEE.

[23] Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). *Swarm intelligence: from natural to artificial systems* (No. 1). Oxford university press.

[24] Boussaid, I., Chatterjee, A., Siarry, P., & Ahmed-Nacer, M. (2011). Hybridizing biogeography-based optimization with differential evolution for optimal power allocation in wireless sensor networks. *IEEE Transactions on Vehicular Technology*, *60*(5), 2347-2353.

[25] Boussaïd, I., Chatterjee, A., Siarry, P., & Ahmed-Nacer, M. (2012). Biogeography-based optimization for constrained optimization problems. *Computers & Operations Research*, *39*(12), 3293-3304.

[26] Chandra, S., & Jayadeva, M. (2009). A.: Numerical Optimization with Applications. Alpha Science International.

[27] Chang, T. J., Yang, S. C., & Chang, K. J. (2009). Portfolio optimization problems in different risk measures using genetic algorithm. *Expert Systems with Applications*, *36*(7), 10529-10537.

[28] Chatterjee, A., & Siarry, P. (2006). Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization. *Computers & Operations Research*, *33*(3), 859-871.

[29] Chelouah, R., & Siarry, P. (2000). A continuous genetic algorithm designed for the global optimization of multimodal functions. *Journal of Heuristics*, *6*(2), 191-213.

[30] Chelouah, R., & Siarry, P. (2000). Tabu search applied to global optimization. *European journal of operational research*, *123*(2), 256-270.

[31] Cheruku, R., Edla, D. R., & Kuppili, V. (2017). SM-RuleMiner: Spider monkey based rule miner using novel fitness function for diabetes classification. *Computers in Biology and Medicine*, *81*, 79-92.

[32] Clerc, M. (2009). A method to improve standard PSO. Retrieved from http://clerc.maurice.free.fr/pso/Design_efficient_PSO.pdf.

[33] Coello, C. A. C. (2002). Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Computer methods in applied mechanics and engineering*, *191*(11), 1245-1287.

[34] Cura, T. (2009). Particle swarm optimization approach to portfolio optimization. *Nonlinear analysis: Real world applications*, *10*(4), 2396-2406.

[35] Deb, K. (2000). An efficient constraint handling method for genetic algorithms. *Computer methods in applied mechanics and engineering*, *186*(2), 311-338.

[36] Deep, K. & Madhuri (2012). Application of Globally Adaptive Inertia Weight PSO to Lennard-Jones Problem. In *Proceedings of the International Conference on Soft Computing for Problem Solving (SocProS 2011) December 20-22, 2011* (pp. 31-38). Springer India.

[37] Deep, K., & Bansal, J. C. (2009). Hybridization of particle swarm optimization with quadratic approximation. *Opsearch*, *46*(1), 3-24.

[38] Deep, K., & Das, K. N. (2008). Quadratic approximation based hybrid genetic algorithm for function optimization. *Applied Mathematics and Computation*, *203*(1), 86-98.

[39] Deep, K., & Thakur, M. (2007). A new crossover operator for real coded genetic algorithms. *Applied mathematics and computation*, *188*(1), 895-911.

[40] Deep, K., Shashi, & Katiyar, V. K. (2011, June). Global optimization of lennard-jones potential using newly developed real coded genetic algorithms. In *Communication Systems and Network Technologies (CSNT), 2011 International Conference on* (pp. 614-618). IEEE.

[41] Dhar, J., & Arora, S. (2017). Designing Fuzzy Rule Base using Spider Monkey Optimization Algorithm in Cooperative Framework. *Future Computing and Informatics Journal*. https://doi.org/10.1016/j.fcij.2017.04.004.

[42] Dib, N., & Sharaqa, A. (2014). Synthesis of thinned concentric circular antenna arrays using teaching-learning-based optimization. *International Journal of RF and Microwave Computer‐Aided Engineering*, *24*(4), 443-450.

[43] Dixon, L. C. W. (1978). The global optimization problem: an introduction. *Towards Global Optimiation 2*, 1-15.

[44] Doğan, B., & Ölmez, T. (2015). A new metaheuristic for numerical function optimization: Vortex Search algorithm. *Information Sciences*, *293*, 125-145.

[45] Dorigo, M. (1992). Optimization, learning and natural algorithms. *Ph. D. Thesis, Politecnico di Milano, Italy*.

[46] Dorigo, M., Maniezzo, V., & Colorni, A. (1991). The ant system: An autocatalytic optimizing process.

[47] Doye, J. P. K. (1996). *The structure, thermodynamics and dynamics of atomic clusters* (Doctoral dissertation, University of Cambridge).

[48] Duvvuru, N., & Swarup, K. S. (2011). A hybrid interior point assisted differential evolution algorithm for economic dispatch. *IEEE Transactions on Power Systems*, *26*(2), 541-549.

[49] Eberhart, R., & Kennedy, J. (1995, October). A new optimizer using particle swarm theory. In *Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on* (pp. 39-43). IEEE.

[50] El-Bizri, S., & Mansour, N. (2017, July). Metaheuristics for Portfolio Optimization. In *International Conference in Swarm Intelligence* (pp. 77-84). Springer, Cham.

[51] Elsayed, S. M., Sarker, R. A., & Essam, D. L. (2011, June). GA with a new multi-parent crossover for constrained optimization. In *Evolutionary Computation (CEC), 2011 IEEE Congress on* (pp. 857-864). IEEE.

[52] Esmaeili, R., & Dashtbayazi, M. R. (2014). Modeling and optimization for microstructural properties of Al/SiC nanocomposite by artificial neural network and genetic algorithm. *Expert Systems with Applications*, *41*(13), 5817-5831.

[53] Farmer, J. D., Packard, N. H., & Perelson, A. S. (1986). The immune system, adaptation, and machine learning. *Physica D: Nonlinear Phenomena*, *22*(1-3), 187-204.

[54] Formato, R. A. (2008). Central force optimization: A new nature inspired computational framework for multidimensional search and optimization. In *Nature Inspired Cooperative Strategies for Optimization (NICSO 2007)* (pp. 221-238). Springer Berlin Heidelberg.

[55] Formato, R. A. (2009). Central force optimization: A new deterministic gradient-like optimization metaheuristic. *Opsearch*, *46*(1), 25-51.

[56] Formato, R. A. (2010). Improved CFO algorithm for antenna optimization. *Progress In Electromagnetics Research B*, *19*, 405-425.

[57] Gandomi, A. H., & Alavi, A. H. (2012). Krill herd: a new bio-inspired optimization algorithm. *Communications in Nonlinear Science and Numerical Simulation*, *17*(12), 4831-4845.

[58] Gandomi, A. H., Yang, X. S., Alavi, A. H., & Talatahari, S. (2013). Bat algorithm for constrained optimization tasks. *Neural Computing and Applications*, *22*(6), 1239-1255.

[59] Gao, W. F., Huang, L. L., Liu, S. Y., & Dai, C. (2015). Artificial bee colony algorithm based on information learning. *IEEE transactions on cybernetics*, *45*(12), 2827-2839.

[60] Gao, W. F., Liu, S. Y., & Huang, L. L. (2014). Enhancing artificial bee colony algorithm using more information-based search equations. *Information Sciences*, *270*, 112-133.

[61] Geem, Z. W. (2009). Particle-swarm harmony search for water network design. *Engineering Optimization*, *41*(4), 297-311.

[62] Geem, Z. W., Kim, J. H., & Loganathan, G. V. (2001). A new heuristic optimization algorithm: harmony search. *Simulation*, *76*(2), 60-68.

[63] Geem, Z. W., Kim, J. H., & Loganathan, G. V. (2002). Harmony search optimization: application to pipe network design. *International Journal of Modelling and Simulation*, *22*(2), 125-133.

[64] Ghosh, P., & Das, S. (2011). Synthesis of thinned planar concentric circular antenna arrays---A differential evolutionary approach. *Progress In Electromagnetics Research B*, *29*, 63-82.

[65] Glover, F. (1989). Tabu search—part I. *ORSA Journal on computing*, *1*(3), 190-206.

[66] Goel, S., Sharma, A., & Bedi, P. (2011, December). Cuckoo Search Clustering Algorithm: A novel strategy of biomimicry. In Information and Communication Technologies (WICT), 2011 World Congress on (pp. 916-921). IEEE.

[67] Gogna, A., & Tayal, A. (2013). Metaheuristics: review and application. *Journal of Experimental & Theoretical Artificial Intelligence*, *25*(4), 503-526.

[68] Green, R. C., Wang, L., Alam, M., & Formato, R. A. (2012). Central force optimization on a GPU: a case study in high performance metaheuristics. *The Journal of Supercomputing*, *62*(1), 378-398.

[69] Gupta, P., Mehlawat, M. K., Inuiguchi, M., & Chandra, S. (2014). Fuzzy portfolio optimization. *Studies in fuzziness and soft computing*, *316*.

[70] Han, M., Liu, C., & Xing, J. (2014). An evolutionary membrane algorithm for global numerical optimization problems. *Information Sciences*, *276*, 219-241.

[71] Hansen, N. (2006). The CMA evolution strategy: a comparing review. *Towards a new evolutionary computation*, 75-102.

[72] Hansen, N., & Ostermeier, A. (1996, May). Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. In *Evolutionary Computation, 1996., Proceedings of IEEE International Conference on* (pp. 312-317). IEEE.

[73] Hazrati, G., Sharma, H., & Sharma, N. (2016, July). Adaptive step-size based Spider Monkey Optimization. In Power Electronics, Intelligent Control and Energy Systems (ICPEICES), IEEE International Conference on (pp. 1-5). IEEE.

[74] Hazrati, G., Sharma, H., Sharma, N., Agarwal, V., & Tiwari, D. C. (2017). Spider Monkey Optimization Algorithm Based on Metropolis Principle. In Proceedings of Sixth International Conference on Soft Computing for Problem Solving (pp. 113-121). Springer, Singapore.

[75] He, Q., & Wang, L. (2007). A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization. *Applied mathematics and computation*, *186*(2), 1407-1422.

[76] Himmelblau, D. M. (1972). *Applied nonlinear programming*. McGraw-Hill Companies.

[77] Holland, J. H. (1975). Adaption in natural and artificial systems. *Ann Arbor MI: The University of Michigan Press*.

[78] Jaddi, N. S., Alvankarian, J., & Abdullah, S. (2017). Kidney-inspired algorithm for optimization problems. *Communications in Nonlinear Science and Numerical Simulation*, *42*, 358-369.

[79] Jamuna, K., & Swarup, K. S. (2011). Biogeography based optimization for optimal meter placement for security constrained state estimation. *Swarm and Evolutionary Computation*, *1*(2), 89-96.

[80] Jana, N. D., & Sil, J. (2016). Levy distributed parameter control in differential evolution for numerical optimization. *Natural Computing*, *15*(3), 371-384.

[81] Jha, P. C., & Hoda, M. N. (2011). *Mathematical Modelling, Optimization and Their Applications*. Narosa.

[82] Jia, G., Wang, Y., Cai, Z., & Jin, Y. (2013). An improved (μ+ λ)-constrained differential evolution for constrained optimization. *Information Sciences*, *222*, 302-322.

[83] Kaboli, S. H. A., Selvaraj, J., & Rahim, N. A. (2017). Rain-fall optimization algorithm: A population based algorithm for solving constrained optimization problems. *Journal of Computational Science*, *19*, 31-42.

[84] Kaelo, P., & Ali, M. M. (2007). Differential evolution algorithms using hybrid mutation. *Computational Optimization and Applications*, *37*(2), 231-246.

[85] Kar, A. K. (2016). Bio inspired computing–A review of algorithms and scope of applications. *Expert Systems with Applications*, *59*, 20-32.

[86] Karaboga, D. (2005). *An idea based on honey bee swarm for numerical optimization* (Vol. 200). Technical report-tr06, Erciyes university, engineering faculty, computer engineering department.

[87] Karaboga, D., & Akay, B. (2011). A modified artificial bee colony (ABC) algorithm for constrained optimization problems. *Applied Soft Computing*, *11*(3), 3021-3031.

