A Study of Function Landscape and Search Space with Evolutionary Algorithms

Author:
Sallam, Karam

Publication Date:
2018

DOI:
https://doi.org/10.26190/unsworks/20342

A Study of Function Landscape and Search Space with Evolutionary Algorithms

Karam Mohamed Goda Mohamed Sallam

A thesis submitted in fulfilment of the requirements

for the degree of

Doctor of Philosophy

School of Engineering and Information Technology

The University of New South Wales

Australia

August 2017

THE UNIVERSITY OF NEW SOUTH WALES
Thesis/Dissertation Sheet

Surname or Family name: **Sallam**

First name: **Karam Mohamed Goda Mohamed** Other name/s:

Abbreviation for degree as given in the University calendar: **Ph.D**

School: **School of Engineering and Information Technology** Faculty: **UNSW Canberra @ ADFA**

Title: **A Study of Function Landscape and Search Space with Evolutionary Algorithms**

Abstract 350 words maximum: (PLEASE TYPE)

Optimization techniques are used extensively to solve many real-world decision-making problems which have different characteristics and mathematical properties that make the process of finding their optimal solutions difficult. Evolutionary computation is one of the fastest growing areas that has been extremely successful when used to solve both unconstrained and constrained optimization problems (COPs) with a wide variety of properties.

In this thesis, using evolutionary algorithms (EAs) to solve optimization problems, whereby a search space is usually defined by the variables' bounds, is considered. However, in a COP, the feasible space, which is bounded by the constraint functions, may represent a relatively small portion of the search space. Existing EAs, using an analogy of black-box optimization, ignore specific information about the functions and search space even though details of their mathematical functions and properties are known. The objective of this research was to study the properties of the functions and search space to derive information that would be useful for designing effective and efficient EAs for solving optimization problems.

Although fitness landscape analysis (FLA) measures are helpful for judging a problem's complexity, they have rarely been used in the design of an EA. In most COPs, the optimal solution lies on the boundary of the feasible space. As this simple information may help to concentrate the search process in certain regions instead of the entire search space, this thesis proposes new EAs that use information from the function/problem landscape and search space for constrained problems to enhance the performances of algorithms for solving continuous optimization problems.
Firstly, a FLA-based differential evolution (DE) algorithm for solving unconstrained optimization problems is developed. Secondly, an algorithm for solving COPs, in which the landscape information from both the objective and constraint functions is considered, is proposed. Thirdly, a new technique for identifying the active constraints is developed and then a reduced search space around the active constraints determined, a concept applied with evolutionary algorithms. Finally, the information from the FLA and mechanism for reducing the search space are used to design an algorithm that incorporates multiple population-based algorithms in a single algorithmic structure.

FOR OFFICE USE ONLY Date of completion of requirements for Award:

Copyright Statement

Karam Mohamed Goda Mohamed Sallam

UNSW, Australia

Authenticity Statement

'I certify that the Library deposit digital copy is a direct equivalent of the final officially approved version of my thesis. No emendation of content has occurred and if there are any minor variations in formatting, they are the result of the conversion to digital format.'

Karam Mohamed Goda Mohamed Sallam
UNSW, Australia

Originality Statement

'I hereby declare that this submission is my own work and to the best of my knowledge and belief, it contains no material previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any other degree or diploma at UNSW or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by colleagues, with whom I have worked at UNSW or elsewhere, during my candidature, is fully acknowledged. I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.'

Karam Mohamed Goda Mohamed Sallam

UNSW, Australia

Acknowledgements

My thanks and appreciation go to Professor Ruhul Sarker for his guidance, support, time, patience and attention to my aspirations. I would also like to thank my co-supervisor, Dr. Daryl Essam, for his useful discussions and suggestions regarding my work, and also for the time and effort he spent reviewing my publications. I would also like to thank Dr. Saber Elsayed, for his useful discussions and suggestions. May ALLAH bless them all and benefit others through their knowledge and dedication.

I would like to thank the University of New South Wales at the Australian Defense Force Academy for providing me with a scholarship to study at that institution. I am also grateful for the assistance provided to me by the schools administration and IT staff. Also, i would like to thank Denise Russell for her suggestions for improving my thesis writing.

I am thankful to my wife (Seham), my little girl (Retaj), and my little boy (Omar) for their love and support, and all the means of comfort they provided me so that I could finish this thesis. I sincerely thank my father, mother, brothers , sisters and grandfather for their prayers for me and their encouragement for me to attain this degree. I would not have been able to complete this thesis without their motivation and supplies. May ALLAH bless and provide them with health and wellness.

I wish to express my appreciation to my friends and office mates for sharing their joyful moments with me. I also thank all of my friends in Egypt.

Abstract

Optimization techniques are used extensively to solve many real-world decision-making problems which have different characteristics and mathematical properties that make the process of finding their optimal solutions difficult. Evolutionary computation is one of the fastest growing areas that has been extremely successful when used to solve both unconstrained and constrained optimization problems with a wide variety of properties.

In this thesis, using Evolutionary Algorithms (EAs) to solve optimization problems, whereby a search space is usually defined by the variables bounds, is considered. However, in a constrained problem, the feasible space, which is bounded by the constraint functions, may represent a relatively small portion of the search space. Existing EAs, using an analogy of black-box optimization, ignore specific information about the functions and search space even though details of their mathematical functions and properties are known. The objective of this research was to study the properties of the functions and search space to derive information that would be useful for designing effective and efficient EAs for solving optimization problems.

The specific interests of this study are in analyzing the landscape and identifying the most attractive region for an effective search. Although Fitness Landscape Analysis (FLA) measures are helpful for judging a problems complexity, they have rarely been used in the design of an EA. In most constrained problems, the optimal solution lies on the boundary of the feasible space. As this simple information may help to concentrate the search process in certain regions instead of the entire

search space, this thesis proposes new EAs that use information from the function/problem landscape and search space for constrained problems to enhance the performances of algorithms for solving continuous optimization problems.

Firstly, a FLA-based Differential Evolution (DE) algorithm for solving unconstrained optimization problems is developed. Secondly, an algorithm for solving constrained problems, in which the landscape information from both the objective and constraint functions is considered, is proposed. Thirdly, a new technique for identifying the active constraints is developed and then a reduced search space around the active constraints determined, a concept applied with evolutionary algorithms. Finally, the information from the FLA and mechanism for reducing the search space are used to design an algorithm that incorporates multiple population-based algorithms in a single algorithmic structure.

All four versions of the developed algorithm are tested by solving standard benchmark problems and their results compared with those obtained from several state-of-the-art algorithms. In all cases, they achieve significant improvements, with the first, second, third and fourth obtaining respective savings in computational time and fitness evaluations of 15.1% and 15.7%, 69.0% and 33.0%, 36.69% and 11.88% and 22.01% and 20.73%, respectively.

Keywords

Evolutionary algorithms, constrained optimization, differential evolution, landscape analysis, reduced search space

List of Publications

Journal Articles:

1. K. M. Sallam, S. M. Elsayed, R. A. Sarker, and D. L. Essam, "Landscape-based adaptive operator selection mechanism for differential evolution," Information Sciences, vol. 418, pp. 383-404, 2017. (based on Chapter 3, sections 3.3 and 3.4)

2. K. M. Sallam, R. A. Sarker, and D. L. Essam, "Reduced search space mechanism for solving constrained optimization problems," Engineering Applications of Artificial Intelligence, vol. 65, pp. 147-158, 2017. (based on Chapter 5, sections 5.2 and 5.4)

Conference Papers:

3. K. M. Sallam, S. M. Elsayed, R. A. Sarker, and D. L. Essam, "Two-phase differential evolution framework for solving optimization problems," in Computational Intelligence (SSCI), 2016 IEEE Symposium Series on. IEEE, 2016, pp. 1-8. (Chapter 3)

4. K. M. Sallam, S. M. Elsayed, R. A. Sarker, and D. L. Essam, "Differential evolution with landscape-based operator selection for solving numerical optimization problems," in Intelligent and Evolutionary Systems: The 20th Asia Pacific Symposium, IES 2016, Canberra, Australia, November 2016, Proceedings. Springer, 2017, pp. 371-387. (Chapter 3)

5. K. M. Sallam, S. M. Elsayed, R. A. Sarker, and D. L. Essam, "Landscape-based Differential Evolution for Constrained Optimization Problems," submitted, in in Computational Intelligence (SSCI), 2017 IEEE Symposium Series on. IEEE, 2017. (Chapter 4)

6. K. M. Sallam, S. M. Elsayed, R. A. Sarker, and D. L. Essam, "Multi-method based orthogonal experimental design algorithm for solving cec2017 competition problems," in Evolutionary Computation (CEC), 2017 IEEE Congress on. IEEE, 2017, pp. 1350-1357. (Chapter 6)

7. K. M. Sallam, R. A. Sarker, D. L. Essam, and S. M. Elsayed, "Neurodynamic differential evolution algorithm and solving cec2015 competition problems," in Evolutionary Computation (CEC), 2015 IEEE Congress on. IEEE, 2015, pp. 1033-1040.

Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviations	Description
ACD	active constraint determination
ACO-MH	ant colony optimization meta-heuristic
AOS	adaptive operator selection
AP	adaptive pursuit
AUC	area under the curve
BBOB	black-box optimization benchmarking
BLX-alpfa	blend crossover
CARP	capacitated arc routing problem
CEO	constrained evolutionary optimization
CHTs	constraint handling techniques
CI	computational intelligence
CMA	covariance matrix adaptation
COCO	comparing continuous optimization
COPs	constrained optimization problems
DE	deferential evolution
DM	dispersion metric
DV	difference vector
EAs	evolutionary algorithms
ELA	exploratory landscape analysis
EP	evolutionary programming
ES	evolution strategy
FDC	fitness distance correlation
FLA	fitness landscape analysis

FVs	function values
G3	generalized generation gap
GA	genetic algorithm
HMHH	heterogeneous meta-hyper-heuristic
HSD	heuristic space diversity
ICA	information characteristics
ILNS	information landscape negative searchability
IMM	Istanbul metropolitan municipality
LHD	Latin hyper-cube design
LPSR	linear population size reduction
LS	length scale
MAP	multi-armed bandit
MMTS	modified multiple trajectory search
MOPs	multi-objective optimization problems
NLD	normalized landscape
NPM	normalized performance measure
PAP	population-based algorithm portfolio
PCX	parent-centric crossover
PIC	partial information content
PMs	penalty methods
PSO	particle swarm optimization
QAP	quadratic assignment problem
R2S	reduced search space
SI	swarm intelligence
SoR	sum-of-ranks
SPX	simulated binary crossover
SPX	simplex crossover

SR	stochastic ranking
SSRT	search space reduction technique
TC	triangular crossover
UCB	upper confidence bound
UNDX	unimodal crossover

Chapter 1

Introduction

This chapter begins by providing the background to and description of the research. Then, the objective of this study and its contributions to scientific knowledge are discussed. Finally, the organization of this thesis is presented.

1.1 Background

Optimization plays an important role in many practical decision-making processes. For several decades, it has attracted the attention of a large number of researchers and practitioners for solving complex optimization problems in many fields, such as scheduling a supply chain, controlling an infrastructure design, allaying natural disasters, organizing transportation systems and optimizing industrial processes, which can be mainly categorized as unconstrained and constrained. Its aim is to determine the values of a problem's design variables by optimizing (either maximizing or minimizing) one or more of its objective functions [**?**]. The importance of using optimization methods to solve real-world application problems can be demonstrated using some examples. The estimated annual savings from scheduling crash tests at the Ford Motor Company are $1 million [1]. By optimizing the locations of fire stations, the Istanbul Metropolitan Municipality (IMM) was able to increase the area covered by the citys fire station from 58.6% to 85.9% [2]. Applying optimization enabled Chevron (one of the worlds leading integrated energy companies) to provide approximately $1 billion per year in benefits [3]. Optimizing ship routing at the Danaos Corporation maximized its profits by $4.5 million in 2011 and helped the company minimize carbon emissions, reduce staff workloads and increase customer satisfaction [4]. The estimated financial benefits from 2011 to 2028 obtained by optimizing water routing and hydroelectric generation on the Columbia River are expected to be between $765 and $952 million [5]. Using optimization has enabled Procter and Gamble to earn approximately $70 billion in annual revenue [6] and the Bank of New York to solve collateral-management challenges involving short-term secured loans which reduced intra-day credit risk by $1.4 Trillion [7].

Researchers and practitioners have used both traditional optimization and Computational Intelligence (CI) approaches to solve complex optimization problems. However, the former encounters many difficulties [8], such as their convergence to a near-optimal or optimal solution relies on the initial solution, they require specific mathematical properties, such as continuity, convexity and differentiability, to be satisfied and they may need to simplify a problem for mathematical representation by making different assumptions [9]. On the contrary, CI approaches, which are resilient to dynamic changes, have the capability to self-organize, do not require particular mathematical characteristics to be satisfied and can evaluate several solutions in parallel, are widely used in practice [10] [11]. However, there is no guarantee that they will obtain the optimum solutions and the quality of their solutions relies on the particular algorithm's design, the selection of its operators and its parameter settings.