[88] Kashan, A. H. (2015). A new metaheuristic for optimization: optics inspired optimization (OIO). *Computers & Operations Research*, *55*, 99-125.

[89] Kaur, A. (2016). Comparison Analysis of CDMA Multiuser Detection using PSO and SMO. *International Journal of Computer Applications*, *133*(2), 47-50.

[90] Kennedy, J., & Eberhart, R. (1995, November). Particle swarm optimization (PSO). In *Proc. IEEE International Conference on Neural Networks, Perth, Australia* (pp. 1942-1948).

[91] Khajehzadeh, M., Taha, M. R., El-Shafie, A., & Eslami, M. (2012). A modified gravitational search algorithm for slope stability analysis. *Engineering Applications of Artificial Intelligence*, *25*(8), 1589-1597.

[92] Khazali, A. H., & Kalantar, M. (2011). Optimal reactive power dispatch based on harmony search algorithm. *International Journal of Electrical Power & Energy Systems*, *33*(3), 684-692.

[93] Khoa, T. H., Vasant, P. M., Singh, M. B., & Dieu, V. N. (2014, December). Swarm based mean-variance mapping optimization (MVMO s) for economic dispatch problem with valve—Point effects. In *Industrial Engineering and Engineering Management (IEEM), 2014 IEEE International Conference on* (pp. 59-63). IEEE.

[94] Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *science*, *220*(4598), 671-680.

[95] Kowsalya, M. (2014). Optimal size and siting of multiple distributed generators in distribution system using bacterial foraging optimization. *Swarm and Evolutionary computation*, *15*, 58-65.

[96] Krishnanand, K. N., & Ghose, D. (2008). Theoretical foundations for rendezvous of glowworm-inspired agent swarms at multiple locations. *Robotics and Autonomous Systems*, *56*(7), 549-569.

[97] Krishnanand, K. N., & Ghose, D. (2009). Glowworm swarm optimisation: a new method for optimising multi-modal functions. *International Journal of Computational Intelligence Studies*, *1*(1), 93-119.

[98] Krishnanand, K. N., & Ghose, D. (2009). Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions. *Swarm intelligence*, *3*(2), 87-124.

[99] Kumar, R., Sharma, D., & Sadu, A. (2011). A hybrid multi-agent based particle swarm optimization algorithm for economic power dispatch. *International journal of electrical power & energy systems*, *33*(1), 115-123.

[100] Kumar, S., & Kumari, R. (2014). Modified position update in spider monkey optimization algorithm. In *International Journal of Emerging Technologies in Computational and Applied Sciences (IJETCAS*.

[101] Kumar, S., Kumar Sharma, V., & Kumari, R. (2014). Self-Adaptive Spider Monkey Optimization Algorithm for Engineering Optimization Problems. *Int. J. Information, Commun. Comput. Technol. II*, 96-107.

[102] Kumar, S., Kumari, R., & Sharma, V. K. (2015). Fitness Based Position Update in Spider Monkey Optimization Algorithm. *Procedia Computer Science*, *62*, 442-449.

[103] Kumar, V., Chhabra, J. K., & Kumar, D. (2014). Automatic cluster evolution using gravitational search algorithm and its application on image segmentation. *Engineering Applications of Artificial Intelligence*, *29*, 93-103.

[104] Kuo, R. J., Hung, S. Y., & Cheng, W. C. (2014). Application of an optimization artificial immune network and particle swarm optimization-based fuzzy neural network to an RFID-based positioning system. *Information Sciences*, *262*, 78-98.

[105] Lalwani, S., Kumar, R., & Gupta, N. (2013). A review on particle swarm optimization variants and their applications to multiple sequence alignment. *Journal of Applied Mathematics and Bioinformatics*, *3*(2), 87.

[106] Lenin, K., Reddy, B. R., & Kalavathi, M. S. (2015). Modified monkey optimization algorithm for solving optimal reactive power dispatch problem. *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, *3*(2), 55-62.

[107] Liang, J. J., Runarsson, T. P., Mezura-Montes, E., Clerc, M., Suganthan, P. N., Coello, C. C., & Deb, K. (2006). Problem definitions and evaluation criteria for the CEC 2006 special session on constrained real-parameter optimization. *Journal of Applied Mechanics*, *41*(8).

[108] Ling, S. H., Iu, H. H., Chan, K. Y., Lam, H. K., Yeung, B. C., & Leung, F. H. (2008). Hybrid particle swarm optimization with wavelet mutation and its industrial applications. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *38*(3), 743-763.

[109] Liu, R., Wang, L., Ma, W., Mu, C., & Jiao, L. (2014). Quadratic interpolation based orthogonal learning particle swarm optimization algorithm. *Natural Computing*, *13*(1), 17-37.

[110] Lozano, J. A. (2006). *Towards a new evolutionary computation: advances on estimation of distribution algorithms* (Vol. 192). Springer Science & Business Media.

[111] Mahadevan, K., & Kannan, P. S. (2010). Comprehensive learning particle swarm optimization for reactive power dispatch. *Applied soft computing*, *10*(2), 641-652.

[112] Mallipeddi, R., & Suganthan, P. N. (2010). Problem definitions and evaluation criteria for the CEC 2010 competition on constrained real-parameter optimization. *Nanyang Technological University, Singapore*.

[113] Maniezzo, A. C. M. D. V. (1992). Distributed optimization by ant colonies. In *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life* (p. 134). Mit Press.

[114] Manik, P., Gupta, A., & Jha, P. C. (2012). Differential evolution approach to promotional effort allocation in segmented market for multi-period promotion strategies

in a planning horizon incorporating repeat purchase. In *Proceedings of the International Conference on Soft Computing for Problem Solving (SocProS 2011) December 20-22, 2011* (pp. 979-991). Springer India.

[115] Markowitz, H. (1952). Portfolio selection. *The journal of finance*, *7*(1), 77-91.

[116] Mezura-Montes, E., & Cetina-Domínguez, O. (2012). Empirical analysis of a modified artificial bee colony for constrained numerical optimization. *Applied Mathematics and Computation*, *218*(22), 10943-10973.

[117] Mezura-Montes, E., & Hernández-Ocaña, B. (2009). Modified bacterial foraging optimization for engineering design. In *Intelligent engineering systems through artificial neural networks*. ASME Press.

[118] Mezura-Montes, E., Coello, C. A. C., & Tun-Morales, E. I. (2004, April). Simple feasibility rules and differential evolution for constrained optimization. In *Mexican International Conference on Artificial Intelligence* (pp. 707-716). Springer Berlin Heidelberg.

[119] Mezura-Montes, E., Velázquez-Reyes, J., & Coello Coello, C. A. (2005, June). Promising infeasibility and multiple offspring incorporated to differential evolution for constrained optimization. In *Proceedings of the 7th annual conference on Genetic and evolutionary computation* (pp. 225-232). ACM.

[120] Mezura-Montes, E., Velázquez-Reyes, J., & Coello, C. C. (2006, July). Modified differential evolution for constrained optimization. In *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on* (pp. 25-32). IEEE.

[121] Millonas, M. M. (1994). Swarms, phase transitions, and collective intelligence. In *SANTA FE INSTITUTE STUDIES IN THE SCIENCES OF COMPLEXITY-PROCEEDINGS VOLUME-* (Vol. 17, pp. 417-417). ADDISON-WESLEY PUBLISHING CO.

[122] Mirjalili, S. (2016). Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Computing and Applications*, *27*(4), 1053-1073.

[123] Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software*, *95*, 51-67.

[124] Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software*, *69*, 46-61.

[125] Mittal, N., Singh, U., Salgotra, R., & Sohi, B. S. A boolean spider monkey optimization based energy efficient clustering approach for WSNs. *Wireless Networks*, 1-17.

[126] Mohan, C., & Shanker, K. (1994). A controlled random search technique for global optimization using quadratic approximation. *Asia-Pacific Journal of Operational Research*, *11*(1), 93-101.

[127] Monteiro Filho, B., Albuquerque, I. M. C., & Neto, F. L. (2017). Fish School Search Algorithm for Constrained Optimization. *arXiv preprint arXiv:1707.06169*.

[128] Muñoz Zavala, A. E., Aguirre, A. H., & Villa Diharce, E. R. (2005, June). Constrained optimization via particle evolutionary swarm optimization algorithm (PESO). In *Proceedings of the 7th annual conference on Genetic and evolutionary computation* (pp. 209-216). ACM.

[129] Naidu, Y. R., & Ojha, A. K. (2015). A hybrid version of invasive weed optimization with quadratic approximation. *Soft Computing*, *19*(12), 3581-3598.

[130] Naidu, Y. R., & Ojha, A. K. (2016). Solving Multiobjective Optimization Problems Using Hybrid Cooperative Invasive Weed Optimization With Multiple Populations. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*.

[131] Niu, B., Fan, Y., Xiao, H., & Xue, B. (2012). Bacterial foraging based approaches to portfolio optimization with liquidity risk. *Neurocomputing*, *98*, 90-100.

[132] Ojha, A. K., & Ota, R. R. (2015). A hybrid method for solving multi–objective geometric programming problem. *International Journal of Mathematics in Operational Research*, *7*(2), 119-137.

[133] Omran, M. G., Engelbrecht, A. P., & Salman, A. (2009). Bare bones differential evolution. European Journal of Operational Research, 196(1), 128-139.

[134] Paik, K. R., Jeong, J. H., & Kim, J. H. (2001). Use of a harmony search for optimal design of coffer dam drainage pipes. *Journal of Korean Society of Civil Engineers*, *21*(2-B), 119-128.

[135] Pal, S. S., Kumar, S., Kashyap, M., Choudhary, Y., & Bhattacharya, M. (2016). Multi-level thresholding segmentation approach based on spider monkey optimization algorithm. In *Proceedings of the Second International Conference on Computer and Communication Technologies* (pp. 273-287). Springer India.

[136] Pant, M., Ali, M., & Singh, V. P. (2009, March). Differential evolution using quadratic interpolation for initializing the population. In *Advance Computing Conference, 2009. IACC 2009. IEEE International* (pp. 375-380). IEEE.

[137] Pant, M., Radha, T., & Singh, V. P. (2007, December). A new particle swarm optimization with quadratic interpolation. In *Conference on Computational Intelligence and Multimedia Applications, 2007. International Conference on* (Vol. 1, pp. 55-60). IEEE.

[138] Parouha, R. P., & Das, K. N. (2015). An efficient hybrid technique for numerical optimization and applications. *Computers & Industrial Engineering*, *83*, 193-216.

[139] Passino, K. M. (2002). Biomimicry of bacterial foraging for distributed optimization and control. *IEEE control systems*, *22*(3), 52-67.

[140] Pulido, G. T., & Coello, C. A. C. (2004, June). A constraint-handling mechanism for particle swarm optimization. In *Evolutionary Computation, 2004. CEC2004. Congress on* (Vol. 2, pp. 1396-1403). Ieee.

[141] Purohit, A., Chaudhari, N. S., & Tiwari, A. (2010, July). Construction of classifier with feature selection based on genetic programming. In *Evolutionary Computation (CEC), 2010 IEEE Congress on* (pp. 1-5). IEEE.

[142] Qin, A. K., & Suganthan, P. N. (2005, September). Self-adaptive differential evolution algorithm for numerical optimization. In *Evolutionary Computation, 2005. The 2005 IEEE Congress on* (Vol. 2, pp. 1785-1791). IEEE.

[143] Rakshit, P., Konar, A., & Nagar, A. K. (2014, July). Artificial bee colony induced multi-objective optimization in presence of noise. In *Evolutionary Computation (CEC), 2014 IEEE Congress on* (pp. 3176-3183). IEEE.

[144] Rana, S., Jasola, S., & Kumar, R. (2011). A review on particle swarm optimization algorithms and their applications to data clustering. *Artificial Intelligence Review*, *35*(3), 211-222.