Of current CI approaches, evolutionary algorithms (EAs), such as differential evolution (DE) [12], Evolution Strategy (ES) [13], Evolutionary Programming (EP) [14] and Genetic Algorithms (GAs) [15], are population-based algorithms that use some types of selection, crossover and mutation operators to generate new candidates and guide the search towards optimal solutions. Of them, DE has been widely applied in many fields, gained popularity for solving continuous optimization problems and has shown its superiority over other methods for solving complex optimization problems with different characteristics [16, 17]. Over the years, several mutation strategies aimed at improving the performance of DE for solving continuous optimization problems have been proposed [18, 19]. However, none performs well for all kind of problems [20–22] and, moreover, no single algorithm has been found to be the best for solving all the problems considered in the literature [23, 24].

A function landscape is represented by a surface in the search space that reflects the fitness function value of each solution. A fitness landscape analysis (FLA) has become a popular method for analyzing the characteristics of optimization problems, such as their ruggedness, complexity, modality, presence of funnels, neutrality and variable separability. It consists of a set of solutions (populations of individuals) in which each individual has a fitness value (also known as an objective function value) and a neighborhood operator that may be expressed as a distance measure [25]. Many FLA methods for classifying a problem as either easy or difficult, such as auto-correlation [26, 27], fitness distance correlation [28], dispersion metric [29], length scale [30] and information characteristics analysis [31], have been suggested. Researchers and practitioners have used FLA to determine a function's modality [32] [25], the sizes of basins of attraction [33], the existence of funnels in a fitness landscape [34] and a problem's ruggedness [26, 27, 30]. It is expected that using such methods during the evolutionary process may help to determine the most appropriate algorithm and/or search operators for a problem.

In a Constrained Optimization Problem (COP), the functional constraints generate a feasible space that is usually much smaller than the search space defined by the variables' bounds. Also, as all constraints with values of zero at the optimal solution are considered active ones, equality constraints are always active. In many real-world problems, the optimal solution lies on the boundary of the feasible region, that is, the intersection of the active constraints [35], with one of the most relevant characteristics of a boundary search its reduced search space as it explores only a portion of the whole search space [36]. Due to its importance, many attempts have been made to search the boundary of the feasible region [35–38]. It is expected that designing an EA that uses a mechanism to focus the search process around the previously determined active constraints could help to obtain the optimal solution more quickly and solve a COP more efficiently than using only an EA.

1.2 Problem Description and Research Gaps

As previously discussed, optimization problems can be either unconstrained or constrained. They have different characteristics and mathematical proprieties, such as their objective functions and constraints may be unimodal or multi-modal, continuous or discontinuous, linear or nonlinear, and their variables discrete or real. Also, the feasible region of a COP can be either a small or large portion of the search space and either one bounded region or a set of disjointed ones or, in some practical problems, even unbounded [**?**]. These different characteristics make the process of locating the optimal solution challenging [**?**].

Of the many solution approaches developed, EAs have demonstrated great success in solving both unconstrained problems and COPs with a wide variety of properties. However, their performances are highly dependent on the algorithms design, and the selection of its operators and parameter settings. Also, no single algorithm and/or search operator performs consistently well for all the types of optimization problems defined in the literature [17, 39]. Moreover, the process for determining the most suitable EA and/or its search operators for solving a particular problem could be carried out using useful information about the problems landscape rather than a trail-and-error approach [?].

FLA has become a popular method for understanding the complexity of an optimization problem for which an EA either succeeds or fails [25]. However, such an approach is often carried out in an offline mode, i.e., the existing experiments were conducted independently from the evolutionary process [25]. Also, it was computationally expensive [40], and limited work has been carried out for solving optimization problems. Although the use of function and search space specific information may help in selecting the proper evolutionary operator in solving a particular problem, it has not been fully explored in designing such an optimization algorithm.

In a COP, the feasible space, which is bounded by the constraint functions, may represent a relatively small portion of the search space [41] and, in many cases, the optimal solution lies in the boundary of the feasible space [35, 36] (the intersection of the active constraints). Although EAs have been successfully used to solve COPs, they still suffer from the drawback of wasting a considerable amount of time (fitness evaluations) searching ineffective areas in the search space [42]. Therefore, it is expected that, using the simple information that the optimal solution lies on the boundary of the feasible region may help to concentrate the search process in certain regions instead of the entire search space. This will improve an algorithms performance by reducing the computational time and number of fitness evaluations it requires to reach the optimal solution.

1.3 Research Objective

Motivated by the research gaps discussed above, ***the overall goal in this thesis, is to propose new EAs that utilize information from a function's/problem's landscape and search space to enhance the performance of the algorithms in solving continuous optimization problems***.

To achieve this goal, this studys sub-objectives are to:

1. analyze the performances of different DE variants;
2. propose a new algorithm which uses landscape information to select the most suitable DE, from a set, during the evolutionary process for solving unconstrained optimization problems;
3. propose an algorithm that uses the strength of more than one DE and chooses the best based on the objective function and constraint landscapes for solving COPs;
4. propose a mechanism for reducing the search space by focusing on the area around the active constraints; and
5. propose an algorithm that uses multiple EAs and both a Reduced Search Space (R2S) mechanism and landscape information.

1.4 Contributions to Scientific Knowledge

The general scientific contribution of this thesis is the development of methods that use information from a problems landscape and reduce the search space to enhance the performances of EAs. Firstly, a FLA-based DE algorithm for solving unconstrained optimization problems is proposed. Secondly, an algorithm for solving COPS, in which the landscape information from both the objective and constraint functions are considered, is proposed. Thirdly, a new technique for identifying the active constraints and then determining a reduced search space around them is developed and applied with several EAs. Finally, the information from a FLA and R2S mechanism are used in the design of an algorithm that incorporates the following multiple population-based algorithms in a single algorithmic structure.

- **Landscape-based DE algorithm for solving unconstrained optimization problems:** it contains a fitness landscape measure for determining the best-performing DE operator in a pool during the evolutionary process. The performances of different DE variants for solving unconstrained problems are analyzed, with the experimental results demonstrating this algorithms superiority over both itself with different selection mechanisms and several state-of-the-art algorithms, and its capability to save 15.1% of computational time and 15.7% of fitness evaluations.

- **Landscape-based algorithm for solving COPs:** in it, information from the objective function and constraint landscapes is used to choose the best operator from a pool during the evolutionary process and its components are analyzed. This new design demonstrates its superiority over both several state-of-the-art algorithms and itself with other selection mechanisms, and achieves 69.0% and 33.0% savings in computational time and fitness evaluations, respectively.

- Reduced search space mechanism for solving COPs: it determines the active constraints based on the entire populations and focuses the search space around them. It is incorporated in six state-of-the-art algorithms, with the results showing that it significantly improves the performances of these algorithms and leads to savings in computational time of 36.69% and fitness evaluations of 11.88%.

- **Multi-EAs algorithm for solving COPs:** this novel algorithmic design uses multiple EAs and both the R2S mechanism and landscape measure. Its different components are analyzed and then the results obtained from its best variants compared with those from several state-of-the-art algorithms, with the results demonstrating its usefulness as it achieves 22.01% and 20.73% savings in computational time and fitness evaluations, respectively.

1.5 Organization of Thesis

This thesis consists of the following 7 chapters.

- Chapter 1: Introduction
- Chapter 2: Background Study and Literature Review
- Chapter 3: Landscape-based Algorithm for Unconstrained Optimization Problems
- Chapter 4: Landscape-based Algorithm for Constrained Optimization Problems
- Chapter 5: Reduced Search Space Mechanism for Constrained Optimization Problems
- Chapter 6: Multi-EAs Framework for Constrained Optimization Optimization Problems
- Chapter 7: Conclusions and Future Research Directions

Chapter 1 presents an introduction to this thesis, beginning with an overview of this research field, followed by a description of the problem as well as the motivation for, and objectives of, this study. Its scientific contributions are then discussed and its organization presented.

In Chapter 2, a review of the basic fundamentals of the topics covered in this thesis is presented. Firstly, unconstrained optimization problems and COPs are introduced and the EAs available for solving them described. Then brief descriptions of landscape analysis measures and constraint-handling techniques are provided. Next, a review of EAs, and FLA for solving unconstrained and constrained optimization problems are provided, and finally, a review of boundary search methods and search space reduction are discussed.

In Chapter 3, brief descriptions of the benchmark unconstrained optimization problems used are provided. A general multi-operator DE framework, in which more than one DE mutation strategy and the problems landscape information are used, is proposed. Also, an analysis of the algorithm's components is undertaken and, finally, its results presented and compared with those of several state-of-the-art algorithms.

In Chapter 4, descriptions of the benchmark COPs used are provided. An algorithm for solving COPs, in which information from the objective function and constraint landscapes is used to choose the best operator from a pool during the evolutionary process, is described. The different components of the proposed algorithm are described, with the experimental results obtained from it for solving different sets of COPs analyzed and compared with those from several state-of-the-art algorithms.

In Chapter 5, a new R2S mechanism is introduced and then integrated with six state-of-the-art algorithms for solving COPs. The different components of the proposed mechanism are described, with the experimental results obtained from algorithms with and without it analyzed and compared.

In Chapter 6, a new multi-method algorithm that uses information from the search space and a problem's landscape is proposed. Its various components are presented and the experimental results obtained from it for solving different sets of COPs analyzed and compared with those from several state-of-the-art algorithms.

In Chapter 7, this studys main findings and conclusions are discussed, and possible future research directions suggested.

Chapter 2

Background Study and Literature Review

This chapter provides an overview of the basic fundamentals of the topics covered in this thesis. It begins with a brief discussion of unconstrained and constrained optimization problems and reviews computational intelligence (CI) methods for solving them. Then brief descriptions of landscape analysis measures and constraint-handling techniques are provided. Next, a review of EAs, and FLA for solving unconstrained and constrained optimization problems is provided. Finally, a review of boundary-search methods and search space reduction mechanism is presented.

2.1 Optimization Problems

Optimization problems are commonly found in many real-world applications, such as scheduling, industrial process optimization, vehicle routing in large-scale networks and gene recognition in bio-informatics. Their task is to decide the values of the design variables via optimizing (either maximizing or minimizing) one or more objective functions [43]. These problems can be categorized into two different categories, unconstrained and constrained. They may comprise distinct kinds of variables, such as integer and real, and constrained ones may also have equality and/or inequality constraints. The objective and constraint functions can be either unimodal or multimodal, linear or nonlinear and continuous or discontinuous. These classifications are briefly discussed below.

Classification based on existence of constraints - as previously stated, an optimization problem is called a constrained one (COP) if its formulation contains one or more constraints, otherwise it is considered an unconstrained one.

Classification based on types of design variables - based on the types of its decision variables, an optimization problem can be discrete (integer), whereby the values of the design variables must be discrete, or continuous (real).

Classification based on types of functions included - in this category, an optimization problem is classified as being a linear, nonlinear, geometric or quadratic programming one. An optimization problem is called linear, if its objective function and all its constraints are linear functions of the design variables. It is called nonlinear, if its objective function or any of its constraints is nonlinear. If its objective function is quadratic and all its constraints are linear functions of its decision variables, an optimization problem is called a quadratic optimization problem. An optimization problem is called a geometric programming problem if its fitness function and constraints can be expressed as the sum of its power terms.

Classification based on the number of objective functions - there are two categories: 1) a single-objective problem is one in which there is only one objective function; and 2) a multi-objective problem has more than one objective function.

This thesis deals with continuous, single-objective, unconstrained and constrained optimization problems, which are defined in the following sections.

2.1.1 Unconstrained optimization problems

An unconstrained optimization problem consists of an objective function and a bound constraint, with the structure of those considered in this thesis being defined as:

$$\begin{aligned} &\text{minimize or maximize } f(\overrightarrow{x}) \\ &\text{subject to:} L_j \leq x_j \leq U_j, j = 1,\, 2,\, ...,\, D \end{aligned} \tag{2.1}$$

where $f(\overrightarrow{x})$ is the fitness function (objective function), $\overrightarrow{x} = [x_1,\, x_2,\, ...,\, x_D]$ a decision variables vector with D dimensions, x_j, has lower and upper bounds L_j and U_j, respectively.