[145] Rao, R. V., Savsani, V. J., & Vakharia, D. P. (2012). Teaching–learning-based optimization: an optimization method for continuous non-linear large scale problems. *Information sciences*, *183*(1), 1-15.

[146] Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2009). GSA: a gravitational search algorithm. *Information sciences*, *179*(13), 2232-2248.

[147] Sabat, S. L., Ali, L., & Udgata, S. K. (2011). Integrated learning particle swarm optimizer for global optimization. *Applied Soft Computing*, *11*(1), 574-584.

[148] Sabat, S. L., Udgata, S. K., & Abraham, A. (2010). Artificial bee colony algorithm for small signal model parameter extraction of MESFET. *Engineering Applications of Artificial Intelligence*, *23*(5), 689-694.

[149] Sadollah, A., Eskandar, H., Bahreininejad, A., & Kim, J. H. (2015). Water cycle algorithm with evaporation rate for solving constrained and unconstrained optimization problems. *Applied Soft Computing*, *30*, 58-71.

[150] Salhi, S., & Petch, R. J. (2007). A GA based heuristic for the vehicle routing problem with multiple trips. *Journal of Mathematical Modelling and Algorithms*, *6*(4), 591-613.

[151] Salhi, S., Smithies, R., & Queen, N. M. (2005). Predicting Colorectal Cancer Recurrence: A Metaheuristic Approach.

[152] Selvam, K., & Kumar, D. V. (2017). Frequency Control of Micro Grid with wind Perturbations Using Levy walks with Spider Monkey Optimization Algorithm. *International Journal of Renewable Energy Research (IJRER)*, *7*(1), 146-156.

[153] Sharma, A., Sharma, A., Panigrahi, B. K., Kiran, D., & Kumar, R. (2016a). Ageist spider monkey optimization algorithm. *Swarm and Evolutionary Computation*, *28*, 58-77.

[154] Sharma, A., Sharma, H., Bhargava, A., & Sharma, N. (2017). Power law-based local search in spider monkey optimisation for lower order system modelling. *International Journal of Systems Science*, *48*(1), 150-160.

[155] Sharma, A., Sharma, H., Bhargava, A., Sharma, N., & Bansal, J. C. (2016b). Optimal placement and sizing of capacitor using Limaçon inspired spider monkey optimization algorithm. *Memetic Computing*, 1-21.

[156] Sharma, H., Bansal, J. C., Arya, K. V., & Yang, X. S. (2016c). Lévy flight artificial bee colony algorithm. *International Journal of Systems Science*, *47*(11), 2652-2670.

[157] Sharma, V., Dahiya, K., & Verma, V. (2010). Capacitated two-stage time minimization transportation problem. *Asia-Pacific Journal of Operational Research*, *27*(04), 457-476.

[158] Sharma, V., Malhotra, R., & Verma, V. (2013). A cost and pipeline trade-off in a transportation problem. *Yugoslav Journal of Operations Research*, *23*(2), 197-211.

[159] Siarry, P., & Berthiau, G. (1997). Fitting of tabu search to optimize functions of continuous variables. *International journal for numerical methods in engineering*, *40*(13), 2449-2457.

[160] Simon, D. (2008). Biogeography-based optimization. *IEEE transactions on evolutionary computation*, *12*(6), 702-713.

[161] Singh, U., & Kamal, T. S. (2012). Synthesis of thinned planar concentric circular antenna arrays using biogeography-based optimisation. *IET microwaves, antennas & propagation*, *6*(7), 822-829.

[162] Singh, U., & Rattan, M. (2014). Design of thinned concentric circular antenna arrays using firefly algorithm. *IET Microwaves, Antennas & Propagation*, *8*(12), 894-900.

[163] Singh, U., & Salgotra, R. (2016). Optimal synthesis of linear antenna arrays using modified spider monkey optimization. *Arabian Journal for Science and Engineering*, *41*(8), 2957-2973.

[164] Singh, U., Salgotra, R., & Rattan, M. (2016). A Novel Binary Spider Monkey Optimization Algorithm for Thinning of Concentric Circular Antenna Arrays. *IETE Journal of Research*, *62*(6), 736-744.

[165] Sivalingam, R., & Chinnamuthu, S. A Hybrid Self-Adaptive Spider Monkey Optimization for Automatic Generation Control.

[166] Sivasubramani, S., & Swarup, K. S. (2011). Environmental/economic dispatch using multi-objective harmony search algorithm. *Electric power systems research*, *81*(9), 1778-1785.

[167] Spider monkey Facts. (n.d.). Retrieved from http://www.softschools.com/facts/animals/spider_monkey_facts/311/

[168] Storn, R., & Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, *11*(4), 341-359.

[169] Strumberger, I., Bacanin, N., & Tuba, M. (2017, June). Enhanced firefly algorithm for constrained numerical optimization. In *Evolutionary Computation (CEC), 2017 IEEE Congress on* (pp. 2120-2127). IEEE.

[170] Sun, C. L., Zeng, J. C., & Pan, J. S. (2009, August). A particle swarm optimization with feasibility-based rules for mixed-variable optimization problems. In *Hybrid Intelligent Systems, 2009. HIS'09. Ninth International Conference on* (Vol. 1, pp. 543-547). IEEE.

[171] Sun, C. L., Zeng, J. C., & Pan, J. S. (2009, December). An improved particle swarm optimization with feasibility-based rules for mixed-variable optimization problems. In *Innovative Computing, Information and Control (ICICIC), 2009 Fourth International Conference on* (pp. 897-903). IEEE.

[172] Symington, M. M. (1990). Fission-fusion social organization inAteles and Pan. *International Journal of Primatology*, *11*(1), 47-61.

[173] Van den Bergh, F., & Engelbrecht, A. P. (2004). A cooperative approach to particle swarm optimization. IEEE transactions on evolutionary computation, 8(3), 225-239.

[174] Van der Merwe, D. W., & Engelbrecht, A. P. (2003, December). Data clustering using particle swarm optimization. In Evolutionary Computation, 2003. CEC'03. The 2003 Congress on (Vol. 1, pp. 215-220). IEEE.

[175] Verma, V., & Puri, M. C. (1996). A branch and bound procedure for cost minimizing bulk transportation problem. *OPSEARCH-NEW DELHI-*, *33*, 145-161.

[176] Wade, A., & Salhi, S. (2003). An ant system algorithm for the mixed vehicle routing problem with backhauls. In *Metaheuristics: computer decision-making* (pp. 699-719). Springer US.

[177] Wang, Z., Liu, S., & Kong, X. (2012). Artificial bee colony algorithm for portfolio optimization problems. *International Journal of Advancements in Computing Technology*, *4*(4), 8-16.

[178] Wille, L. T., & Vennik, J. (1985). Computational complexity of the ground-state determination of atomic clusters. *Journal of Physics A: Mathematical and General*, *18*(8), L419.

[179] Wolpert, D. H., & Macready, W. G. (1995). *No free lunch theorems for search* (Vol. 10). Technical Report SFI-TR-95-02-010, Santa Fe Institute.

[180] Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, *1*(1), 67-82.

[181] Wu, G., Qiu, D., Yu, Y., Pedrycz, W., Ma, M., & Li, H. (2014). Superior solution guided particle swarm optimization combined with local search techniques. *Expert Systems with Applications*, *41*(16), 7536-7548.

[182] Wu, Q. H., Cao, Y. J., & Wen, J. Y. (1998). Optimal reactive power dispatch using an adaptive genetic algorithm. *International Journal of Electrical Power & Energy Systems*, *20*(8), 563-569.

[183] Yang, X. S. (2011). Review of meta-heuristics and generalised evolutionary walk algorithm. *International Journal of Bio-Inspired Computation*, *3*(2), 77-84.

[184] Yazdani, M., & Jolai, F. (2016). Lion Optimization Algorithm (LOA): A nature-inspired metaheuristic algorithm. *Journal of computational design and engineering*, *3*(1), 24-36.

[185] Yu, X., & Zhang, X. (2014). Enhanced comprehensive learning particle swarm optimization. *Applied Mathematics and Computation*, *242*, 265-276.

[186] Zhang, J., & Sanderson, A. C. (2007, September). JADE: Self-adaptive differential evolution with fast and reliable convergence performance. In *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on* (pp. 2251-2258). IEEE.

[187] Zhao, B., Guo, C. X., & Cao, Y. J. (2005). A multiagent-based particle swarm optimization approach for optimal reactive power dispatch. *IEEE transactions on power systems*, *20*(2), 1070-1078.

[188] Zheng, Y. J. (2015). Water wave optimization: a new nature-inspired metaheuristic. *Computers & Operations Research*, *55*, 1-11.

[189] Zhu, G., & Kwong, S. (2010). Gbest-guided artificial bee colony algorithm for numerical function optimization. *Applied Mathematics and Computation*, *217*(7), 3166-3173.

[190] Zhu, H., Wang, Y., Wang, K., & Chen, Y. (2011). Particle Swarm Optimization (PSO) for the constrained portfolio optimization problem. *Expert Systems with Applications*, *38*(8), 10161-10169.

[191] Zielinski, K., & Laur, R. (2006, July). Constrained single-objective optimization using particle swarm optimization. In *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on* (pp. 443-450). IEEE.

# APPENDICES

# APPENDIX I

# LIST OF CONSTRAINED BENCHMARK PROBLEMS FROM IEEE CEC 2006

**Problem g01**

Minimize $f(x) = 5\sum_{i=1}^{4} x_i - 5\sum_{i=1}^{4} x_i^2 - \sum_{i=5}^{13} x_i$

$$\begin{aligned}
g_1(x) &= (2x_1 + 2x_2 + x_{10} + x_{11}) - 10 \le 0,\\
g_2(x) &= (2x_1 + 2x_3 + x_{10} + x_{12}) - 10 \le 0,\\
g_3(x) &= (2x_2 + 2x_3 + x_{11} + x_{12}) - 10 \le 0,\\
g_4(x) &= -8x_1 + x_{10} \le 0,\\
g_5(x) &= -8x_2 + x_{11} \le 0,\\
g_6(x) &= -8x_3 + x_{12} \le 0,\\
g_7(x) &= -2x_4 - x_5 + x_{10} \le 0,\\
g_8(x) &= -2x_6 - x_7 + x_{11} \le 0,\\
g_9(x) &= -2x_8 - x_9 + x_{12} \le 0,\\
&0 \le x_i \le 1,\ i = 1,\cdots,9,\\
&0 \le x_i \le 100,\ i = 10,11,12,\\
&0 \le x_{13} \le 1.
\end{aligned}$$

This problem has global minima at $x^* = (1,1,\cdots\cdots 1,3,3,3,1)$ with $f_{min} = -15$.

**Problem g02**

Minimize $f(x) = -\left| \frac{\sum_{i=1}^{n} \cos^4(x_i) - 2\prod_{i=1}^{n} \cos^2(x_i)}{\sqrt{\sum_{i=1}^{n} i x_i^2}} \right|$

Subject to:

$$g_1(x) = 0.75 - \prod_{i=1}^{n} x_i \le 0$$

$$g_2(x) = \sum_{i=1}^{n} x_i - 7.5n \le 0$$

where $n = 20$ and $0 \le x_i \le 10$ $(i = 1,\ldots\ldots,n)$. The global minimum $x^* =$

(3.16246061572185, 3.12833142812967, 3.09479212988791, 3.06145059523469, 3.02792915885555, 2.99382606701730, 2.95866871765285, 2.92184227312450, 0.49482511456933, 0.48835711005490, 0.48231642711865, 0.47664475092742, 0.47129550835493, 0.46623099264167, 0.46142004984199, 0.45683664767217, 0.45245876903267, 0.44826762241853, 0.44424700958760, 0.44038285956317)

with $f_{min} = 0.80361910412559$.

**Problem g03**

Minimize $f(x) = -(\sqrt{n})^n \prod_{i=1}^{n} x_i$

Subject to: $h_1(x) = \sum_{i=1}^{n} x_i^2 - 1 = 0$

where $n = 10$ and $0 \le x_i \le 1 \ (i = 1, 2, . . . , n)$, for each variable.

The global minimum is at $x* =$

(0.31624357647283069, 0.316243577414338339, 0.316243578012345927, 0.316243575664017895, 0.316243578205526066, 0.31624357738855069, 0.316243575472949512, 0.316243577164883938, 0.316243578155920302, 0.316243576147374916) with $f_{min} = -1.00050010001$.

**Problem g04**

Minimize $f(x) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141$

Subjectto:

$$
\begin{aligned}
g_1(x) &= 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 - 0.0022053x_3x_5 - 92 \le 0,\\
g_2(x) &= -85.334407 - 0.0056858x_2x_5 - 0.0006262x_1x_4 + 0.0022053x_3x_5 \le 0,\\
g_3(x) &= 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x^2{}_3 - 110 \le 0,\\
g_4(x) &= -80.51249 - 0.0071317x_2x_5 - 0.0029955x_1x_2 - 0.0021813x^2{}_3 + 90 \le 0,\\
g_5(x) &= 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4 - 25 \le 0,\\
g_6(x) &= -9.300961 - 0.0047026x_3x_5 - 0.0012547x_1x_3 - 0.0019085x_3x_4 + 20 \le 0,
\end{aligned}
$$

where $78 \le x_1 \le 102,\ 33 \le x_2 \le 45$ and $27 \le x_i \le 45 \ (i = 3, 4, 5)$. The optimum solution is

$x^* = (78, 33, 29.995256025682, 45, 36.775812905788)$ where $f_{min} = -30665.539$.

**Problem g05**

Minimize $f(x) = 3x_1 + 0.000001x_1^3 + 2x_2 + (0.000002/3)x_2^3$

Subject to:

$$
\begin{aligned}
g_1(x) &= -x_4 + x_3 - 0.55 \le 0,\\
g_2(x) &= -x_3 + x_4 - 0.55 \le 0,\\
h_3(x) &= 1000\, sin(-x_3 - 0.25) + 1000\, sin(-x_4 - 0.25) + 894.8 - x_1 = 0,\\
h_4(x) &= 1000\, sin(x_3 - 0.25) + 1000\, sin(x_3 - x_4 - 0.25) + 894.8 - x_2 = 0,\\
h_5(x) &= 1000\, sin(x_4 - 0.25) + 1000\, sin(x_4 - x_3 - 0.25) + 1294.8 = 0,
\end{aligned}
$$

where $0 \le x_1 \le 1200,\ 0 \le x_2 \le 1200, -0.55 \le x_3 \le 0.55$ and $-0.55 \le x_4 \le 0.55$. The best known solution $x^* = (679.9453, 1026.067, 0.1188764, -0.3962336)$ where $f_{min} = 5126.49671$.

**Problem g06**

Minimize $f(x) = (x_1 - 10)^3 + (x_2 - 20)^3$

Subject to:

$$g_1(x) = -(x_1 - 5)^2 - (x_2 - 5)^2 + 100 \le 0,$$
$$g_2(x) = (x_1 - 6)^2 + (x_2 - 5)^2 - 82.81 \le 0,$$

where $13 \le x_1 \le 100$ and $0 \le x_2 \le 100.$ The optimum solution is $x^* = (14.095, 0.84296)$ where $f_{min} = -6961.81388$.

**Problem g07**

Minimize $f(x) = x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2 + 4(x_4 - 5)^2 + (x_5 - 3)^2 + 2(x_6 - 1)^2 + 5x_7^2 + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45$

Subject to:

$$g_1(x) = -105 + 4x_1 + 5x_2 - 3x_7 + 9x_8 \le 0,$$
$$g_2(x) = 10x_1 - 8x_2 - 17x_7 + 2x_8 \le 0,$$
$$g_3(x) = -8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 \le 0,$$
$$g_4(x) = 3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x^2_3 - 7x_4 - 120 \le 0,$$
$$g_5(x) = 5x^2_1 + 8x_2 + (x_3 - 6)^2 - 2x_4 - 40 \le 0,$$
$$g_6(x) = x^2_1 + 2(x_2 - 2)^2 - 2x_1x_2 + 14x_5 - 6x_6 \le 0,$$
$$g_7(x) = 0.5(x_1 - 8)^2 + 2(x_2 - 4)^2 + 3x^2_5 - x_6 - 30 \le 0,$$
$$g_8(x) = -3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10} \le 0,$$

where $-10 \le x_i \le 10 \ (i = 1, \ldots, 10)$. The optimum solution is

$x^* = (2.171996, 2.363683, 8.773926, 5.095984, 0.9906548, 1.430574, 1.321644, 9.828726, 8.280092, 8.375927)$ where $f_{min} = 24.3062091$.

**Problem g08**

Minimize $f(x) = -\dfrac{\sin^3(2\pi x_1)\sin(2\pi x_2)}{x_1^3(x_1 + x_2)}$

subject to:

$$g_1(x) = x^2_1 - x_2 + 1 \le 0,$$
$$g_2(x) = 1 - x_1 + (x_2 - 4)^2 \le 0,$$

where $0 \le x_1 \le 10$ and $0 \le x_2 \le 10.$ The optimum is located at

$x^* = (1.22797135260752599, 4.24537336612274885)$ where $f_{min} = -0.0958250414180359.$

**Problem g09**

Minimize $f\left(x\right)=(x_1-10)^2+5(x_2-12)^2+x_3^4+3(x_4-11)^2+10x_5^6+7x_6^2+x_7^4-4x_6x_7-10x_6-8x_7$

Subjectto:

$$
\begin{aligned}
g_1\left(x\right)&=-127+2x_1^2+3x_2^4+x_3+4x_4^2+5x_5\le 0,\\
g_2\left(x\right)&=-282+7x_1+3x_2+10x_3^2+x_4-x_5\le 0,\\
g_3\left(x\right)&=-196+23x_1+x_2^2+6x_6^2-8x_7\le 0,\\
g_4\left(x\right)&=\ 4x_1^2+x_2^2-3x_1x_2+2x_3^2+5x_6-11x_7\le 0,
\end{aligned}
$$

where $-10\le x_i\le 10$ for $\left(i=1,\ ...,7\right)$. The optimum solution is

$x^*=(2.33049935147405174,1.95137236847114592,-0.477541399510615805,$
$4.36572624923625874,-0.624486959100388983,1.03813099410962173,\ 1.5942266780671519)$
with $f_{min}=\ 680.630057374402$.

**Problem g10**

Minimize $f\left(x\right)\ =x_1+x_2+x_3$

Subject to:

$$
\begin{aligned}
g_1\left(x\right)&=-1+0.0025(x_4+x_6)\le 0,\\
g_2\left(x\right)&=-1+0.0025(x_5+x_7-x_4)\le 0,\\
g_3\left(x\right)&=-1+0.01(x_8-\ x_5)\le 0,\\
g_4\left(x\right)&=-x_1x_6+833.33252x_4+100x_1-83333.333\le 0,\\
g_5\left(x\right)&=-x_2x_7+1250x_5+x_2x_4-1250x_4\le 0,\\
g_6\left(x\right)&=-x_3x_8+1250000\ x_3x_5-2500x_5\le 0,
\end{aligned}
$$

where $100\le x_1\le 10000,\ 1000\le x_i\le 10000\ \left(i=\ 2,3\right)$ and $10\le x_i\le 1000\ (i=4,\ ...,8)$. The optimum solution is at

$x^*=$ (579.306685017979589, 1359.97067807935605, 5109.97065743133317,
182.01769963061534, 295.601173702746792, 217.982300369384632, 286.41652592786852,
395.601173702746735) with $f_{min}=\ 7049.24802052867$.

**Problem g11**

Minimize $f\left(x\right)=x_1^2+(x_2-1)^2$

Subject to:

$h_1\left(x\right)=x_2-\ x_1^2=0,$

The bounds on the variables are: $-1\le x_i\le 1,\ i=\ 1,2$.This problem has global minima at

$x^*=\left(-0.707036070037170616,0.500000004333606807\right)$ with $f_{min}=\ 0.7499$.

**Problem g12**

Minimize $f(x) = -(100-(x_1-5)^2-(x_2-5)^2-(x_3-5)^2)/100$

Subjectto:

$$g(x) = (x_1 - p)^2 + (x_2 - q)^2 + (x_3 - r)^2 - 0.0625 \le 0,$$

where $0 \le x_i \le 10 (i = 1,2,3)$ and $p,q,r = 1,2,...,9$. The feasible region of the search space consists of $9^3$ disjoint spheres. A point $(x_1, x_2, x_3)$ is feasible if and only if there exist $p,q,r$ such that the above inequality holds. The optimum is located at $x^* = (5,5,5)$ where $f_{min} = -1$. The solution lies within the feasible region.

**Problem g13**

Minimize $f(x) = e^{(x_1x_2x_3x_4x_5)}$

Subjectto:

$$h_1(x) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 = 0,$$
$$h_2(x) = x_2x_3 - 5x_4x_5 = 0,$$
$$h_3(x) = x_1^3 + x_2^3 + 1 = 0,$$

where $-2.3 \le x_i \le 2.3$ $(i = 1,2)$ and $-3.2 \le x_i \le 3.2$ $(i = 3,4,5)$. The optimum solution is

$x^* = (-1.71714224003, 1.59572124049468, 1.8272502406271,$

$-0.763659881912867, -0.76365986736498)$ with $f_{min} = 0.053941514041898.$

**Problem g14**

Minimize $f(x) = \sum_{i=1}^{10} x_i \left( c_i + \ln \frac{x_i}{\sum_{j=1}^{10} x_j} \right)$

Subjectto:

$$h_1(x) = x_1 + 2x_2 + 2x_3 + x_6 + x_{10} - 2 = 0,$$
$$h_2(x) = x_4 + 2x_5 + x_6 + x_7 - 1 = 0,$$
$$h_3(x) = x_3 + x_7 + x_8 + 2x_9 + x_{10} - 1 = 0,$$

where the bounds are $0 \le x_i \le 10$ $(i = 1, ..., 10)$ and $c_1 = -6.089, c_2 = -17.164,$

$c_3 = -34.054, c_4 = -5.914, c_5 = -24.721, c_6 = -14.986, c_7 = -24.1, c_8 = -10.708, c_9 = -26.662,$
$c_{10} = -22.179$. The best known solution is at $x^* = (0.0406684113216282,$

$0.147721240492452, 0.783205732104114, 0.00141433931889084, 0.485293636780388,$
$0.000693183051556082, 0.0274052040687766, 0.0179509660214818, 0.0373268186859717,$
$0.0968844604336845)$ with $f_{min} = -47.7648884594915.$

**Problem g15**

Minimize $f(x) = 1000 - x_1^2 - 2x_2^2 - x_3^2 - x_1x_2 - x_1x_3$

Subjectto:

$$h_1(x) = x_1^2 + x_2^2 + x_3^2 - 25 = 0,$$
$$h_2(x) = 8x_1 + 14x_2 + 7x_3 - 56 = 0,$$

where the bounds are $0 \le x_i \le 10$ $(i = 1, 2, 3)$. The best known solution is at

$$x^* = (3.51212812611795133,\ 0.216987510429556135,\ 3.55217854929179921)$$

with $f_{min} = 961.715022289961.$

**Problem g16**

Minimize

$$f(x) = -0.0000005843 y_{17} + 0.000117 y_{14} + 0.1365 + 0.00002358 y_{13} + 0.000001502 y_{16}$$
$$+ 0.0321 y_{12} + 0.004324 y_5 + \frac{c_{15}}{c_{16}} + 37.48 \frac{y_2}{c_{12}}$$