2.1.2 Constrained optimization problems (COPs)

Optimizing a COP is more challenging than optimizing an unconstrained one, because to obtain an optimal solution, additional constraints have to be satisfied. A COP can be expressed mathematically as:

$$\begin{aligned} \text{minimize or maximize } & f(\overrightarrow{x}) \\ \text{subject to: } & g_k(\overrightarrow{x}) \leq 0,\, k = 1,\, ...,\, s \\ & h_e(\overrightarrow{x}) = 0,\, e = 1,\, ...,\, q \\ & L_j \leq x_j \leq U_j \end{aligned} \tag{2.2}$$

This problem has s inequality constraints, $g_k(\overrightarrow{x})$, q equality constraints, $h_e(\overrightarrow{x})$, and each variable, x_j, has a lower and upper bound, L_j, U_j, respectively. The aim is to determine the values of all variables, $x_1,\, x_2,\, ...,\, x_D$, that minimize (or maximize) the objective function, $f(\overrightarrow{x})$, while satisfying all the constraints, including the bound ones. The feasible region of this problem can be either a small or large portion of the search space, and either a one bounded region or a set of disjointed ones and, in some practical problems, even unbounded [43]. The optimal solution may lie either on the boundary of feasible space or inside it.

Over the years, researchers and practitioners have used both traditional and computational intelligence (CI) approaches, such as EAs and Swarm Intelligence (SI), to solve complex optimization problems. However, the former encounter many difficulties [8], such as: 1) their convergence to a near-optimal or optimal solution relies on the initial solution; 2) they require specific mathematical properties, like continuity, convexity and differentiability, to be satisfied; and 3) they may need to simplify a problem by making different assumptions for the convenience of mathematical modeling [9]. Therefore, researchers have begun to use CI approaches because of their many advantages, for example, as evolutionary algorithms (EAs) are resilient to dynamic changes and have the capability of self-organization, they do not need particular mathematical characteristics to be achieved, can evaluate several solutions in parallel, are widely used in practice [10], and they have often been proven to work better than traditional methods [11]. However, there is no guarantee that EA-based methods can obtain exact solutions and the quality of their solutions relies on the design of algorithms, selection of operators and settings of their parameters. In fact, as no single EA has been found to be the best for solving all types of optimization problems, generally, using more than one algorithm in a single framework may help to overcome this limitation [23, 24]. In the following section, an overview of EAs is presented.

2.2 Evolutionary Algorithms (EAs)

EAs are nature-inspired optimization methods that are used to evolve populations of individuals from generation to generation, to produce increasingly better solutions to an optimization problem [44]. They have a long history of successfully tackling different optimization problems [43].

Over the years, many EAs, such as differential evolution (DE) [12], evolutionary programming (EP) [14], genetic algorithms (GAs) [15] and evolution strategy (ES) [13], have been proposed. All EAs have similar structures, with the only differences between them being, their order and type of operations, and the ways in which they generate their initial population [45]. In their internal processes, individuals are represented (encoded) in different ways, i.e., binary, integer, real-value and string, and common operators, such as mutation, recombination (crossover) and/or selection, are used to generate new candidates. In a recombination operator, two or more selected parents generate one or more offspring. In contrast, a mutation operator is applied to one candidate in order to produce a perturbed candidate with the hope of maintaining diversity. Each new candidate will only survive to the next population only if it is better than its parent. These steps are iterated until a good solution is obtained or a stopping criterion is met [45].

As, based on the extant literature, DE has been acknowledged as a successful optimization algorithm for solving various optimization problems in the continuous domain [16, 17, 43, 46]. The following sub-sections briefly describe the main features of a few well-known EAs.

2.2.1 Differential evolution (DE)

DE is an EA variant which was proposed by Storn and Price [12]. It is popular because it usually converges fast, is simple to implement and the same parameter values can be applied for various optimization problems. In the literature, it is shown to perform better than several other EAs for a wide range of problems [43]. DE is consistent and reliable for solving many real-life nonlinear COPs, such as those in power and chemical systems, communication, and pattern reconciliation [47]. It starts with an initial population, and then constructs a new solution from current or existing ones. To do this, three operators, "mutation, crossover and selection", are employed during the search process. Its structure and main operators are described below. For more details, readers are referred to [16, 17, 46].

2.2.1.1 Mutation

DE uses a mutation operator before crossover. In a simple mutation (DE/rand/1), three candidates are randomly selected and a mutant individual is produced by multiplying a scaling factor (F) by the difference (DV) between two individuals, with their result summed to a third one as [43]:

$$\vec{V}_{z,G} = \vec{x}_{r_1,G} + F \times (\vec{x}_{r_2,G} - \vec{x}_{r_3,G}) \tag{2.3}$$

where $r_z \in [1, NP]$, $z = 1, 2, 3$ are random numbers such that $z \neq r_1 \neq r_2 \neq r_3$, NP is the number of individuals (population size), F is a positive number used to scale DV, and G is the current generation. In the literature, several mutation operators, with different searching capabilities [43] , such as: DE/best/1 [48], DE/rand-to-best/1 [49], DE/current-to-rand/1 [50], DE/current-to-best/1 [48], DE/rand/2 [48], and DE/current-to-ϕbest/1 [51], have been proposed.

2.2.1.2 Crossover

Generally, after mutation strategy, crossover operator is employed, to maintain population's diversity [17]. It is used to generate an offspring by mixing variables of the parents and of the mutated vectors. The two common simple crossover operators are exponential and binomial. In the former, firstly, an integer, c, in the decision space is randomly selected so that $c \in [1,\ D]$, where D is the number of variables, c acts as a starting point of the target vector for crossover. Another integer, C, selected from $[c, D]$ denotes the number of elements of the donor vector that are used in the offspring vector. Once c and C are selected, an offspring is generated as:

$$u_{i,j,G} = \begin{cases} v_{i,j,G} & \text{for } j = \langle c \rangle_D \,, \, \langle c+1 \rangle_D \,, \, ..., \, \langle c+C-1 \rangle_D \\ x_{i,j,G} & \text{otherwise} \end{cases} \tag{2.4}$$

where $j = 1,\ 2,\ ...,\ D$ and the angular bracket, $\langle c \rangle_D$, denotes a modulo function of modulus D with a starting index of c.

The binomial crossover, which is applied on every j variable if a random number is less than the crossover rate (Cr), is calculated by:

$$u_{i,j,G} = \begin{cases} v_{i,j,G} & \text{if } (rand \leq Cr \text{ or } j = j_{rand}) \\ x_{i,j,G} & \text{otherwise} \end{cases} \tag{2.5}$$

where $rand \in [0, 1]$ and $j_{rand} \in [1,\ 2,\ \ldots,\ D]$ are randomly selected to guarantee that at least one value is obtained from the offspring [43].

2.2.1.3 Selection

DE uses a selection operator to choose a vector from the parent and offspring ones based on their best fitness function values (FVs). In this step, the value of the objective functions for both the parent and offspring vectors are compared, and the one with better objective FV survives to the next generation according to:

$$x_{i,G+1} = \begin{cases} u_{i,G+1} & \text{if } (f(u_{i,G+1}) \leq f(x_{i,G}) \text{ or } j = jrand) \\ x_{i,G} & \text{otherwise} \end{cases} \tag{2.6}$$

From the literature, it is clear that no single DE algorithm performs well for all kinds of optimization problems. Mezura-Montes et al. [52] compared the

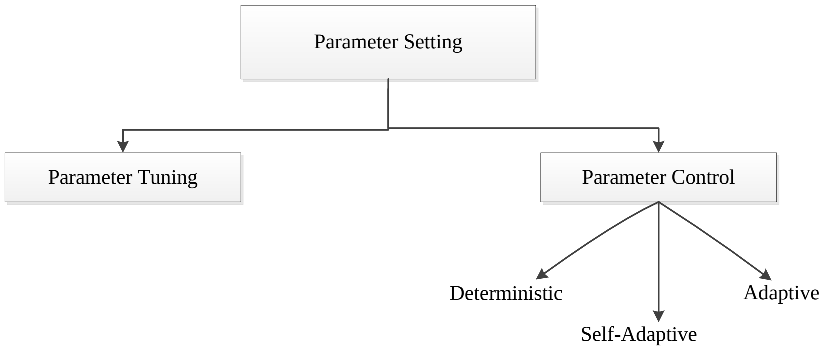


Figure 2.1: Taxonomy of parameter setting

performances of a number of them for solving unconstrained global optimization problems. Jeyakumar and Shanmugavelayutham [53] carried out a convergence analysis of 14 DE algorithms for solving 14 unconstrained global optimization functions, with the results obtained from these two studies demonstrated the fact that no single DE algorithm performs well for all kinds of optimization problems.

2.2.1.4 Control parameters of DE

In the literature, it is clear that the performance of DE is sensitive to the choice of the mutation and crossover operators, as well as their associated parameters (the amplification factor, F, crossover rate, Cr, and population size, NP) [43]. F is used to make a balance between exploration and exploitation. Smaller values of F, means av increase in the convergence rate, while larger values means maintain population diversity. Cr is used to determine the rate of change in an individual of the population.

Setting the control parameter values in EAs generally, and in DE specifically, can be divided into two processes, parameter tuning and parameter control [54]. Figure 2.1 shows the taxonomy of parameter setting.

2.2.1.5 Parameter tuning

In it good parameter values are found before an algorithm is run, and then are not changed during a run, that is, all the parameter values are static during the search process. However, the main disadvantage of parameter tuning is its lack of flexibility during this process [55]. Some of the work using static parameter values is discussed below.

It was recommended by Storn and Price [56] that NPs should be between $5 \times D$ to $20 \times D$, and suggested that a good initial choice of F could be 0.5. Gamperle et al. [57] claimed that choosing the proper parameters for DE is challenging. In [57], different parameter settings for DE are evaluated and it was found that NPs of between $3D$ and $8D$, with $F = 0.6$ and Cr in the range of [0.3, 0.9], are good choices. Ronkkonen et al. [58] advised using $F \in [0.4, 0.95]$ and $Cr \in [0, 0.2]$ for separable functions and $Cr \in [0.9, 1]$ for non-separable ones. However, the selection of DE's appropriate parameter is yet an open question.

2.2.1.6 Parameter control

In these methods, the parameter values change during the running of the algorithm, can be classified as deterministic, adaptive and self-adaptive, as shown in 2.1.

Deterministic methods: the values of the parameters are updated based on some deterministic rules, regardless of any feedback from the algorithm, i.e., a rule is applied when a number of generations has passed since the last time that rule was triggered [59]. As an example, Qian and Li [60] proposed a strategy for adapting F using the diversity of the current population and the number of members of the current Pareto-front. However, a fixed value was used for Cr.

Adaptive methods: parameter values are updated based on some form of feedback from the search, which is used to define the magnitude of the changes to the parameters and/or its direction [54, 61–68]. For example, Zhang and Sanderson [64] introduced an adaptive DE algorithm with an external archive, known as JADE. In it, for each $\overrightarrow{x}_i$, Cr and F are generated according to normal and Cauchy distributions of the means, μCr, μF, respectively, and a standard deviation of 0.1.

Self-adaptive methods: parameters are coded into chromosomes, and their values are updated by applying evolutionary steps (mutation, crossover and selection) [69].

For example, Brest et al. [69] presented a self-adaptive DE algorithm, known as jDE, in which the control parameters Cr and F are determined using a self-adaptive mechanism, by first encoding them into each solution vector and by then adapting them by means of evolution. In it, the values of Cr and F are randomly initialized for every individual in the population, and then they are randomly adjusted in subsequent generations based on two probabilities of τ_1 and τ_2. These new values are used to generate new solutions.

Abbas [70] developed self-adaptive operators for solving Multi-objective Optimization Problems (MOPs). Its crossover and mutation rates are initialized from a uniform distribution in $[0, l]$ and then the parameters are updated using the same concept as its mutation strategy equation. Later this work was extended [71, 72]. Teo [73] developed a DE algorithm with a self-adaptive population size based on self-adaptive Pareto DE algorithm [70]. Tirronen and Neri [74] proposed a technique to adapt NP based on fitness diversity, while population diversity is used to adapt the F and Cr values. Fan and Yan [66] presented a self-adaptive DE (SDE) algorithm, in which F and Cr can be updated by zoning the evolution.

In summary, DE algorithms with adaptive and self-adaptive techniques have often been more successful than classical ones [66, 69, 71, 72]. However, as no single parameter has been proved to be the best for all kinds of test problems [22, 75, 76], researchers have begun using ensembles of operators, such as mutation and crossover strategies, along with their associated parameter values. Ali et al. [76] developed a new DE algorithm, sTDE-dR, in which a different adaptive method is used to define the values of F and Cr. The population is divided into k tribes, with a different adaptive mechanism used for each one to update these control parameters.

2.2.2 Genetic algorithm (GA)

A GA is one of the most popular population-based algorithms for solving optimization problems. Like DE, GA uses different operators, including crossover, mutation and elitism-preserving techniques [15]. Crossover is the process for generating a new solution by exchanging genes among chromosomes, while elitism ensures the best so far found solution is kept and a mutation operator is used to maintain diversity to escape local solution. GA has been successfully used for solving many difficult real-world problems. However, it faces more difficulties than DE when handling multimodal problems [77]. Its primary operators are described below.

2.2.2.1 Genetic representation

Due to variations in the nature of problems, different genetic representations are required, such as binary, permutation and real-value, to represent individuals [43, 78].

2.2.2.2 Selection

This is the process of selecting individuals from a population according to their fitness function values (FVs), which are evaluated with respect to a given objective function. Generally speaking, the highest-ranked chromosome will have a greater possibility of being selected than the others and the worst will be eliminated. Several selection methods, such as roulette wheel, rank and tournament [43], have been introduced.

2.2.2.3 Crossover:

Once the selection process is completed, a crossover operator is triggered to generate new offspring by exchanging sub-parts of two parent candidates. Several crossover operators have been introduced [43]. Some crossover operators suitable for discrete problems are the uniform, single-point and multi-point [79]. Others for continuous problems are the average [80], flat [81], blend (BLX-α) [82], simulated binary (SBX) [83], unimodal distribution (UNDX) [84], parent-centric (PCX) [85], simplex (SPX) [86], and triangular (TC) [87]. Elsayed et al. [88] compared the performances of 10 GA variants for solving COPs, with results that showed SBX with non-uniform mutation was the best, and also none of them was suitable to solve all kind of problems defined in the literature.

SBX was originally proposed by Agrawal and Deb [83]. It has been widely used in practice and has been found to work better for test problems with a continuous search space, where the optimum point has unknown bounds [88]. It has also been successfully applied to solve COPs [43, 88, 89]. To generate new individuals using SBX, firstly, two parents, $\overrightarrow{x}_j^1 = (x_1^1, x_2^1, ..., x_D^1)$ and $\overrightarrow{x}_j^2 = (x_1^2, x_2^2, ..., x_D^2)$, are chosen from the entire population and then two offspring, $\overrightarrow{y}_j^1 = (y_1^1, y_2^1, ..., y_D^1)$ and $\overrightarrow{y}_j^2 = (y_1^2, y_2^2, ..., y_D^2)$ are generated as follows. Firstly β_{qj} is calculated by equating the area under the probability curve equal to u_j as:

$$\beta_{qj} = \begin{cases} (2u_j)^{1/\eta_c+1} & \text{if } u_j \leq 0.5 \\ (\frac{1}{2(1-u_j)})^{1/\eta_c+1} & \text{otherwise} \end{cases} \tag{2.7}$$

where u_i is a random number uniformly generated in the range of [0, 1], and η_c is a user-defined parameter, called the SBX distribution index.

After obtaining the value of β_{qj}, the offspring is calculated as:

$$y_j^1 = 0.5[(1+\beta_{qj})x_j^1 + (1-\beta_{qj})x_j^2], \forall j = 1, 2, \cdots, N_x \tag{2.8}$$

$$y_j^2 = 0.5[(1-\beta_{qj})x_j^1 + (1+\beta_{qj})x_j^2]\, j = 1, 2, \cdots, N_x \tag{2.9}$$

2.2.2.4 Mutation:

a mutation operator can be considered a random deformation of strings with a certain probability. Its primary purpose is to preserve genetic diversity over generations, and as a result, avoid local optima. There are many mutations for both discrete and continuous problems, such as Gaussian [90], logarithmic, uniform [91], non-uniform [91] and polynomial [44]. As in [92], non-uniform mutation functions performed well in solving COPs, its main steps are described below.

A child is mutated as [43]:

$$y_j\acute{(}G) = y_j(G) + \delta_j(G) \tag{2.10}$$

$$\delta_{j,G} = \begin{cases} (U_j - y_j(G))\left(1 - r(G)^{\left(1-\frac{G}{N_G}\right)^{b_q}}\right) & \text{if } r \leq 0.5 \\ (L_j - y_j(G))\left(1 - r(G)^{\left(1-\frac{G}{N_G}\right)^{b_q}}\right) & \text{if } r > 0.5 \end{cases} \tag{2.11}$$

where L_j and U_j are the lower and upper bounds of variable x_j, respectively,

$r(G) \in [0, 1]$ is a random number, and G and N_G are the current generation number and maximum number of generations, respectively. $'b'_q$ values are used to control the speed of the step length [43].

2.2.3 Evolution strategy (ES)

ES is a type of EA, in which operators of variation and selection are applied to a population of solutions to solve an optimization problem [92]. It has been found to be superior to many other common optimization method [92–96] and also has a strong theoretical background [95, 97, 98]. Of the different variants of ES introduced over the years, such as (1+1)-ES, (μ +l)-ES, (μ , l)-ES, (μ, λ)-ES, CMA-ES [99]. CMA-ES has shown its capability to efficiently solve diverse types of optimization problems [99] and its superiority to other ES algorithms [100, 101]. It uses the following steps [99].

Algorithm 1 CMA-ES algorithm

1: Generate an initial individual $\overrightarrow{x}_m$ and calculate its fitness function value;
2: Sample new solutions, such that $\overrightarrow{x}_{z,G+1} = \overrightarrow{x}_m + \sigma N(0,\ C_G)\, \forall z = 1 : NP$;
3: Evaluate and sort the new offspring.
4: Select μ individuals as a parental vector. Then calculate their center using $\overrightarrow{x}_{m,G+1} = \sum_{k=1}^{\mu} w_k \overrightarrow{x}_{k,G}$ where $\sum_{k=1}^{\mu} w_k = 1$ and $w_1 \geq w_2 \geq ... \geq w_\mu$.
5: Update the evolution path p^s_{G+1} and p^σ_{G+1}.
6: Adapt the covariance matrix (C_{G+1}).
7: Update the global step size (σ_{G+1}).
8: Go to steps 2, and repeat theses steps until a stopping criterion is met.

2.3 Ensembles of EAs

As there is no one algorithm and/or operator capable of consistently solving a wide range of test problems, several methods that use different algorithms or search operators in a single framework, such as ensemble-based (a mix of operators) [75], hyper-heuristics (heuristics selecting from heuristics) [102], mutli-methods (using more than one optimization algorithm) [103, 104], multi-operators (using more than one search operator in a single optimization algorithm) [23] and heterogeneous (using different algorithms with different behaviors) [105] approaches, have been developed.

These approaches need a selection mechanism to choose or put more emphasis on the best-performing one during the evolutionary process. The following subsection briefly described the adaptive operator selection mechanism.

2.3.1 Adaptive operator selection (AOS)

As previously stated, no single operator and/or algorithm is capable of solving a wide range of optimization problems. Therefore, researchers and practitioners have used frameworks that contain more than one search operator or algorithm in a single algorithmic framework, called multi-operator or multi-methods. The selection procedures of these algorithms, which are usually called AOSs, rely on different criteria, such as improvement in the feasibility rate and/or solution quality and/or constraint violations [24], convergence differences and progress ratios [106], a re-enforcement learning mechanism [107, 108] and pheromone updates of the Ant Colony Optimization Meta-Heuristic (ACO-MH) [109]. A comprehensive review of AOS is introduced by Maturana et al. [110], who divides it into its major components of credit assignment and operator selection. The former term defines how to reward an operator and/or algorithm based on its recent performance during a run and the latter uses these rewards to choose one operator and/or algorithm from a pool.

Credit assignment The feedback provided by operators and/or algorithms is used to calculate their rewards, which are maintained in a memory and are then used by the selection component to select the next operator and/or algorithm to be used by the framework.

Operator selection Using the reward values, operator selection methods choose operators and/or algorithms to produce offspring through two mechanisms: 1) probability-based methods, such as Probability Matching PM [111] and Adaptive Pursuit (AP) [112]; and 2) bandit-based methods [113].

- **Probability matching and adaptive pursuit methods:** often use a roulette wheel-like process to select an operator and/or algorithm. Some AOS methods based on probabilities, such as those used in the ZEPDE [114], DMPSADE [66], JADE [51], and jDE [69] algorithms, can be found in the literature.
- **Bandit based methods:** are based on the Multi-Armed Bandit (MAB) paradigm [108] [115] , in which each operator is viewed as one arm of a MAB problem, with the rewards based mainly on the fitness improvement brought by the corresponding operator to the individual to which it is applied. In the literature, there are a number of MAB methods, such as the Upper Confidence Bound (UCB) algorithm [107, 116], the Area Under the Curve (AUC), and the Sum-of-Ranks (SoR) [117]. More details can be found in [118].

Selection of operators and/or algorithms based on the characteristics of a problem or its features is rarely investigated in the literature. The fitness landscape analysis (FLA) is usually helpful to judge function complexity [25, 40]. The use of FLA in the selection of operators/algorithms, may boost the performance of the algorithm, if it is carefully incorporated. In the next section, an overview of the landscape measures is provided.

2.4 Landscape Analysis

Generally, a fitness landscape consists of a set of solutions (populations of individuals), each solution has a fitness function value (FV) and a neighborhood operator which may be a distance measure [25, 119]. Measuring the fitness landscape of a problem helps researchers to classify a problem as an easy or difficult one to solve [120].

2.4.1 Landscape measures

Many landscape measures for understanding and analyzing the different characteristics of optimization problems have been proposed [25, 27].They are classified in two groups [121], statistical- and information-based, which are briefly reviewed below.

2.4.1.1 Statistical-based measures

- **Auto-correlation** is a method that measures the degree of correlation between neighboring points on a landscape path and is often used to measure the ruggedness of a fitness function [26, 27]. Given a series of points generated by a random walk method, the auto-correlation value can be calculated by:

$$\rho(a) = \frac{1}{\sigma^2(f) \times (k-a)} \times \sum_{t=1}^{k-a} (f(x_t) - \bar{f}) \times (f(x_{t+a}) - \bar{f}) \tag{2.12}$$

where a is the step size between two points, k the number of random walk points, σ^2 the variance of the objective values and $\bar{f}$ the mean fitness. If the auto-correlation value is low, the landscape is considered rugged. There is also another measure called correlation length [26], which is computed as:

$$l = \frac{1}{ln(|\rho(1)|)} \quad \text{for } \rho(1) > 1 \tag{2.13}$$

where $\rho(1)$ is the auto-correlation between two points when $a = 1$. Smaller values of l indicate more rugged search landscapes and larger ones more smoother landscape.

- The **Fitness Distance Correlation (FDC)**, which was proposed by Jones and Forrest [28], is another landscape method used to realize a problem's difficulty [122]. It measures the correlation between the objective FV and the distance to the nearest optimal solution in the search domain. Given a number of individuals, $X = \{\overrightarrow{x}_1, \overrightarrow{x}_2, ..., \overrightarrow{x}_{NP}\}$ and their objective FV, $F = \{f_1, f_2, ..., f_{NP}\}$, the FDC value is computed by:

$$FDC = \frac{\sum_{i=1}^{NP}(f_i - \bar{f}) \times (d_i - \bar{d})}{n \times \sigma_f \times \sigma_d} \tag{2.14}$$

where d_i is the distance between the i^{th} candidate and best one in the population, and σ_f, σ_d, $\bar{f}$, and $\bar{d}$, are the standard deviation and average of the fitness function and distance, respectively. In case of minimization, values of FDC near 1 and 0 indicates that the search is easy and difficult, respectively.

- The **searchability measure** assesses the searchability of a problem, which is the capability of an operator/algorithm to move to an area of the search region with a better FV. It is computed by subtracting the problem information landscape vector from a reference one, which is the landscape of an easy function that is easy to be optimized by any algorithm, regardless of dimension [123].

An information matrix, $M = [a_{i,j}]$, for a minimization problem is formed using

$$a_{i,j} = \begin{cases} 1 & \text{if } f(x_i) < f(x_j) \\ 0.5 & \text{if } f(x_i) = f(x_j) \\ 0 & \text{otherwise} \end{cases} \tag{2.15}$$

Not all the entries in the information landscape are necessary for defining the information landscape [123, 124]. As due to matrix symmetry, the lower triangle can be omitted, the values on the diagonal are always 0.5, and the row and column of the optimal solution are also unnecessary. Therefore, an information matrix can be decreased to a vector, $Ls = (ls1, ls_2, ..., ls_{|Ls|})$, where the number of values is $|LS| = \frac{(NP-1)\times(NP-2)}{2}$.

Given two landscapes, Ls_f and Ls_{ref}, the difference value between them is computed by:

$$LD = \frac{1}{|Ls_f|} \times \sum_{i=1}^{|Ls_f|} |ls_{1i} - ls_{2i}| \tag{2.16}$$

When LD is close to 1, the problem is assumed to be difficult and when it is close to 0, easy.