Subject to:

$$g_1(x) = \frac{0.28}{0.72} y_5 - y_4 \le 0,$$
$$g_2(x) = x_3 - 1.5x_2 \le 0,$$
$$g_3(x) = 3496 \frac{y_2}{c_{12}} - 21 \le 0$$
$$g_4(x) = 110.6 + y_1 - \frac{62212}{c_{17}} \le 0$$
$$g_5(x) = y_1 - 213.1 \ge 0,$$
$$g_6(x) = 405.23 - y_1 \ge 0,$$
$$g_7(x) = y_2 - 17.505 \ge 0,$$
$$g_8(x) = 1053.6667 - y_2 \ge 0,$$
$$g_9(x) = y_3 - 11.275 \ge 0,$$
$$g_{10}(x) = 35.03 - y_3 \ge 0,$$
$$g_{11}(x) = y_4 - 214.228 \ge 0,$$
$$g_{12}(x) = 665.585 - y_4 \ge 0,$$
$$g_{13}(x) = y_5 - 7.458 \ge 0,$$
$$g_{14}(x) = 584.463 - y_5 \ge 0,$$
$$g_{15}(x) = y_6 - 0.961 \ge 0,$$
$$g_{16}(x) = 265.916 - y_6 \ge 0,$$
$$g_{17}(x) = y_7 - 1.612 \ge 0,$$
$$g_{18}(x) = 7.046 - y_7 \ge 0,$$

$$g_{19}(x) = y_8 - 0.146 \geq 0,$$
$$g_{20}(x) = 0.222 - y_8 \geq 0,$$
$$g_{21}(x) = y_9 - 107.99 \geq 0,$$
$$g_{22}(x) = 273.366 - y_9 \geq 0,$$
$$g_{23}(x) = y_{10} - 922.693 \geq 0,$$
$$g_{24}(x) = 1286.105 - y_{10} \geq 0,$$
$$g_{25}(x) = y_{11} - 926.832 \geq 0,$$
$$g_{26}(x) = 1444.046 - y_{11} \geq 0,$$
$$g_{27}(x) = y_{12} - 18.766 \geq 0,$$
$$g_{28}(x) = 537.141 - y_{12} \geq 0,$$
$$g_{29}(x) = y_{13} - 1072.163 \geq 0,$$
$$g_{30}(x) = 3247.039 - y_{13} \geq 0,$$
$$g_{31}(x) = y_{14} - 8961.448 \geq 0,$$
$$g_{32}(x) = 26844.086 - y_{14} \geq 0,$$
$$g_{33}(x) = y_{15} - 0.063 \geq 0,$$
$$g_{34}(x) = 0.386 - y_{15} \geq 0,$$
$$g_{35}(x) = y_{16} - 71084.33 \geq 0,$$
$$g_{36}(x) = 140000 - y_{16} \geq 0,$$
$$g_{37}(x) = y_{17} - 2802713 \geq 0,$$
$$g_{38}(x) = 12146108 - y_{17} \geq 0,$$

where:

$$y_1 = x_2 + x_3 + 41.6,$$
$$c_1 = 0.024x_4 - 4.62,$$
$$y_2 = \frac{12.5}{c_1} + 12,$$
$$c_2 = 0.0003535x^2{}_1 + 0.5311x_1 + 0.08705y_2x_1,$$
$$c_3 = 0.052x_1 + 78 + 0.002377y_2x_1,$$
$$y_3 = \frac{c_2}{c_3},$$
$$y_4 = 19y_3,$$
$$c_4 = 0.04782(x_1 - y_3) + \frac{0.1956(x_1 - y_3)^2}{x_2} + 0.6376y_4 + 1.594y_3,$$
$$c_5 = 100x_2,$$
$$c_6 = x_1 - y_3 - y_4,$$
$$c_7 = 0.950 - \frac{c_4}{c_5},$$
$$y_5 = c_6c_7,$$
$$y_6 = x_1 - y_5 - y_4 - y_3,$$
$$c_8 = (y_5 + y_4)0.995,$$
$$y_7 = \frac{c_8}{y_1},$$
$$y_8 = \frac{c_8}{3798},$$

$c_9 = y_7 - \frac{0.0663 y_7}{y_8} - 0.3153,$
$y_9 = \frac{96.82}{c_9} + 0.321 y_1,$
$y_{10} = 1.29 y_5 + 1.258 y_4 + 2.29 y_3 + 1.71 y_6,$
$y_{11} = 1.71 x_1 - 0.452 y_4 + 0.580 y_3,$
$c_{10} = \frac{12.3}{752.3}$
$c_{11} = (1.75 y_2)(0.995 x_1),$
$c_{12} = 0.995 y_{10} + 1998,$
$y_{12} = c_{10} x_1 + \frac{c_{11}}{c_{12}},$
$y_{13} = c_{12} - 1.75 y_2,$
$y_{14} = 3623 + 64.4 x_2 + 58.4 x_3 + \frac{143612}{y_9 + x_5},$
$c_{13} = 0.995 y_{10} + 60.8 x_2 + 48 x_4 - 0.1121 y_{14} - 5095,$
$y_{15} = \frac{y_{13}}{c_{13}},$
$y_{16} = 148000 - 331000 y_{15} + 40 y_{13} - 61 y_{15} y_{13},$
$c_{14} = 2324 y_{10} - 28740000 y_2,$
$y_{17} = 14130000 - 1328 y_{10} - 531 y_{11} + \frac{c_{14}}{c_{12}},$
$c_{15} = \frac{y_{13}}{y_{15}} - \frac{y_{13}}{0.52},$
$c_{16} = 1.104 - 0.72 y_{15},$
$c_{17} = y_9 + x_5,$

and where the bounds are $704.4148 \le x_1 \le 906.3855,\ 68.6 \le x_2 \le 288.88,$

$0 \le x_3 \le 134.75, 193 \le x_4 \le 287.0966$ and $25 \le x_5 \le 84.1988$. The best known solution is at

$x^* = (705.174537070090537,\ 68.5999999999999943,\ 102.899999999999991,$

$282.324931593660324,\ 37.5841164258054832)$ with $f_{min} = -1.90515525853479.$

**Problem g17**

Minimize $f(x) = f(x_1) + f(x_2)$

where

$$f_1(x_1) = \begin{cases} 30x_1 & 0 \le x_1 < 300 \\ 31x_1 & 300 \le x_1 < 400 \end{cases}$$

$$f_2(x_2) = \begin{cases} 28x_2 & 0 \le x_2 < 100 \\ 29x_2 & 100 \le x_2 < 200 \\ 30x_2 & 200 \le x_2 < 1000 \end{cases}$$

Subject to:

$$h_1(x) = -x_1 + 300 - \frac{x_3x_4}{131.078}\cos(1.48477 - x_6) + \frac{0.90798x_3^2}{131.078}\cos(1.47588),$$
$$h_2(x) = -x_2 - \frac{x_3x_4}{131.078}\cos(1.48477 + x_6) + \frac{0.90798x_4^2}{131.078}\cos(1.47588),$$
$$h_3(x) = -x_5 - \frac{x_3x_4}{131.078}\sin(1.48477 + x_6) + \frac{0.90798x_4^2}{131.078}\sin(1.47588),$$
$$h_4(x) = 200 - \frac{x_3x_4}{131.078}\sin(1.48477 - x_6) + \frac{0.90798x_3^2}{131.078}\sin(1.47588),$$

where the bounds are $0 \le x_1 \le 400,\ 0 \le x_2 \le 1000,\ 340 \le x_3 \le 420,\ 340 \le x_4 \le 420,$

$-1000 \le\ x_5 \le 1000$ and $0 \le\ x_6 \le 0.5236.$ The best known solution isat

$x^* = (201.784467214523659,\ 99.\ 9999999999999005,\ 383.071034852773266, 420,$

$-10.9076584514292652,\ 0.0731482312084287128)$ where $f\left(x\right) = 8853.53967480648$

**Problem g18**

Minimize $f\left(x\right)\ = -0.5(x_1x_4 - x_2x_3 + x_3x_9 - x_5x_9 + x_5x_8 - x_6x_7)$

Subject to:

$$\begin{aligned}
g_1\left(x\right) &= x^2{}_3 + x^2{}_4 - 1 \le 0,\\
g_2\left(x\right) &= x^2{}_9 - 1 \le 0,\\
g_3\left(x\right) &= x^2{}_5 + x^2{}_6 - 1 \le 0,\\
g_4\left(x\right) &= x^2{}_1 +\ (x_2 -\ x_9)^2 - 1 \le 0,\\
g_5\left(x\right) &=\ (x_1 -\ x_5)^2 +\ (x_2 -\ x_6)^2 - 1 \le 0,\\
g_6\left(x\right) &=\ (x_1 -\ x_7)^2 +\ (x_2 -\ x_8)^2 - 1 \le 0,\\
g_7\left(x\right) &=\ (x_3 -\ x_5)^2 +\ (x_4 -\ x_6)^2 - 1 \le 0,\\
g_8\left(x\right) &=\ (x_3 -\ x_7)^2 +\ (x_4 -\ x_8)^2 - 1 \le 0,\\
g_9\left(x\right) &= x^2{}_7 +\ (x_8 -\ x_9)^2 - 1 \le 0,\\
g_{10}\left(x\right) &= x_2x_3 -\ x_1x_4 \le 0,\\
g_{11}\left(x\right) &= -x_3x_9 \le 0,\\
g_{12}\left(x\right) &= x_5x_9 \le 0,\\
g_{13}\left(x\right) &= x_6x_7 -\ x_5x_8 \le 0,
\end{aligned}$$

where the bounds are $-10 \le x_i \le 10\left(i = 1,...,8\right)$ and $0 \le x_9 \le 20.$ The best known -solution is

at $x^* = (-0.657776192427943163, -0.153418773482438542,\ 0.323413871675240938,$

$-0.946257611651304398, -0.657776194376798906, -0.753213434632691414,$

$0.323413874123576972, -0.346462947962331735, 0.59979466285217542)$

where $f_{min} = -0.866025403784439.$

**Problem g19**

Minimize $f(x) = \sum_{j=1}^{5}\sum_{i=1}^{5} c_{ij}x_{(10+i)}x_{(10+j)} + 2\sum_{j=1}^{5} d_j x^3_{(10+j)} - \sum_{i=1}^{10} b_i x_i$

Subject to:

$$g_j(x) = -2\sum_{i=1}^{5} c_{ij} x_{(10+i)} - 3d_j x^2_{(10+j)} - e_j + \sum_{i=1}^{10} a_{ij} x_i \le 0, \quad j = 1,...,5.$$

where $b = [-40, -2, -0.25, -4, -4, -1, -40, -60, 5, 1]$ and the remaining data is on Table B.1. The bounds are $0 \le x_i \le 10 \ (i = 1, \ . . . ,15)$. The best known solution is at

$x* = (1.66991341326291344e-17,\ 3.95378229282456509e-16,\ 3.94599045143233784,$ $1.06036597479721211e-16, 3.2831773458454161, 9.99999999999999822, 1.1282941467$ $1605333e-17, 1.2026194599794709e - 17,\ 2.50706276000769697e\ -15,\ 2.2462412298$ $7970677e-15,\ 0.370764847417013987, 0.278456024942955571, 0.523838487672241171,$ $0.388620152510322781,\ 0.298156764974678579)$

with $f_{min}$ = 32.6555929502463.