- The **Dispersion Metric (DM)** was introduced by Lunacek and Whitley [20] to predict the existence of funnels, which are global basin shapes consisting of grouped local solutions in a landscape [29]. It measures the mean distance between two high-quality solutions, and is calculated by:

$$DIS_{\phi\%} = \frac{1}{(\phi \times n) \times (\phi \times n - 1)} \times \sum_{i=1}^{\phi\times n-1} \sum_{j=i+1}^{\phi\times n} ||x_i - x_j|| \tag{2.17}$$

where ϕ represents the top $\phi\%$ of the population. Given a sample of solution points, DM is calculated as the difference between the dispersion of all individuals in the sample and that of the set of best solutions. Negative and positive values of DM indicate the absence and presence of multiple funnels, respectively.

- The **Length Scale (LS)** is a landscape measure that can be used to reflect the ruggedness, smoothness and neutrality of a problem [30, 125]. It is calculated as:

$$LS = \frac{|f_1 - f_2|}{||x_1 - x_2||} \tag{2.18}$$

For more information regarding the fitness landscape analysis measures, readers are referred to [25, 40]

2.4.1.2 Information-based measure

To determine the modality of a function, information characteristics analysis (ICA) may be used. It was proposed by Vassilev et al. [31] to determine the characteristics of the fitness landscapes of discrete problems and was later adapted by Steer et al. [126] and Malan and Engelbrecht [127] to characterize those of continuous ones. Given a sample of individuals generated by a random walk, $X = \{\overrightarrow{x}_1, \overrightarrow{x}_2, ..., \overrightarrow{x}_n\}$, and their FVs, $F = \{f_1, f_2, ..., f_n\}$, the difference between consecutive FVs is used to produce a string, $\prod_k(\varepsilon) = \pi_1,\ \pi_2,\ ...,\ \pi_n$, where $\pi_k \in \{-1,\ 0,\ 1\}$. The value of π_k is computed by:

$$\pi_k(\varepsilon) = \begin{cases} -1 & \text{if } f_k - f_{k-1} \ < -\varepsilon \\ 0 & \text{if } |f_k - f_{k-1}| \leq \varepsilon \\ 1 & \text{if } f_k - f_{k-1} \ > \varepsilon \end{cases} \tag{2.19}$$

where ε is a sensitivity parameter. The entropic measure, $H(\varepsilon)$, of the string, $\pi_k(\varepsilon)$, is calculated by:

$$H(\varepsilon) = \sum_{d \neq e} P_{[de]} log_6 P_{[de]} \tag{2.20}$$

where d and d are elements in the string $\{-1, 0, 1\}$ and $P_{[de]}$ is the probability of the occurrence of the sub-blocks, de. The result is a value in the range of [0, 1], which is an estimate of a problem's ruggedness. The Partial Information Content (PIC), a measure proposed by Vassilev et al. [31] to measure a function's modality, is calculated by producing a new string, $\prod_k^{/}(\varepsilon)$, from the old one, $\prod_k(\varepsilon)$ by removing all zeroes and repeated digits, $\prod_k^{/}(\varepsilon) = \pi_1$, π_2, ..., π_μ, where $\pi_k \neq 0$ and $\pi_k \neq \pi_{k-1}$. It is computed by:

$$PIC(\varepsilon) = \frac{\mu}{n} \tag{2.21}$$

where μ is the length of $\prod_k^{/}(\varepsilon)$. PIC determines changes in the slopes in the random walk, which is an indication of local optima, with the number of local optima computed by:

$$N \approx \frac{PIC(\varepsilon) \times n}{2} \tag{2.22}$$

2.5 Constraint Handling Techniques with EAs

In this section, a number of constraint-handling techniques (CHTs) are discussed.

When solving constrained optimization problems with EAs, solutions that satisfy all the constraints are called feasible and those that fail to satisfy at least one of them are called infeasible. As dealing with infeasible solutions in COPs is a big issue, researchers have proposed a number of CHTs for coping with their constraints. These have been incorporated with EAs which have a good history of solving unconstrained optimization problems. One simple way of dealing with infeasible solutions is to completely reject them and continue the search with only feasible ones. However, this has the large drawback that some of the information contained in an infeasible solution may be lost [128]. Mezura-Montes and Coello Coello classify CHTs into two groups [128], early and current CHTs. The former group consists of the following techniques: 1) Penalty methods; 2) decoders; 3) special operators; and 4) separation of objective function and constraints. While the later group includes the following methods: 1) feasibility rules; 2) ϵ-constraint method; 3) stochastic ranking (SR); 4) novel special operators; 5) novel penalty methods; 6) methods using MOP techniques; and 7) hybrid methods. The following section discusses the most popular CHTs.

2.5.1 Penalty methods (PMs)

PM is the most common technique used by evolutionary computation (EC) researchers to handle constraints [92]. It is considered the simplest way of adapting an EA to Constrained Evolutionary Optimization (CEO) and can be performed without needing to re-implement the three basic evolutionary procedures (i.e., selection, crossover, mutation). It does this by penalizing infeasible solutions by summing a penalty to their objective function. Generally, penalty functions can be formulated as:

$$f(\overrightarrow{x}) = \begin{cases} f(\overrightarrow{x}) & \text{if all constraints are feasible} \\ f(\overrightarrow{x}) = f(\overrightarrow{x}) \pm [\sum_{k=1}^{s} \varrho_k \times G_k + \sum_{e=1}^{q} \theta_e \times H_e] & \text{otherwise} \end{cases} \tag{2.23}$$

where G_k and H_e are functions of constraints $g_k(\overrightarrow{x})$ and $h_e(\overrightarrow{x})$, respectively, and ϱ_k and θ_e penalty factors. In the literature, there are different types of penalty methods [129], such as death penalty [128], static penalty [130]. Some of the recently methods are discussed in the following subsections.

Dynamic Penalty: this technique assumes that the penalty factors depend on the number of generations and vary from generation to generation [131]. It needs fewer parameters [128, 131]. However, in practice, it is difficult to develop good dynamic penalty functions for static functions.

Adaptive Penalty: in it, the values of the penalty parameters are determined, based on information gathered from the search process [92]. Therefore, the value of each penalty parameter is adapted every generation and varies according to the quality of the solutions in the population. These functions are easy to implement and do not require any interaction from the user to define their values [130]. For some methods proposed in this category, a reader can refer to [128].

2.5.2 Separation of objective function and constraints

In this approach, the search space is divided into two steps in which constraints and objective function are handled separately. The aim of the first step is to find feasible solutions regardless of their objective FV, while the second is to optimize objective FV.

- **Superiority of feasible solutions approach [132]**- This method is based on the following heuristic: "every feasible solution is better than any other infeasible solution" [132]. Its main purpose is to provide feasible solutions with an advantage in the selection process by using the following rules when two solutions are compared: i) every feasible solution is better than every infeasible one; ii) of two feasible solutions, the one with the best fitness is preferred; and iii) of two infeasible solutions, the one with the smallest total constraint violations is better. In this approach, the selection mechanism performs only pair-wise comparisons. Thus, no penalty factor is needed. However, this method may lead to loss of diversity [131].

- **Multi-objective optimization approach**- In this category, COPs are solved as MOPs, whereby their constraints and objective functions are optimized using a multi-objective concept [133–139]. Although this is attractive to many researchers, it has the drawback that, in many cases, solving a COP in this way is more difficult than solving it as an essentially a single-objective optimization problem.

2.5.3 ϵ-Constraint method

This method converts a COP into unconstrained one [140]. In its process, it is used to compare two solutions, when the constraints violations of them are smaller than ϵ value, the fittest one is selected. Otherwise, the solution with small constraint violation is preferred. Several methods have been proposed to control the ϵ-level [140–143].

2.5.4 Stochastic ranking (SR)

This method [144] was proposed to strike the right balance between objective and penalty functions stochastically, which is very difficult to achieve in a penalty function technique. In it, a probability, P_f, was proposed either to select infeasible solutions based on either only their objective functions, or their sum of constraint violations. Although SR is simple, it cannot guarantee diversity as its ranking is stochastic.

2.5.5 Specialized operators

These methods help an EA to evolve solutions in the feasible region by either including special techniques to search for feasible solutions [44]; i.e., using special operators to transform infeasible solutions into feasible ones, or searching the boundaries between feasible and feasible regions [35, 36, 145].

2.5.6 Hybrid methods

In this category, an ensemble of CHTs is used to handle constraints. In [146], four CHTs: feasibility rules, stochastic ranking, a self-adaptive penalty function and the ϵ-constrained method, which are used in an ensemble way to solve COPs. Elsayed et al. [147] proposed a DE algorithm, which uses four mutations, two crossover and two CHTs (Superiority of Feasible Solutions and ϵ-constrained method).

2.6 EAs Review

In this section, the literature on solving unconstrained and constrained optimization problems with EAs as well as, the landscape analysis techniques used with EAs, is reviewed.

2.6.1 EAs for unconstrained optimization problems

In this section, EAs used to solve unconstrained optimization problems were briefly reviewed.

2.6.1.1 Single algorithms for unconstrained problems

Zhang et al. [64] proposed an ADE algorithm with an optional external memory (JADE), in which the Cr_i and F_z of each individual $\vec{x_z}$ at each generation was independently generated according to a normal and Cauchy distribution, respectively. A repair mechanism is used to handle the value falling outside range as: when the value of Cr_z fell outside the range of [0,1], it was repaired to a value within it. If the value of F_i was greater than 1, it was truncated to 1 or re-generated if $F_z < 0$. Tanabe and Fukunage [148] proposed a success-history-based parameter adaptation for DE (SHADE) in which instead of using a single pair (μ_{CR}, μ_F) to guide parameter adaptation as in JADE, the mean values of S_{CR} and S_F for each generation were stored in memory as μ_{CR} and μ_F, respectively. Later, Tanabe and Fukunage [67] improved SHADE algorithm by using a linear population size reduction (LPSR) to dynamically re-size the population NP during a run as the number of fitness evaluations increases, which is called L-SHADE. It showed good performance in comparison with other algorithms for solving a set of unconstrained optimization problems. Later many improved versions of L-SHADE have been proposed [149–153].

2.6.1.2 Ensembles of operators for unconstrained problems

In this subsection, a brief review of different algorithms that use ensembles of different operators, as well as the selection mechanisms, is provided.

A new version of DE, $jDE_{NP,MM}$, for solving optimization problems, where

MM stands for multiple mutation strategies because two are incorporated, was proposed by Zamuda and Brest [154]. It is also uses a methodology for reducing its population size. Each new solution is generated by one of the two mutation strategies by using the following selection mechanism. The first mutation strategy is used when $NP \geq 100$ or 2) when $rand < 0.75$, and $rand$ is a uniform random number $\in [0, 1]$, otherwise the second was used. $jDE_{NP,MM}$ was tested on 22 real world applications and demonstrated better performances than two other algorithms.

Wang et al. [68] developed a composite DE (CoDE), in which three mutation strategies are randomly combined with three fixed control parameter settings to generate a new trial vector. At each generation, three vectors are generated for each target vector, with the one with the best objective FV surviving to the next generation. It was inferred from their experimental results, that CoDE obtained better results when compared with state-of-the-art algorithms.

A self-adaptive DE (SaDE) for solving unconstrained real-parameter optimization problems was proposed by Qin et al. [155]. In it, both the strategies for generating the trial vector and its associated control parameter values, are gradually self-adapted according to a success rate calculated based on previous learning experiences. Firstly, all the mutation strategies have equal probabilities of generating a new solution. However the probabilities are updated after the initial LP generations as follows: the updated probability for each mutation strategy is the number of successful individuals divided by the number of all the individuals generated by that strategy in the previous LP. This algorithm performed much better than both traditional DE and several state-of-the-art DE variants with adaptive parameters.

Mallipeddi et al. [156] proposed an ensemble of mutation strategies and control parameters in DE (EPSDE) for solving unconstrained optimization problems. In it, each solution in the initial population is randomly assigned a mutation strategy with its associated parameter values taken from pools of different mutation strategies and the values of each control parameter. The process for selecting the next generation of parent individuals is as follows: if the generated solution 1) is fittest than its parent in the previous generation, its mutation strategy and parameter values are stored; or 2) is worse than its parent, the parent vector is reinitialized with a new mutation strategy and associated parameter values. The performance of EPSDE compared favorably with a set of well-known DE methods.

A multi-population-based framework in which three mutation strategies are combined in a novel DE variant called MPEDE, was proposed by Wu et al. [22]. Its initial population is dynamically divided into several sub-populations, three small indicator ones of equal sizes and one larger reward one. Each mutation strategy is used to evolve one indicator sub-population, with the reward sub-population assigned to the current best-performing mutation strategy. That selection based on the ratios between the fitness improvements and consumed function evaluations. MPEDE was tested on the suite of CEC 2005 benchmark functions and was shown to be competitive with other state-of-the-art algorithms. For more information regarding multi-operator algorithms, readers are referred to [17].

A new adaptive multi-population DE with dynamic population reduction, sTDE-dR, in which the initial population is divided into multiple tribes, was proposed by Ali et al. [76]. A distinct mutation strategy and crossover operator, selected from pools of different ones is used to evolve one sub-population, and a different adaptive method is used to determine the values of F and Cr for each tribe. The mean success of each tribe in each generation is used to calculate the ratio of its participation in the next generation. The proposed algorithm was validated by solving a challenging set of benchmarks from the CEC2014 real-parameter single-objective competition, and when compared with several state-of-the-art algorithms, showed superior performance.

Ali et al. [21] introduced a multi-population DE (mDE-bES) for solving large-scale optimization problems, in which the initial population is divided into sub-populations of equal sizes, which are evolved using different DE mutation strategies for a specified number of generations. An information-sharing method, in which some individuals are randomly selected, based on a neighborhood structure to migrate between sub-populations, is also used. mDE-bES was tested on a set of 19 large-scale continuous optimization problems. Its results showed that its performance and scalability behavior were competitive with those of state-of-the-art algorithms.

2.6.1.3 Ensembles of algorithms for unconstrained problems

Over the last few years, the concept of multi-method algorithms, which uses the strengths of different EAs, has emerged. Vrugt and Robinson [157] introduced the multi-algorithm genetically adaptive multi-objective (AMALGAM) algorithm that has demonstrated to be a robust algorithm for solving MOPs. Later, this work was extended [157] by proposing an algorithm known as AMALGAM for single-objective optimization (AMALGAM-SO) which uses the strengths of CMA-ES, a GA and a particle swarm optimizer (PSO) in a single algorithm for evolving a population. The number of offspring for each algorithm are dynamically updated by a self-adaptive learning strategy. When experimented on a number of unconstrained optimization problems, it similarly performed to existing algorithms for small dimension problems but was superior to the existing ones for higher-dimensional multi-modal and more complex optimization problems.

Peng et al. [158] proposed a new approach, called the population-based algorithm portfolio (PAP), which exploits the advantages of four existing population-based algorithms (the self-adaptive DE with neighborhood search (SaNSDE), PSO with inertia weight (wPSO), generalized generation gap (G3) model with a PCX operator (G3PCX) and CMA-ES). Each single algorithm is run for a portion of the specified resources (time and cost) and an information-sharing scheme is used to encourage interaction among them. PAP was comprehensively evaluated by investigating 11 instances of it applied on 27 benchmark functions, with the results showing that it outperforms its constituent algorithms.

Olorunda and Engelbrecht[159] proposed a heterogeneous cooperative algorithm which uses the strengths of a GA, DE and PSO, to evolve each subpopulation in a single algorithmic framework. It was shown to perform consistently and competitively on all the considered problems. Grobler et al. [160], introduced six methods for controlling heuristic space diversity (HSD) during the optimization process. In their experiments, a heterogeneous meta-hyper-heuristic (HMHH) method with four common meta-heuristic algorithms as its set of constituent algorithms (a GA, guaranteed convergence PSO (GCPSO), SaNSDE and CMA-ES) is used as a basis for investigating the management of HSD. Their proposed algorithm showed good performance when compared with a popular PAP one.

Li et al. [161] proposed a new multi-method algorithm which combined the advantages of both modified CoDE (MCoDE) and modified JADE (MJADE) in a single evolutionary framework which they called HMJCDE. In it, an adaptive mechanism, based on the rate of improvement in objective FVs, is employed to determine which algorithm to use to evolve the entire population. The performance of HMJCDE was assessed by solving well-known unconstrained problems and was shown to be better than those of JADE, CoDE and other state-of-the-art algorithms.

Zhao et al. [162] introduced an algorithm which combines the search capabilities of both the SaDE algorithm and a modified multiple trajectory search (MMTS), called SaDE-MMTS, for solving large-scale optimization problems. As previously explained, in SaDE, the DE mutation strategies used and their associated control parameter values, are self-adapted by learning from their successes, while MMTS adopts a clearing procedure to select its search agents while using self-adaptive step sizes. Switching between these algorithms is implemented, based on their success rates in previous generations.

Elsayed et al. [104] proposed a united multi-operator EAs (UMOEAs) approach for solving unconstrained optimization problems. The algorithm starts with an initial population that is then divided into sub-populations, each of which is evolved using a different multi-operator EA, with the best-performing one determined based on its success rate. Later, Elsayed et al. [163] proposed an improved version of UMOEAs, called UMOEAsII, in which each multi-operator EA runs with multiple search operators. In it, an adaptive operator selection (AOS) mechanism, based on the quality of solutions produced and the diversity of the population, places emphasis on the better-performing multi-operator algorithm and its search operators. UMOEAsII was tested on the CEC2016 competition's single-objective real-parameter optimization problems, which demonstrating its capability to obtain better results than those of other state-of-the-art algorithms. In doing so it won the competition.

2.6.1.4 EAs utilizing landscape techniques for unconstrained problems

Recently, researchers have utilized fitness landscapes to decide and choose appropriate algorithms and/or operators for solving optimization problems.

Merz and Freisleben [164] performed a fitness landscape analysis of a number of Quadratic Assignment Problems (QAP), with the results used to classify problem instances according to the difficulty of their local searches. An autocorrelation analysis is used to examine the suitability of a local search and the global structure is investigated by employing a FDC analysis. Bischl et al. [165] suggested using a model to select the best-performing algorithm from a set of four for solving Black-Box optimization Benchmarking (BBOB) problems [166, 167], based on Exploratory Landscape Analysis (ELA) techniques. Then, the modality, separability, and global structure of an optimization problem are determined as the first step in identifying the landscape (performed off-line). Next, to select the best-performing algorithm a machine-learning model is constructed and validated based on two different cross-validation schemes. Nevertheless, the results may not be generalizable for problems with different dimensions. Because the low-level features are obtained in a separate step, the authors did not add the computational cost for calculating them to the number of function evaluations. Also, as the selection of the algorithm pool is manual, its validation on unobserved problems is weakened.

Garden and Engelbrecht [168] employed a self-organizing feature map to cluster and analyze the landscapes of two commonly used benchmark problems. They concluded that while there are functions that represent a wide range of properties, some other properties are not fully covered in these benchmarks. In [169], a prediction model for predicting when a PSO algorithm will fail to solve a particular optimization problem is developed, with decision trees used to predict the failures of seven different PSOs, by utilizing a number of fitness landscape measures.

Munoz et al.[170] used a regression model (multi-layer feed-forward neural network) to determine the best of 8 parameter combinations for the CMA-ES algorithm. A knowledge base of 1800 problem instance drawn from the Comparing Continuous Optimization (COCO) benchmark [167], were used to train the model. Data from the CEC2005 competition [171] was then used to validate the model and the seven ELA measures that were used to analyze and characterize each problem. However, all the landscape analyzes were performed off-line and the sample size of $15 \times 10^3 \times D$ that was used to calculate the ELA measurement is too expensive to be practical. Also, during the validation phase, the accuracy of the model was compared with only random configurations of unseen problems. Munoz et al. [172] proposed a new version of information content of fitness sequences (ICoFiS), in which a Latin hypercube design (LHD) was used to generate initial samples and also the distance between two solutions in the analysis was included. COCO was used as the testing functions and the lower and upper bounds for the sample size were set at values of $10^2 \times D$ and $10^3 \times D$, respectively, which is still computationally expensive in terms of the number of fitness evaluations.

Sun et al. [173] proposed a novel exploratory landscape analysis (ELA) method, called Maximum Entropic Epistasis (MEE), to measure the level of the interactions between variables in continuous domain. It determines interactions between two solutions $\vec{x}_1$ and $\vec{x}_2$, by calculating the maximal information coefficient (MIC) between $\vec{x}_2$ and the partial derivative of the fitness function with respect to $\vec{x}_1$. Interestingly, MEE was investigated to examine its possibility to guide the evolutionary search process for an optimal solution to a specified problem. It showed equal or better performance than the algorithm designed in [174]. However, as

MEE uses the partial derivative of the fitness function, it may be computationally expensive.

As the area covered by the points generated by the random walk-based methods is limited and not enough for landscape analysis, Jana et al. [175] proposed a chaos-based random walk algorithm to quantify the complexity and deception of a problem. The accuracy of determining the ruggedness of a problem was compared to those obtained from simple and progressive random walk-based algorithms, with the results were slightly better.

2.6.2 EAs for constrained optimization problems

As previously mentioned, due to the success of EAs in solving unconstrained optimization problems and the importance of solving COPs in many scientific area, researchers have used CHTs with EAs to solve COPs. Some recent significant advances in this direction are discussed in the following subsections.

2.6.2.1 Single algorithm for solving COPs

Takahama and Sakai [176] proposed εDE, which incorporates the the ε constrained method with DE. In it, a feasible-elite preserving strategy and gradient-based mutation were used. Later, to handle COPs with many equality constraints, an improved εDE [141] was proposed. It adopted faster reduction of the relaxation of the equality constraints and a higher gradient-based mutation rate. Takahama and Sakai [140] extended their previous work [141, 176], by introducing an archive which stored solutions and maintained population diversity. Their algorithm is called ϵDEag and it achieved excellent performances, in cases in which many infeasible solutions needed to be repaired. However, it consumed a great deal of time.

Wang and Cai [177] developed a DE algorithm, that is combined with multi-objective optimization to solve COPs, called CMODE. It was tested on CEC2006 benchmark problems, with very competitive performance to state-of-art algorithms. However, when the offspring is feasible and parents are infeasible, the idea of dominance might not be effective. Ning Dong et al. [178] transformed COP into a bi-objective optimization problem by using a novel unbiased bi-objective optimization model, in which Pareto ranking was used as a selection method.

An adaptive ranking mutation operator (ARMOR), based on DE, was introduced by Gong et al. [179]. In it the individuals were adaptively ranked based on three situations: 1) infeasible, the constraint violations were used to rank the solutions; 2) semi-feasible, the individuals were ordered based on the transformed objective function; and 3) feasible, the solutions were ranked based on objective function value.

Elsayed et al. [180], introduced a multi-parent crossover GA (GAMPC) for solving the real-application problems introduced in [181]. A randomized operator was applied as mutation, and also an archive of good solutions was used to maintain population diversity. It was the winner of the CEC2011 competition. Later, it was enhanced by using a diversity operator, instead of mutation, for quicker convergence [182].

2.6.2.2 Ensembles of operators for COPs

Mallipeddi and Suganthan [183] proposed an algorithm that uses four CHTs with DE (ECHT-DE) to solve COPs. In its internal process, four population were randomly initialized with each CHT was used with one sub-population with an information sharing method which shared offspring among them. It was tested by solving CEC2010 CEC2010 competition for COPs, and it came second in the rank. However, it was expensive in terms of computational time.

Da Silva et al. [184] used an adaptive penalty method with DE for solving a set of COPs. In it, a mechanism for automatically choosing one of four DE mutation strategies is introduced. This occurs as: 1) for the first 10% of the total number of generations allowed, four individuals are generated and the best chosen; 2) for the next 30% new individuals are generated by only the variant which produced the best individual in the current population; 3) all the variants are used again to generate new individuals for another 10%. 4) the best strategy is used to generate new solutions for the next 30%; and 5) a third evaluation stage is performed using all the operators, with the best-performing variant chosen for the remaining generations. Based on the results obtained from solving several structural and mechanical engineering optimization problems, it was claimed that the algorithm performs better than certain GA techniques used in the continuous domain.

Elsayed et al. [185] introduced an improved version on SAMODE [24], called ISAMODE-CMA, in which CMA-ES was periodically applied to enhance its local search capability, and a dynamic penalty is used to handle constraints. In their method, a mix of modified versions of binomial and exponential crossovers is adopted. A restart mechanism is used to escape local solution. Also, the sub-population sizes are updated based on an improvement index that considers both feasibility ratio and solution quality. ISAMODE-CMA was compared with different state-of-the-art approaches and showed superior performance. Elsayed et al. [147] developed a DE algorithm, called SAS-DE, in which four DE mutation, two crossover operators, and two CHTs, are used. The performance of SAS-DE was tested by solving a set of benchmark problems [186], with consistently better performance than other well-known algorithms.

Wang et al. [187] introduced a new algorithm, that uses two mutation strategies each of which was used with equal probability of 0.5. In it, an infeasible solution archiving and replacement mechanism are presented. For this, during an entire evolution process, if an offspring dominates its parent, the parent solution may be replaced. The experimental results showed that their proposed method was superior to state-of-the-art algorithms.

A DE algorithm (DE-AOPS), that uses a new technique for dynamically choosing the best combinations of parameters, from given continuous ranges, as well as DE operators, is proposed [188]. Also, a new CHT is proposed in which DE-AOPS starts with handling some constraints, with the largest sum of the constraint violations till a pre-defined number of generations, and gradually considers all of them. The performance of DE-AOPS is judged using a well-known set of COPs, with consistently better performance than other state-of-the-art algorithms.