Table B.1: Data set for test problem g19

| *j* | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $e_j$ | -15 | -27 | -36 | -18 | -12 |
| $c_{1j}$ | 30 | -20 | -10 | 32 | -10 |
| $c_{2j}$ | -20 | 39 | -6 | -31 | 32 |
| $c_{3j}$ | -10 | -6 | 10 | -6 | -10 |
| $c_{4j}$ | 32 | -31 | -6 | 39 | -20 |
| $c_{5j}$ | -10 | 32 | -10 | -20 | 30 |
| $d_j$ | 4 | 8 | 10 | 6 | 2 |
| $a_{1j}$ | -16 | 2 | 0 | 1 | 0 |
| $a_{2j}$ | 0 | -2 | 0 | 0.4 | 2 |
| $a_{3j}$ | -3.5 | 0 | 2 | 0 | 0 |
| $a_{4j}$ | 0 | -2 | 0 | -4 | -1 |
| $a_{5j}$ | 0 | -9 | -2 | 1 | -2.8 |
| $a_{6j}$ | 2 | 0 | -4 | 0 | 0 |
| $a_{7j}$ | -1 | -1 | -1 | -1 | -1 |
| $a_{8j}$ | -1 | -2 | -3 | -2 | -1 |
| $a_{9j}$ | 1 | 2 | 3 | 4 | 5 |
| $a_{10j}$ | 1 | 1 | 1 | 1 | 1 |

**Problem g20**

Minimize $f(x) = \sum_{i=1}^{24} a_i x_i$

Subject to:

$$g_i(x) = \frac{(x_i + x_{(i+12)})}{\sum_{j=1}^{24} x_j + e_i} \le 0, \qquad i = 1, 2, 3$$
$$g_i(x) = \frac{(x_{(i+3)} + x_{(i+15)})}{\sum_{j=1}^{24} x_j + e_i} \le 0, \qquad i = 4, 5, 6$$
$$h_i(x) = \frac{x_{(i+12)}}{b_{(i+12)} \sum_{j=13}^{24} \frac{x_j}{b_j}} - \frac{c_i x_i}{40 b_i \sum_{j=1}^{12} \frac{x_j}{b_j}} = 0, \qquad i = 1, \ldots, 12,$$
$$h_{13}(x) = \sum_{i=1}^{24} x_i - 1 = 0,$$
$$h_{14}(x) = \sum_{i=1}^{12} \frac{x_i}{d_i} + k \sum_{i=13}^{24} \frac{x_i}{b_i} - 1.671 = 0,$$

where k= (0.7302)(530)$\left(\frac{14.7}{40}\right)$and the data set is detailed on Table III.2. The bounds are $0 \le x_i \le 10 \ (i = 1, \ldots, 24)$. The best known solution is at

$x^*$ = (1.28582343498528086 e-18, 4.83460302526130664 e-34, 0, 0, 6.30459929660781851 e-18, 7.57192526201145068 e-34, 5.03350698372840437 e-34, 9.28268079616618064 e-34, 0, 1.76723384525547359 e-17, 3.55686101822965701 e-34, 2.99413850083471346 e-34, 0.158143376337580827, 2.29601774161699833 e-19, 1.06106938611042947 e-18, 1.3196834431950 e-18, 0.530902525044209539, 0, 2.89148310257773535e - 18, 3.34892126180666159 e-18, 0, 0.310999974151577319, 5.41244666317833561 e-05, 4.84993165246959553 e-16).

This solution is a little infeasible and no feasible solution is found so far.

**Problem g21**

Minimize $f(x) = x_1$

Subject to:

$$g_1(x) = -x_1 + 35x_2^{0.6} + 35x_3^{0.6} \le 0,$$
$$h_1(x) = -300x_3 + 7500x_5 - 7500x_6 - 25x_4x_5 + 25x_4x_6 + x_3x_4 = 0,$$
$$h_2(x) = 100x_2 + 155.365x_4 + 2500x_7 - x_2x_4 - 25x_4x_7 - 15536.5 = 0,$$
$$h_3(x) = -x_5 + ln(-x_4 + 900) = 0,$$
$$h_4(x) = -x_6 + ln(x_4 + 300) = 0,$$
$$h_5(x) = -x_7 + ln(-2x_4 + 700) = 0,$$

where the bounds are $0 \le x_1 \le 1000,\ 0 \le x_2, x_3 \le 40,\ 100 \le x_4 \le 300,\ 6.3 \le x_5 \le 6.7,$ $5.9 \le x_6 \le 6.4$ and $4.5 \le x_7 \le 6.25$. The best known solution is at

$x^* =$(193.724510070034967, 5.56944131553368433e-27, 17.3191887294084914,100.047897801386839, 6.68445185362377892, 5.99168428444264833,6.21451648886070451)where $f_{min} = 193.724510070035$.

Table B.2: Data set for test problem g20

| | | | | | |
|---|---|---|---|---|---|
| 1 | 0.0693 | 44.094 | 123.7 | 31.244 | 0.1 |
| 2 | 0.0577 | 58.12 | 31.7 | 36.12 | 0.3 |
| 3 | 0.05 | 58.12 | 45.7 | 34.784 | 0.4 |
| 4 | 0.2 | 137.4 | 14.7 | 92.7 | 0.3 |
| 5 | 0.26 | 120.9 | 84.7 | 82.7 | 0.6 |
| 6 | 0.55 | 170.9 | 27.7 | 91.6 | 0.3 |
| 7 | 0.06 | 62.501 | 49.7 | 56.708 | |
| 8 | 0.1 | 84.94 | 7.1 | 82.7 | |
| 9 | 0.12 | 133.425 | 2.1 | 80.8 | |
| 10 | 0.18 | 82.507 | 17.7 | 64.517 | |
| 11 | 0.1 | 46.07 | 0.85 | 49.4 | |
| 12 | 0.09 | 60.097 | 0.64 | 49.1 | |
| 13 | 0.0693 | 44.094 | | | |
| 14 | 0.0577 | 58.12 | | | |
| 15 | 0.05 | 58.12 | | | |
| 16 | 0.2 | 137.4 | | | |
| 17 | 0.26 | 120.9 | | | |
| 18 | 0.55 | 170.9 | | | |
| 19 | 0.06 | 62.501 | | | |
| 20 | 0.1 | 84.94 | | | |
| 21 | 0.12 | 133.425 | | | |
| 22 | 0.18 | 82.507 | | | |
| 23 | 0.1 | 46.07 | | | |
| 24 | 0.09 | 60.097 | | | |

**Problem g22**

Minimize $f(x) = x_1$

Subject to:

$$
\begin{aligned}
g_1(x) &= -x_1 + x_2^{0.6} + x_3^{0.6} + x_4^{0.6} \le 0,\\
h_1(x) &= x_5 - 100000x_8 + 1\times10^7 = 0,\\
h_2(x) &= x_6 + 100000x_8 - 100000x_9 = 0,\\
h_3(x) &= x_7 + 100000x_9 - 5\times10^7 = 0,\\
h_4(x) &= x_5 + 100000x_{10} - 3.3\times10^7 = 0,\\
h_5(x) &= x_6 + 100000x_{11} - 4.4\times10^7 = 0,\\
h_6(x) &= x_7 + 100000x_{12} - 6.6\times10^7 = 0,\\
h_7(x) &= x_5 - 120x_2x_{13} = 0,\\
h_8(x) &= x_6 - 80x_3x_{14} = 0,\\
h_9(x) &= x_7 - 40x_4x_{15} = 0,\\
h_{10}(x) &= x_8 - x_{11} + x_{16} = 0,\\
h_{11}(x) &= x_9 - x_{12} + x_{17} = 0,\\
h_{12}(x) &= -x_{18} + ln(x_{10} - 100) = 0,\\
h_{13}(x) &= -x_{19} + ln(-x_8 + 300) = 0,\\
h_{14}(x) &= -x_{20} + ln(x_{16}) = 0,\\
h_{15}(x) &= -x_{21} + ln(-x_9 + 400) = 0,\\
h_{16}(x) &= -x_{22} + ln(x_{17}) = 0,\\
h_{17}(x) &= -x_8 - x_{10} + x_{13}x_{18} - x_{13}x_{19} + 400 = 0,\\
h_{18}(x) &= x_8 - x_9 - x_{11} + x_{14}x_{20} - x_{14}x_{21} + 400 = 0,\\
h_{19}(x) &= x_9 - x_{12} - 4.60517x_{15} + x_{15}x_{22} + 100 = 0,
\end{aligned}
$$

where the bounds are $0 \le x_1 \le 20000,\ 0 \le x_2, x_3, x_4 \le 1\times10^6, 0 \le x_5, x_6, x_7 \le 4\times10^7,$

$100 \le x_8 \le 299.99, 100 \le x_9 \le 399.99, 100.01 \le x_{10} \le 300,\ 100 \le x_{11} \le 400, 100 \le x_{12} \le 600,$

$0 \le x_{13}, x_{14}, x_{15} \le 500,\ 0.01 \le x_{16} \le 300,\ 0.01 \le x_{17} \le 400,\ -4.7 \le x_{18},\ x_{19},\ x_{20},\ x_{21},\ x_{22} \le 6.25.$
The best known solution is at $x^* =$(236.430975504001054, 135.82847151732463, 204.818152544824585, 6446.54654059436416, 3007540.83940215595, 4074188.65771341929, 32918270.5028952882, 130.075408394314167, 170.817294970528621, 299.924591605478554, 399.258113423595205, 330.817294971142758, 184.51831230897065, 248.64670239647424, 127.658546694545862, 269.182627528746707, 160.000016724090955, 5.29788288102680571, 5.13529735903945728, 5.59531526444068827, 5.43444479314453499, 5.07517453535834395) where $f_{min} = 236.430975504001$.

**Problem g23**

Minimize $f(x) = -9x_5 - 15x_8 + 6x_1 + 16x_2 + 10(x_6 + x_7)$

Subject to:

$$\begin{aligned}
g_1(x) &= x_9x_3 + 0.02x_6 - 0.025x_5 \le 0,\\
g_2(x) &= x_9x_4 + 0.02x_7 - 0.015x_8 \le 0,\\
h_1(x) &= x_1 + x_2 - x_3 - x_4 = 0,\\
h_2(x) &= 0.03x_1 + 0.01x_2 - x_9(x_3 + x_4) = 0,\\
h_3(x) &= x_3 + x_6 - x_5 = 0,\\
h_4(x) &= x_4 + x_7 - x_8 = 0,
\end{aligned}$$

where the bounds are $0 \le x_1, x_2, x_6 \le 300$, $0 \le x_3, x_5, x_7 \le 100$, $0 \le x_4, x_8 \le 200$ and $0.01 \le x_9 \le 0.03$. The best known solution is at $x^* =$(0.00510000000000259465, 99.9947000000000514, 9.01920162996045897e - 18, 99.9999000000000535, 0.000100000000027086086, 2.75700683389584542e -14, 99.9999999999999574, 2000.0100000100000100008) where $f_{min} =$-400.055099999999584.

**Problem g24**

Minimize $f(x) = -x_1 - x_2$

Subject to:

$$\begin{aligned}
g_1(x) &= -2x_1^4 + 8x_1^3 - 8x_1^2 + x_2 - 2 \le 0,\\
g_2(x) &= -4x_1^4 + 32x_1^3 - 88x_1^2 + 96x_1 + x_2 - 36 \le 0,
\end{aligned}$$

The bounds on the variables are $0 \le x_1 \le 3$, $0 \le x_2 \le 4$. This problem has one global minima at $x^* =$ (2.32952019747762, 3.17849307411774) with $f_{min} =$-5.50801327159536.