2.6.2.3 Hybrid and Ensembles of Algorithms for COPs

Elsayed et al. [189] proposed an algorithm that utilizes four EAs (DE, GA, ES and EP) with each one has two different operators (EAwAC). It uses information sharing procedure to replace the worst four individuals in each sub-population with the best four individuals (the best one from each sub-population). The number of individuals in each sub-population as well as the number assigned to each operator are dynamically updated, based on both the objective FV and feasibility rate. The experimental results showed that EAwAC was competitive to, if not better than, well-known algorithms.

Zaman et al. [190] developed a new multi-method EA which adaptively places more emphasis on the most suitable EA, based on each algorithm's performance during previous generations. It uses the strengths of both a GA and DE (GA-DE) to solve dynamic economic dispatch (DED) problems. To enhance its convergence rate, a repair technique is used to move infeasible solutions towards feasible ones. Its effectiveness was tested by solving a number of dynamic economic dispatch (DED) problems. In this it was superior to state-of-the-art algorithms, in terms of its quality of solutions and reliability.

2.6.2.4 EAs utilizing landscape techniques for COPs

As previously stated, FLA techniques can be used to measure the relative complexity of a problem. However, many studies using FLA measures have been limited to combinatorial problems, with a few studies done on constrained optimization problems [25, 164, 191].

Tayarani and Prugel-Bennett [192] used auto-correlation measure to study the ruggedness of the quadratic assignment problem (QAP). In their study, they showed how the similarity and difference between real-world problems and the random problems. In [193], an AOS mechanism, based on a set of four fitness landscape analysis techniques, is used to train an on-line regression learning model (dynamic weighted majority) to predict the weight of each operator in each generation. It determines the most suitable of four crossover operators for solving a set of capacitated arc-routing problem (CARP) instances. It uses an instantaneous reward, which is considered the value computed in the last evaluation. However, compared with some well-known algorithms, this algorithm did not show significant benefits.

Poursoltan and Neumann [194] build a model to predict the best-performing algorithm from (DE, ES and PSO) for a specific COP. Its inputs are the features of a given COP instance, these features are constraint coefficients relationships such as number of constraints, feasibility ratio in vicinity of optimum, angle between constraint hyperplanes and standard deviation. The experimental results show that prediction model is able to give a good prediction. However, as a training and testing mechanism is used, the algorithm may be limited to the test problems considered and its performance deteriorate when solving another set of problems.

2.7 Boundary-based Methods and Search Space Reduction

In the literature, a number of evolutionary approaches have been developed, that consider the fact that for most practical problems the optimal solution exists on the boundary of feasible space. Michalwicz [37] was the first to propose a GA, known as Numerical Optimization for Constrained Optimization (GENOCOP), that used three mutation strategies (uniform, boundary and non-uniform), where the boundary operator generates random points on the boundary of the feasible region, and it also incorporated three different crossover operators (arithmetical, simple and heuristic). In [195], for searching the edge of feasibility, a geometrical crossover was presented. It is a complicated crossover mechanism which transforms an infeasible solution to become feasible. This crossover is suitable to solve COPs with linear equality constraints. However, it has some drawbacks, including that: 1) it may fail in solving COP with nonlinear constraints; and 2) the design of an efficient geometrical crossover technique is dependent on the problem.

To continue the research direction in [195], different sphere-based operators were designed to restrict the search to the boundary of the feasible region [145]. These operators were used with different optimization algorithms, such as GA and ES, with the sphere function adopted in the design of search operators because of its simplicity and attractive symmetrical proprieties. However, these operators only work on certain types of problems, and it would be difficult to design a specific operator to solve each individual problem. In [196], a self-adaptive boundary search GA was proposed and applied to solve a water distribution problem. The approach is able to reduce the computational effort required to in reach high quality solutions, in comparison to traditional GA.

Based on the boundary search concept introduced in [37], a new ant colony optimization (ACO) algorithm was proposed by Leguizamon and Coello Coello [36] for solving nonlinear COPs. It is well-known that any line segment that connects a feasible point with an infeasible point will intersect the boundary of the feasible region at some point. A binary search method, such as bisection, can be used to search the line segment to locate points on the boundary of feasibility space. This assists the convergence of the proposed ACO algorithm. Although this algorithm shows superior performance, it has some limitations, such as 1) it requires both feasible and infeasible points to conduct the search, and this could be problematic for many problems at the beginning of the evolutionary process; and 2) it could be computationally expensive because of the bisection process.

According to [35], it needs at least one active constraint to start the search process and the edges of the feasible space is represented by a new function that determines the thicknesses of the edges around the active constraints. The thickness enables the search process to focus on the boundary of the feasible region. However, the main drawback of this approach, is that it assumes that the active and inactive constraints are known. In another study, a GA that utilizes infeasible solutions that are near to a constraint boundary was proposed by Singh et al. [89]. As when these solutions are used with the feasible solutions, new solutions may be generated either on, or near to, the constraint boundary. However, this can not be done unless some feasible solutions are present.

In [41], a search space reduction technique (SSRT) is proposed, as an initial step of GA for solving COPs. Its main idea, is to direct some of the randomly selected infeasible solutions in the initial population, to move towards the feasible space. Due to the fact, that the optimal solution of COPs must lie on each and every equality constraint, Barkat Ullah et al. [38] proposed a heuristic technique to emphasis the search process on the landscape of equality constraints. From the current location of an individual solution, this heuristic is able to reach a solution on the equality constraint, and then search the constraint landscape. However, it may encounter problems when there are many variables, especially when many of them are nonlinear; this is because it needs to apply numerical search methods to find out the values of nonlinear variables [197]. Also, it assumes that the equality constraint function must contain all variables. If this is not the case, it needs to explore the landscape of other constraints.

2.8 Chapter Summary and Research Directions

Solving optimization problems is deemed a vital research area that assists decision-making processes in several practical applications. This chapter presented an introduction to unconstrained and constrained optimization problems and how EAs are used to solve them. Also, different constraint handling techniques was presented. The landscape analysis measures and search space reduction along with current well-known methods which uses FLA methods and search space reduction with EAs were discussed.

Of the many solution approaches developed, EAs have demonstrated great success in solving both unconstrained and constrained problems with a wide variety of properties. However, their performances were highly dependent on the algorithms design, selection of its operators and parameter settings. Also, no single algorithm and/or search operator performed consistently well for all the types of optimization problems defined in the literature. Although, ensemble-based methods have shown success in tackling these shortcomings, finding an appropriate design of these approaches still challenging, for example, it is difficult to determine (a) which algorithm and/or operators should be used in the pool, (b) which algorithm and/or operator should be chosen during the evolutionary process, and based on which selection criteria. Moreover, existing selection approaches do not utilize any information about function's/ problem's landscape. This motivates us to develop algorithms that utilizes problem's landscape information to select the best-performing operators and/or algorithms, as will be discussed in Chapters 3, 4 and 6.

Also, in a COP, the feasible space, which is bounded by the constraint functions, may represent a relatively small portion of the search space, and the existence of the equality constrains make this feasible space tiny. Also in many real-world application, the optimal solution lies in the boundary of the feasible region, which is the intersection of the active constraints. Although EAs have been successfully used to solve COPs, they still suffer from the drawback of wasting a considerable amount of time (fitness evaluations) searching ineffective areas in the search space. So this motivates us to develop a mechanism that may efficiently be able to to find that intersection, or a nearer point, as will be discussed in Chapters 5 and 6.

Chapter 3

Landscape-based Algorithm for Unconstrained Problems

In this chapter, a new landscape-based DE algorithm is proposed for solving unconstrained optimization problems. After a brief introduction, the descriptions of the benchmark problems which are solved using the proposed algorithms, is described. Then the proposed algorithm and its main components are presented. Then, the effects of the proposed algorithm's components on its performance are discussed. Finally, the experimental results from using it to solve different sets of unconstrained optimization problems are analyzed and then compared with those from state-of-the-art algorithms.

3.1 Introduction

As previously reported, optimization problems can be divided into many different classes, based on their characteristics and mathematical properties. Therefore,

Part of this work has previously appeared in
1. K. M. Sallam, S. M. Elsayed, R. A. Sarker, and D. L. Essam, "Landscape-based adaptive operator selection mechanism for differential evolution," Information Sciences, vol. 418, pp. 383-404, 2017.
2. K. M. Sallam, S. M. Elsayed, R. A. Sarker, and D. L. Essam, "Differential evolution with landscape-based operator selection for solving numerical optimization problems," in Intelligent and Evolutionary Systems: The 20th Asia Pacific Symposium, IES 2016, Canberra, Australia, November 2016, Proceedings. Springer, 2017, pp. 371-387.
3. K. M. Sallam, S. M. Elsayed, R. A. Sarker, and D. L. Essam, "Two-phase differential evolution framework for solving optimization problems," in Computational Intelligence (SSCI), 2016 IEEE Symposium Series on. IEEE, 2016, pp. 1-8.

an algorithm with certain operators and/or parameters and/or CHTs may work efficiently on one set of test problems, but poorly on another.

To tackle this problem, researchers have proposed many different methods that use different algorithms or search operators in a single framework, such as ensemble-based (a mix of operators) [75], hyper-heuristics (heuristics selecting from heuristics) [102], mutli-methods (using more than one optimization algorithm) [103, 104], multi-operators (using more than one search operator in a single optimization algorithm) [23] and heterogeneous (using different algorithms with different behaviors) [105] approaches. To clarify, each of these algorithms uses a pool of different algorithms/operators and a selection mechanism to determine the best-performing one/s during the search process. Their selection procedures are based on different criteria, such as a re-enforcement learning mechanism [107, 108], improvement in the solution quality and/or constraint violations and/or feasibility rate [24], and convergence differences and progress ratios [106]. However, none of these approaches considers any function characteristics in the process of selecting the best-performing operator and/or algorithm.

Landscape information is usually helpful to judge the function complexity. The carefully incorporating it in the selection of operators may improve the performance of an algorithm. However, selection methods based on a landscape analysis are very rare [165, 193] and have some limitations, such as: (1) a landscape analysis was performed using an off-line mode, i.e., with the initial experiments independently conducted, to calculate the values of the landscape statistics obtained from the evolutionary process in order to solve a problem; (2) calculating landscape measures is computationally expensive; and (3) as the training and testing mechanism used may restrict an/the algorithm to the test problems considered, its performance can deteriorate when solving another set of problems.

The aim in this chapter is to propose a new algorithm, that is capable of solving a wide range of unconstrained optimization problems. To achieve this objective, a nulti-operator DE algorithm is proposed. Its novel design utilizes landscape information with an adaptive operator selection (AOS) mechanism, which is used to give more importance to the best-performing operator during the run.

The performance of the proposed approach is tested by solving 45 unconstrained optimization problems taken from CEC2014 [198] and CEC2015 [199]. Several experiments are designed and carried out to analyze the effects of different parameters and components on its performance, and the results from the final variant of the proposed algorithm compered with those from well-known algorithms in the literature.

This chapter is organized as follows: the next section presents the benchmark problems; the proposed algorithm and its components are described in Section 3.3, the experimental results and analysis are discussed in Section 3.4; and, finally, the chapter summary is presented in Section 3.5.

3.2 Benchmark Problems

To judge the performances of the proposed algorithms, a total of 45 widely used test functions (introduced in the CEC2014 and CEC2015 competitions) are employed.

The sets of the CEC2014 and CEC2015 benchmark problems [198, 199] contain 30 and 15 test functions, respectively, as detailed in Tables 3.1 and 3.2, respectively. The criterion for both these competitions was that each problem was independently run 51 times, with the stopping condition $10000 \times D$ fitness function evaluation (FEs), where D is the number of decision variables.