# APPENDIX II

# LIST OF CONSTRAINED BENCHMARK PROBLEMS FROM IEEE CEC 2010

**Problem C01**

$$Min\ f(x) = -\left|\frac{\sum_{i=1}^{D} cos^4(z_i) - 2\prod_{i=1}^{D} cos^2(z_i)}{\sqrt{\sum_{i=1}^{D} iz_i^2}}\right|, \quad z = x - o$$

$$g_1(x) = 0.75 - \prod_{i=1}^{D} z_i \ \leq 0$$

$$g_2(x) = \sum_{i=1}^{D} z_i - 7.5D \ \leq 0$$

$x \in\ [0, 10]^D$

**Problem C02**

$$Min\ f(x) = \max(z), \qquad z = x - o, y = z - 0.5$$

$$g_1(x) = 10 - \frac{1}{D}\sum_{i=1}^{D}[z_i^2 - 10\cos(2\pi z_i) + 10] \ \leq 0$$

$$g_2(x) = \frac{1}{D}\sum_{i=1}^{D}[z_i^2 - 10\cos(2\pi z_i) + 10] - 15 \ \leq 0$$

$$h(x) = \frac{1}{D}\sum_{i=1}^{D}[y_i^2 - 10\cos(2\pi y_i) + 10] - 20 = 0$$

$x \in\ [-5.12, 5.12]^D$

**Problem C03**

$$Min\ f(x) = \sum_{i=1}^{D-1}(100\ (z_i^2 - \ z_{i+1})^2 + (z_i - 1)^2), \ \ z = x - o$$

$$h(x) = \sum_{i=1}^{D-1} (z_i^2 - z_{i+1})^2 = 0$$

$x \in [-1000, 1000]^D$

**Problem C04**

$Min\, f(x) = \max(z)$, $\qquad z = x - o$

$$h_1(x) = \frac{1}{D}\sum_{i=1}^{D}\left(z_i cos\left(\sqrt{|z_i|}\right)\right) = 0$$

$$h_2(x) = \sum_{i=1}^{(D/2)-1} (z_i - z_{i+1})^2 = 0$$

$$h_3(x) = \sum_{i=(D/2)+1}^{D-1} (z_i^2 - z_{i+1})^2 = 0$$

$$h_4(x) = \sum_{i=1}^{D} z = 0$$

$x \in [-50, 50]^D$

**Problem C05**

$Min\, f(x) = \max(z)$, $\qquad z = x - o$

$$h_1(x) = \frac{1}{D}\sum_{i=1}^{D}\left(-z_i sin\left(\sqrt{|z_i|}\right)\right) = 0$$

$$h_2(x) = \frac{1}{D}\sum_{i=1}^{D}\left(-z_i cos\left(0.5\sqrt{|z_i|}\right)\right) = 0$$

$x \in [-600, 600]^D$

**Problem C06**

$Min\, f(x) = \max(z)$

$z = x - o, \qquad y = (x + 483.6106156535 - o)M - 483.6106156535$

$$h_1(x) = \frac{1}{D}\sum_{i=1}^{D}\left(-y_i sin\left(\sqrt{|y_i|}\right)\right) = 0$$

$$h_2(x) = \frac{1}{D}\sum_{i=1}^{D}\left(-y_i cos\left(0.5\sqrt{|y_i|}\right)\right) = 0$$

$x \in [-600, 600]^D$

**Problem C07**

$$Min\, f(x) = \sum_{i=1}^{D-1}(100\,(z_i^2 - z_{i+1})^2 + (z_i - 1)^2)\,, \;\; z = x + 1 - o, y = x - o$$

$$g(x) = 0.5 - exp\left(-0.1\sqrt{\frac{1}{D}\sum_{i=1}^{D} y_i^2}\right) - 3exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(0.1y)\right) + \exp(1) \leq 0$$

$x \in [-140, 140]^D$

**Problem C08**

$$Min\, f(x) = \sum_{i=1}^{D-1}(100\,(z_i^2 - z_{i+1})^2 + (z_i - 1)^2)\,, \;\; z = x + 1 - o, y = (x - o)M$$

$$g(x) = 0.5 - exp\left(-0.1\sqrt{\frac{1}{D}\sum_{i=1}^{D} y_i^2}\right) - 3exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(0.1y)\right) + \exp(1) \leq 0$$

$x \in [-140, 140]^D$

**Problem C09**

$$Min\, f(x) = \sum_{i=1}^{D-1}(100\,(z_i^2 - z_{i+1})^2 + (z_i - 1)^2)\,, \;\; z = x + 1 - o, y = x - o$$

$$h(x) = \sum_{i=1}^{D}\left(y_i sin\left(\sqrt{|y_i|}\right)\right) = 0$$

$x \in [-500, 500]^D$

**Problem C10**

$$Min\, f(x) = \sum_{i=1}^{D-1}(100\,(z_i^2 - z_{i+1})^2 + (z_i - 1)^2)\,, \;\; z = x + 1 - o, y = (x - o)M$$

$$h(x) = \sum_{i=1}^{D} \left( y_i sin\left(\sqrt{|y_i|}\right)\right) = 0$$

$x \in [-500, 500]^D$

**Problem C11**

$$Min\ f(x) = \frac{1}{D}\sum_{i=1}^{D} \left( -z_i cos\left(2\sqrt{|z_i|}\right)\right) \quad z = (x - o)M, y = x + 1 - o$$

$$h(x) = \sum_{i=1}^{D-1} (100\ (y_i^2 - y_{i+1})^2 + (y_i - 1)^2) = 0$$

$x \in [-100, 100]^D$

**Problem C12**

$$Min\ f(x) = \sum_{i=1}^{D} \left( z_i sin\left(\sqrt{|z_i|}\right)\right), \quad z = x - o$$

$$h(x) = \sum_{i=1}^{D-1} (z_i^2 - z_{i+1})^2 = 0$$

$$g(x) = \sum_{i=1}^{D} (z - 100\cos(0.1z) + 10) \le 0$$

$x \in [-1000, 1000]^D$

**Problem C13**

$$Min\ f(x) = \frac{1}{D}\sum_{i=1}^{D} \left( -z_i sin\left(\sqrt{|z_i|}\right)\right), \quad z = x - o$$

$$g_1(x) = -50 + \frac{1}{100D}\sum_{i=1}^{D} z_i^2 \le 0$$

$$g_2(x) = \frac{50}{D}\sum_{i=1}^{D} sin\left(\frac{1}{50}\pi z\right) \le 0$$

$$g_3(x) = 75 - 50 \left( \sum_{i=1}^{D} \frac{z_i^2}{4000} - \prod_{i=1}^{D} \cos\left(\frac{z_i}{\sqrt{i}}\right) + 1 \right) \le 0$$

$x \in [-500, 500]^D$

**Problem C14**

$$Min\, f(x) = \sum_{i=1}^{D-1} (100\,(z_i^2 - z_{i+1})^2 + (z_i - 1)^2)\,, \quad z = x + 1 - o, y = x - o$$

$$g_1(x) = \sum_{i=1}^{D} \left(-y_i cos\left(\sqrt{|y_i|}\right)\right) - D \leq 0$$

$$g_2(x) = \sum_{i=1}^{D} \left(y_i cos\left(\sqrt{|y_i|}\right)\right) - D \leq 0$$

$$g_3(x) = \sum_{i=1}^{D} \left(y_i sin\left(\sqrt{|y_i|}\right)\right) - 10D \leq 0$$

$x \in [-1000, 1000]^D$

**Problem C15**

$$Min\, f(x) = \sum_{i=1}^{D-1} (100\,(z_i^2 - z_{i+1})^2 + (z_i - 1)^2)\,, \quad z = x + 1 - o, y = (x - o)M$$

$$g_1(x) = \sum_{i=1}^{D} \left(-y_i cos\left(\sqrt{|y_i|}\right)\right) - D \leq 0$$

$$g_2(x) = \sum_{i=1}^{D} \left(y_i cos\left(\sqrt{|y_i|}\right)\right) - D \leq 0$$

$$g_3(x) = \sum_{i=1}^{D} \left(y_i sin\left(\sqrt{|y_i|}\right)\right) - 10D \leq 0$$

$x \in [-1000, 1000]^D$

**Problem C16**

$$Min\, f(x) = \sum_{i=1}^{D} \frac{z_i^2}{4000} - \prod_{i=1}^{D} \cos\left(\frac{z_i}{\sqrt{i}}\right) + 1\,, \qquad z = x - o$$

$$g_1(x) = \sum_{i=1}^{D} [z_i^2 - 100\cos(\pi z_i) + 10] \leq 0$$

$$g_2(x) = \prod_{i=1}^{D} z_i \leq 0$$

$$h_1(x) = \sum_{i=1}^{D} \left( z_i sin\left(\sqrt{|z_i|}\right)\right) = 0$$

$$h_2(x) = \sum_{i=1}^{D} \left( -z_i sin\left(\sqrt{|z_i|}\right)\right) = 0$$

$x \in [-10, 10]^D$

**Problem C17**

$$Min\ f(x) = \sum_{i=1}^{D-1} (z_i - z_{i+1})^2, \qquad z = x - o$$

$$g_1(x) = \prod_{i=1}^{D} z_i \leq 0$$

$$g_2(x) = \sum_{i=1}^{D} z_i \leq 0$$

$$h(x) = \sum_{i=1}^{D} \left( z_i sin\left(4\sqrt{|z_i|}\right)\right) = 0$$

$x \in [-10, 10]^D$

**Problem C18**

$$Min\ f(x) = \sum_{i=1}^{D-1} (z_i - z_{i+1})^2, \qquad z = x - o$$

$$g(x) = \frac{1}{D}\sum_{i=1}^{D} \left( -z_i sin\left(\sqrt{|z_i|}\right)\right) \leq 0$$

$$h(x) = \frac{1}{D}\sum_{i=1}^{D} \left( z_i sin\left(\sqrt{|z_i|}\right)\right) = 0$$

$x \in [-50, 50]^D$

# APPENDIX III
# LIST OF UNCONSTRAINED BENCHMARK PROBLEMS

## SCALABLE PROBLEMS

**Problem 1: Sphere Function**

$$f_1(x) = \sum_{i=1}^{D} x_i^2$$

$x \in [-5.12, 5.12]^D$

Optimal value = 0

**Problem 2: De Jong's Function**

$$f_2(x) = \sum_{i=1}^{D} i x_i^4$$

$x \in [-5.12, 5.12]^D$

Optimal value = 0

**Problem 3: Griewank Function**

$$f_3(x) = \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

$x \in [-600, 600]^D$

Optimal value = 0

**Problem 4: Rosenbrock Function**

$$f_4(x) = \sum_{i=1}^{D} [100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2]$$

$x \in [-100, 100]^D$

Optimal value = 0

**Problem 5: Rastrigin Function**

$$f_5(x) = \sum_{i=1}^{D} (x_i^2 - 10cos(2\pi x_i) + 10)$$

$x \in [-5.12, 5.12]^D$

Optimal value = 0

**Problem 6: Ackley Function**

$$f_6(x) = -20exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right)$$

$-exp\left(\frac{1}{D}\sum_{i=1}^{D} cos(2\pi x_i)\right) + 20 + e$

$x \in [-30,30]^D$

Optimal value = 0

**Problem 7: Alpine Function**

$f_7(x) = \sum_{i=1}^{D}|x_i sin(x_i) + 0.1x_i|$

$x \in [-10,10]^D$

Optimal value = 0

**Problem 8: Michalewicz Function**

$f_8(x) = -\sum_{i=1}^{D} sin(x_i)\left[\frac{sin(ix_i^2)}{\pi}\right]^{20}$ $x \in [-100,100]^D$

$x \in [0,\pi]^D$

Optimal value = 0

**Problem 9: Cosine Mixture Function**

$f_9(x) = \sum_{i=1}^{D} x_i^2 - 0.1\sum_{i=1}^{D} \cos(5\pi x_i)$

$x \in [-1,1]^D$

Optimal value = 0

**Problem 10: Exponential Function**

$f_{10}(x) = -exp(-0.5\sum_{i=1}^{D} x_i^2)$

$x \in [-1,1]^D$

Optimal value = -1

**Problem 11: Zakharov Function**

$f_{11}(x) = \sum_{i=1}^{D} x_i^2 + \left(\frac{1}{2}\sum_{i=1}^{D} ix_i\right)^2 + \left(\frac{1}{2}\sum_{i=1}^{D} ix_i\right)^4$

$x \in [-5.12,5.12]^D$

Optimal value = 0

**Problem 12: Cigar Function**

$f_{12}(x) = x_1^2 + 100000 \sum_{i=2}^{D} x_i^2$

$x \in [-10,10]^D$

Optimal value = 0

**Problem 13: Brown3 Function**

$f_{13}(x) = \sum_{i=1}^{D-1} \left[ (x_i^2)^{(x_{i+1}^2+1)} + (x_{i+1}^2)^{(x_i^2+1)} \right]$

$x \in [-1,4]^D$

Optimal value = 0

**Problem 14: Schewel Function**

$f_{14}(x) = \sum_{i=1}^{D} |x_i| + \prod_{i=1}^{D} |x_i|$

$x \in [-10,10]^D$

Optimal value = 0

**Problem 15: Salomon Function**

$$f_{15}(x) = 1 - cos\left(2\pi\sqrt{\sum_{i=1}^{D} x_i^2}\right) + 0.1\sqrt{\sum_{i=1}^{D} x_i^2}$$

$x \in [-100,100]^D$

Optimal value = 0

**Problem 16: Axis Parallel Hyperellipsoid Function**

$$f_{16}(x) = \sum_{i=1}^{D} i x_i^2$$

$x \in [-5.12,5.12]^D$

Optimal value = 0

**Problem 17: Pathological Function**

$$f_{17}(x) = \sum_{i=1}^{D-1}\left[0.5 + \frac{sin^2\sqrt{100x_i^2 + x_{i+1}^2} - 0.5}{1 + 0.001({x_i}^2 + x_{i+1}^2 - 2x_i x_{i+1})^2}\right]$$

$x \in [-100,100]^D$

Optimal value = 0

**Problem 18: Sum of Different Powers**

$$f_{18}(x) = \sum_{i=1}^{D} |x_i|^i$$

$x \in [-1,1]^D$

Optimal value = 0

**Problem 19: Step Function**

$$f_{19}(x) = \sum_{i=1}^{D} (\lfloor x_i + 0.5 \rfloor)^2$$

$x \in [-100,100]^D$

Optimal value = 0

**Problem 20: Quartic Function**

$f_{20}(x) = \sum_{i=1}^{D} i x_i^4 + random[0,1)$

$x \in [-1.28,1.28]^D$

Optimal value =0

**Problem 21: Inverted Cosine Wave Function Function**

$$f_{21}(x) = -\sum_{i=1}^{D-1} exp\left(-\left(\frac{x_i^2+x_{i+1}^2+0.5x_i x_{i+1}}{8}\right)\right) * cos\left(4\sqrt{x_i^2 + x_{i+1}^2 + 0.5x_i x_{i+1}}\right)$$

$x \in [-5,5]^D$

Optimal value =-D+1

**Problem 22: Neumaier3 Problem**

$f_{22}(x) = \left|\sum_{i=1}^{D}(x_i - 1)^2 - \sum_{i=1}^{D} x_i \, x_{i+1}\right|$

$x \in [-900,900]^D$

Optimal value =0

**Problem 23: Rotated Hyper Ellipsoid Function**

$f_{23}(x) = \sum_{i=1}^{D} x_i^2$

$x \in [-65.539,65.536]^D$

Optimal value =0

**Problem 24: Levi Montalvo 1 Function**

$f_{24}(x) = \frac{\pi}{D}[10sin^2(\pi y_1) + \sum_{i=1}^{D-1}(y_i - 1)^2\left(1 + 10sin^2(\pi y_{i+1})\right) + (y_D - 1)^2$

Where $y_i = 1 + \frac{1}{4}(x_i + 1)$

$x \in [-10,10]^D$

Optimal value =0

**Problem 25: Levi Montalvo 2 Function**

$f_{25}(x) = 0.1\left(sin^2(3\pi x_1) + \sum_{i=1}^{D-1}\left[(x_i - 1)^2\left(1 + sin^2(3\pi x_{i+1})\right)\right]\right) + (x_D - 1)^2\left(1 + sin^2(2\pi x_D)\right)$

$x \in [-5,5]^D$

Optimal value =0

**Problem 26: Ellipsoidal Function**

$f_{26}(x) = \sum_{i=1}^{D}(x_i - i)^2$

$x \in [-\mathrm{D}, \mathrm{D}]^D$

Optimal value = 0

**Problem 27: Shifted Parabola CEC 2005 Function**

$f_{27}(x) = \sum_{i=1}^{D} z_i^2 + bias$

z=(x-o), x=$[x_1, x_2, \ldots, x_D]$, O=$[o_1, o_2, \ldots, o_D]$

$x \in [-100,100]^D$

Optimal value = -450

**Problem 28: Shifted Schwefel CEC 2005 Function**

$f_{28}(x) = \sum_{i=1}^{D}\left(\sum_{j=1}^{i} z_j\right)^2 + bias$

z=(x-o), x=$[x_1, x_2, \ldots, x_D]$, O=$[o_1, o_2, \ldots, o_D]$

$x \in [-100,100]^D$

Optimal value = -450

**Problem 29: Shifted Greiwank CEC 2005 Function**

$f_{29}(x) = \sum_{i=1}^{D}\frac{z_i^2}{4000} - \prod_{i=1}^{D}\cos\left(\frac{z_i}{\sqrt{i}}\right) + 1 + f_{bias}$

z=(x-o), x=$[x_1, x_2, \ldots, x_D]$, O=$[o_1, o_2, \ldots, o_D]$

$x \in [-600,600]^D$

Optimal value = -180

**Problem 30: Shifted Ackley CEC 2005 Function**

$$f_{30}(x) = -20exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right)$$

-$exp\left(\frac{1}{D}\sum_{i=1}^{D} cos(2\pi x_i)\right) + 20 + e + f_{bias}$

z=(x-o), x=$[x_1, x_2, \dots, x_D]$, O=$[o_1, o_2, \dots, o_D]$

$x \in [-32,32]^D$

Optimal value = -140

## NON-SCALABLE PROBLEMS

**Problem 31: Goldstein Price Function**

$$f_{31}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 4x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)]$$
$$\times [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$$

$x \in [-2,2]^D$

$D$ = 2

Optimal value = 3

**Problem 32:** Six Hump Camel Function

$$f_{32}(x) = \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right)x_1^2 + x_1x_2 + (4x_2^2 - 4)x_2^2x_1 \in [-3,3]$$

$x_1 \in [-3,3]$

$x_2 \in [-2,2]$

$D$ = 2

Optimal value = -1.0316

**Problem 33:** Easom's Function

$$f_{33}(x) = -cos(x_1)cos(x_2)exp[-(x_1 - \pi)^2 - (x_2 - \pi)^2]$$

$x \in [-100,100]^D$

$D$ = 2

Optimal value = -1

**Problem 34: Beale Function**

$f_{34}(x) = (1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2^2)^2 + (2.625 - x_1 + x_1x_2^3)^2$

$x \in [-4.5,4.5]^D$

$D$ = 2

Optimal value = 0

**Problem 35: Colville Function**

$f_{35}(x) = 100(x_1 - x_2^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 + 10.1((x_2 - 1)^2 + (x_4 - 1)^2) + 19.8(x_2 - 1)(x_4 - 1)$

$x \in [-10,10]^D$

$D$ = 4

Optimal value = 0

**Problem 36: Branin Rcos Function**

$f_{36}(x) = \left(x_2 - \frac{5.1x_1^2}{4\pi^2} + \frac{5x_1}{\pi} - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)cos(x_1) + \quad 10$

$x_1 \in [-5,10]$

$x_2 \in [0,15]$

$D$ = 2

Optimal value = 0.3979

**Problem 37: Kowalik Function**

$f_{37}(x) = \left[a_i - \frac{x_1(b_i^2 + b_ix_2)}{b_i^2 + b_ix_3 + x_4}\right]^2$

Where a= {0.1957, 0.1947, 0.1735, 0.1600, 0.0844, 0.0627, 0.0456, 0.0342, 0.0323, 0.0235,0.0246}

b= {4.0, 2.0, 1.0, 0.5, 0.25, 0.1667, 0.125, 0.1, 0.0833, 0.07143, 0.0625}

$x \in [-5,5]^D$

$D$ = 4

Optimal value = 0.000307486

**Problem 38: 2D Tripod Function**

$f_{38}(x) = p(x_2)\big(1 + p(x_1)\big) + \big|x_1 + 50p(x_2)\big(1 - 2p(x_1)\big)\big| + \big|x_2 + 50\big(1 - 2p(x_2)\big)\big|$

With: $\begin{cases} p(u) = 1 \; if \; sign(u) \geq 0 \\ \quad\;\; = 0 \; if \; sign(u) < 0 \end{cases}$

$x \in [-100,100]^D$

$D$ = 2

Optimal value = 0

**Problem 39: Shekel foxholes Function**

$f_{39}(x) = \left[\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j+\sum_{i=1}^{2}\left(x_i - A_{ij}\right)^6}\right]^{-1}$

$x \in [-65.536, 65.536]^D$

$D$ = 2

Optimal value = 0.998

**Problem 40: Hartman3 Function**

$f_{40}(x)$ = -$\sum_{i=1}^{4} \alpha_i exp\left[-\sum_{j=1}^{3} A_{ij}\left(x_j - P_{ij}\right)^2\right]$

$x \in [0, 1]^D$

$D$ = 3

Optimal value = -3.86278

**Problem 41: Hartman6 Function**

$f_{41}(x)$ = -$\sum_{i=1}^{4} \alpha_i exp\left[-\sum_{j=1}^{6} A_{ij}\left(x_j - P_{ij}\right)^2\right]$

$x \in [0,1]^D$

$D$ = 6

Optimal value = -3.32237

**Problem 42: Shekel5 Function**

$f_{42}(x)$ = -$\sum_{j=1}^{5}\left[\sum_{i=1}^{4}\left(x_i - C_{ij}\right)^2 + \beta_j\right]^{-1}$

$x \in [0,10]^D$

$D$ = 4

Optimal value = -10.1532

**Problem 43: Shekel7 Function**

$f_{43}(x)$ = -$\sum_{j=1}^{7}\left[\sum_{i=1}^{4}\left(x_i - C_{ij}\right)^2 + \beta_j\right]^{-1}$

$x \in [0, 10]^D$

$D$ = 4

Optimal value = -10.4029

**Problem 44: Shekel10 Function**

$f_{44}(x)$ = -$\sum_{j=1}^{10}\left[\sum_{i=1}^{4}\left(x_i - C_{ij}\right)^2 + \beta_j\right]^{-1}$

$x \in [0,10]^D$

$D$ = 4

Optimal value = -10.5364

**Problem 45: Dekkers and Aarts Function**

$f_{45}(x)$ = $10^5 x_1^2 + x_2^2 - (x_1^2 + x_2^2)^2 + 10^{-5}(x_1^2 + x_2^2)^4$

$x \in [-20,20]^D$

$D$ = 2

Optimal value = -24777

**Problem 46: Shubert Function**

$f_{46}(x)$ = -$\sum_{i=1}^{5} icos\left((i+1)x_1 + 1\right) \sum_{i=1}^{5} icos\left((i+1)x_{2+1}\right)$

$x \in [-10, 10]^D$

$D$ = 2

Optimal value = -186.7309

# APPENDIX IV

# PERFORMANCE INDEX (PI)

In order to compare the relative performance of algorithms, the Performance Index (PI) defined in [39] has been used. This index gives weighted importance to success rate and average number of function evaluations for successful runs. Formula for calculating the values of PI's for both the algorithms have been given below:

$$PI = \frac{1}{N_P}\sum_{i=1}^{N_P}\left(k_1\alpha_1^i + k_2\alpha_2^i\right),$$

Where $\alpha_1^i = \frac{Sr^i}{Tr^i}$, and $\alpha_2^i = \begin{cases} \frac{Mf^i}{Af^i} & if \;\; Sr^i > 0 \\ 0 & if \;\; Sr^i = 0 \end{cases}$

$i = 1, 2, \ldots, N_P$

| | |
|---|---|
| $Sr^i$ | number of successful runs of $i$th problem |
| $Tr^i$ | total number of runs of $i$th problem |
| $Mf^i$ | minimum of average number of function evaluations of successful runs used by both the algorithms for obtaining the solution of the $i$th problem |
| $Af^i$ | average number of function evaluations of successful runs used by an algorithm for obtaining solution of the $i$th problem |
| $N_P$ | total number of problems evaluated |

$k_1$ and $k_2$ are the weights corresponding success rate and average number of function evaluations respectively such that $0 \leq k_1, k_2 \leq 1$ and $k_1 + k_2 = 1$. The PI graph is plotted by taking $k_1 = w, w = 0, 0.2, 0.4, 0.6, 0.8, 1$. Hence, $k_2 = 1 - k_1 = 1 - w$.