Table 3.1: Properties of the CEC2014 test problems

	No.	Functions	Optimal solution
Unimodal Functions	1	Rotated High Conditioned Elliptic Function	100
	2	Rotated Bent Cigar Function	200
	3	Rotated Discus Function	300
Simple Multimodal Functions	4	Shifted and Rotated Rosenbrock'sFunction	400
	5	Shifted and Rotated Ackley's Function	500
	6	Shifted and Rotated Weierstrass Function	600
	7	Shifted and Rotated Griewank's Function	700
	8	Shifted Rastrigin's Function	800
	9	Shifted and Rotated Rastrigin's Function	900
	10	Shifted Schwefel's Function	1000
	11	Shifted and Rotated Schwefel's Function	1100
	12	Shifted and Rotated Katsuura Function	1200
	13	Shifted and Rotated HappyCat Function	1300
	14	Shifted and Rotated HGBat Function	1400
	15	Shifted and Rotated ExpandedGriewank's plus Rosenbrock's Function	1500
	16	Shifted and Rotated Expanded Scaffer's F6 Function	1600
Hybrid Functions	17	Hybrid Function 1 (N=3)	1700
	18	Hybrid Function 2 (N=3)	1800
	19	Hybrid Function 3 (N=4)	1900
	20	Hybrid Function 4 (N=4)	2000
	21	Hybrid Function 5 (N=5)	2100
	22	Hybrid Function 6 (N=5)	2200
Composition Functions	23	Composition Function 1 (N=5)	2300
	24	Composition Function 2 (N=3)	2400
	25	Composition Function 3 (N=3)	2500
	26	Composition Function 4 (N=5)	2600
	27	Composition Function 5 (N=5)	2700
	28	Composition Function 6 (N=5)	2800
	29	Composition Function 7 (N=3)	2900
	30	Composition Function 8 (N=3)	3000
Search Range $[-100, 100]^D$			

3.3 Multi-operator DE Algorithm with Landscape-based AOS

In this section, a landscape-based AOS algorithm for DE (LSAOS-DE), is presented. It uses more than one DE mutation strategy in a single algorithmic framework. Also, it has an adaptive operator selection process, based on both

Table 3.2: Properties of the CEC2015 test problems

	No.	Functions	Optimal solution
Unimodal Functions	1	Rotated High Conditioned Elliptic Function	100
	2	Rotated CigarFunction	200
Simple Multimodal Functions	3	Shiftedand Rotated Ackley's Function	and
	4	Shifted and Rotated Rastrigin's Function	400
	5	Shifted and Rotated Schwefel's Function	500
Hybrid Functions	6	Hybrid Function 1 (N=3)	600
	7	Hybrid Function 2 (N=4)	700
	8	Hybrid Function 3(N=5)	800
Composition Functions	9	Composition Function 1 (N=3)	900
	10	Composition Function 2 (N=3)	1000
	11	Composition Function 3 (N=5)	1100
	12	Composition Function 4 (N=5)	1200
	13	Composition Function 5 (N=5)	1300
	14	Composition Function 6 (N=7)	1400
	15	Composition Function 7 (N=10)	1500
Search Range $[-100, 100]^D$			

the (1) problem landscape and (2) performances of operators, to choose the best-performing DE mutation strategy during the run.

3.3.1 LSAOS-DE

As previously discussed, existing multi-operator algorithms usually use an AOS mechanism because of its successful generation of new offspring. The basic steps in the proposed LSAOS-DE algorithm, which uses both the problem's landscape information and also operators' performances to adaptively choose the most suitable DE operator, are presented in Algorithm 2 and Figure 3.1.

To fully cover the search space, it is initiated using the Latin Hyper-cube Design (LHD) [200] as described in section 3.3.1.1 and a pool of m DE mutation strategies with diverse characteristics is defined. Each operator is assigned to the same number of individuals and a new solution generated according to its assigned mutation strategy. At the same time, the information landscape metric and performance history of each operator are calculated using Algorithm 3 and equation 3.4, respectively. This process continues for CS generations, after which the normalized average values of the landscape metric and performance history measure are computed for each operator. Based on these values, the best-performing operator is selected to evolve the entire population in the subsequent cycle (for CS generations). Once this step is completed, and owing to the fact that the performance of one operator may change during the evolutionary process, i.e., it may perform well in early generations but poorly in later ones or vice versa, all the m operators are re-used to evolve the entire population and the landscape measure is calculated and recorded again. Note that, after every CS generation, the success rate is set to a value of zero, as is the landscape metric. This process is continued until a pre-defined number of fitness evaluations (*limit*) is reached, and then the best-performing operator found so far is used to evolve the entire population until a stopping criterion is met.

At the same time, during the evolutionary process, NP is adaptively re-sized using linear population size reduction [67],it has a large value at the start of the evolutionary process which is then linearly reduced (by deleting the worst individual from the population) until NP^{min} is reaches, as:

$$NP_{t+1} = round[(\frac{NP^{min} - NP^{init}}{MAX_{FES}}) \times FES + NP^{init}] \tag{3.1}$$

where NP^{min} is the smallest number of individuals the proposed algorithm will use, FES and MAX_{FES} the current and maximum numbers of fitness evaluations. The purpose of adapting NP is to maintain diversity during the early stages of the evolutionary process while placing greater emphasis on the intensification process in later stages [67].

Algorithm 2 Proposed LSAOS-DE algorithm

$Counter \leftarrow 0$; Generate an initial population (X) of size NP using Latin Hypercube Design; $FES \leftarrow 0$;
Calculate the fitness values of X;
$FES \leftarrow FES + NP$;
while $FES \leq MAX_{FES}$ **do**
  $C \leftarrow C + 1$;
  if $FES \leq limit$ **then**
    $C \leftarrow C + 1$;
    if $C < CS$ **then**
      Randomly assign each operator to $([\frac{NP}{m}])$ solutions;
      Generate new population using the assigned operators.
      Calculate the LD_{op} considering those individuals updated by operator op;
      Calculate the success rate SR_{op} (Equation 3.4) considering those individuals updated by operator op;
    end if
    if $\text{mod}(C, CS == 0)$ **then**
      Measure the performance of every m using Equation 4.5;
      Determine the best-performing operator, i.e., the one with the maximum value of NPM;
    end if
    if $C > CS$ and $C < 2CS$ **then**
      Evolve the current entire population using best DE mutation strategy;
    end if
    if $C == 2CS$ **then**
      $Counter \leftarrow 0$, reset m, $SR_{op} = [\ ]$, $LD_{op} = [\ ]$;
    end if
  else
    Evolve the population using the best DE operator;
  end if
  $FES \leftarrow FES + NP$;
  Update NP using equation (Equation 3.1);
end while

It is worth mentioning that the proposed algorithm has the same structure as multi-method/multi-operator, heterogeneous and ensemble-based ones while its pool consists of several different DE mutation strategies and its selection mechanism relies on both the information landscape and the performance history of each DE strategy.

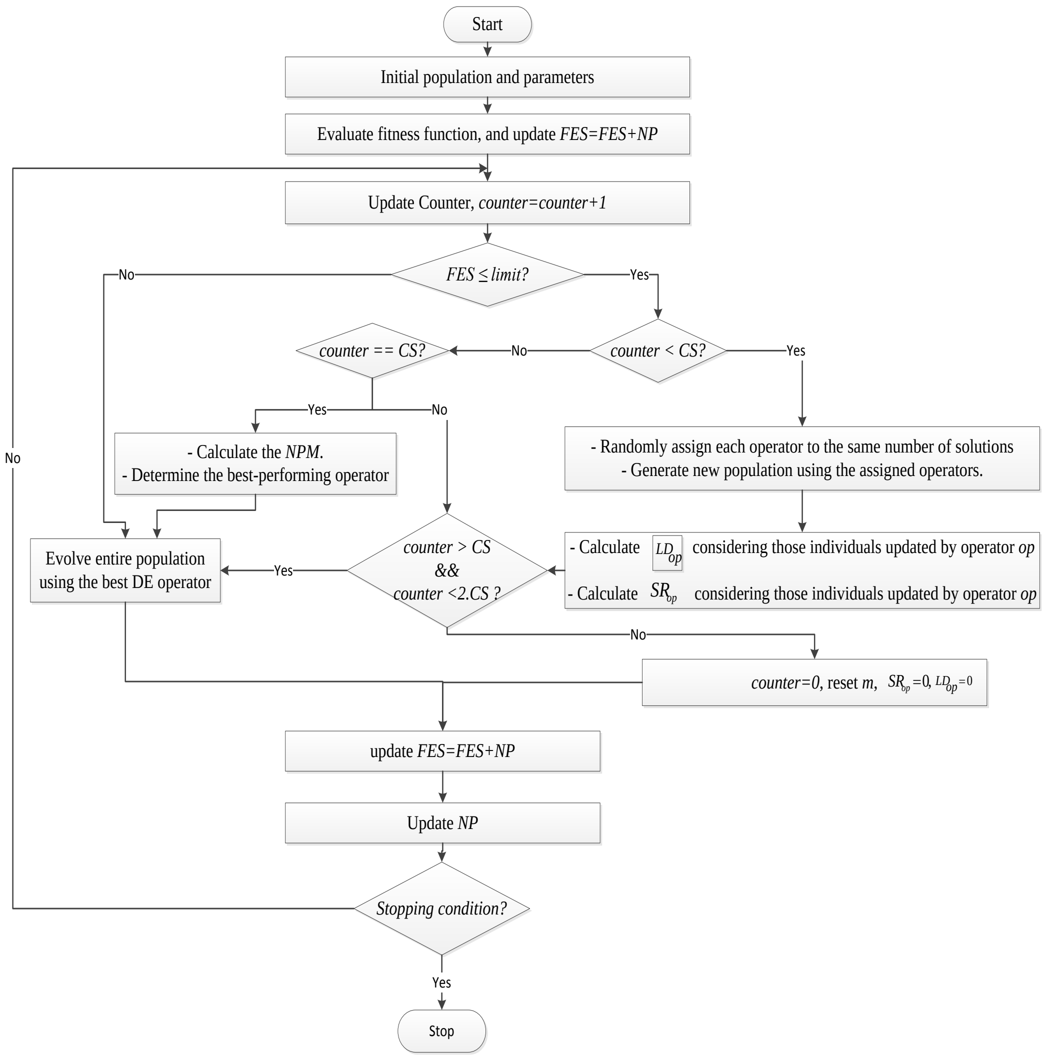


Figure 3.1: The general structure of LSAOS-DE

3.3.1.1 Initialization phase

In this phase, the LHD, which is a type of stratified sampling, is used to generate the initial population as it is capable of producing a sample of points that efficiently covers a search region.

The initialization is performed by:

$$x_{i,j,0} = x_{j,min} + lhd(1, NP) \times (x_{j,max} - x_{j,min})$$
$$i = 1,\ 2,\ ....,\ NP\, and\, j = 1,\ 2,\ ...,\ D \tag{3.2}$$

where lhd is a LHD function that generates random numbers from independent standard Normal distributions [200, 201].

3.3.1.2 Selection phase

In this section, the two selection criteria, landscape method and performance history measure, are discussed.

Information landscape negative searchability (ILNS) measure This is based on the difference between the information landscape vector of the problem to be solved and the well-known spherical function used as a reference landscape due to its simplicity and scalability [202] which is constructed by:

$$f_{ref}(\overrightarrow{x}) = \sum_{j=1}^{D}(x_j - x_j^b)^2 \tag{3.3}$$

where $\overrightarrow{x}_i^b$ is the best individual in the sample.

After constructing the vector landscapes of the problem to be optimized and the reference function (LS_f and LS_{ref}, respectively), the ILNS measure is computed using Algorithm 3 and the information landscape measure using equation 4.3 .

Normalized performance measure (NPM) In this section, the overall NPM of each operator is computed using the operator's searchability index (LD) and success rate SR_{op} which is defined as the number of successful offspring generated

Algorithm 3 Algorithm for computing ILNS index.

1: Input: population of individuals (X) updated by operator op;
2: Determine the location of the best individual in the sample, $\overrightarrow{x}^b$;
3: Construct the pairwise comparison matrix M using equation 2.15;
4: Construct vector LS_f that represent the information matrix of the problem;
5: Construct the reference function, f_{ref}, by using equation 3.3;.
6: Construct the vector LS_{ref} that represent the information landscape of the reference function;
7: Compute the value of the Information Landscape negative searchability index using equation 4.3;

by an operator (op) divided by the number of individuals assigned to that op as:

$$SR_{op} = \frac{\text{Number of improved offsprings}}{\text{Number of all individuals evolved by operator}\, op} \tag{3.4}$$

The capability of each operator to explore the landscape is calculated by Algorithm 3. The normalized values of the SR and LD metrics are calculated respectively by:

$$NM_{SR_{op}} = \frac{M_{SR_{OP}}}{\sum_{OP=1}^{m} M_{SR_{OP}}} \tag{3.5}$$

$$NM_{LD_{op}} = \frac{1 - M_{LD_{op}}}{\sum_{OP=1}^{m}(1 - M_{LD_{op}})} \tag{3.6}$$

where $\{op = 1, 2, ..., m\}$ and M_{SR} and M_{LD} are the mean values of the SR and landscape information, respectively.

Subsequently, the overall normalized performance of each operator is computed by:

$$NPM_{OP} = NM_{SR_{OP}} + NM_{LD_{op}} \tag{3.7}$$

3.3.2 Parameter adaptation

To calculate the values of the crossover rate (Cr_i) and control (F_i) parameters, the same adaptation procedure as reported in [67] is used, in which a historical memory of length H for both F and Cr, is used with their values denoted as μ_F and μ_{CR}, and initially set to 0.5. Each individual $\overrightarrow{X}_i$ is associated with its (F_i) and (CR_i), the values of which are respectively created by:

$$Cr_i = randni(\mu_{CR,r_i}, 0.1) \tag{3.8}$$

$$F_i = randci(\mu_{F,r_i}, 0.1) \tag{3.9}$$

where r_i is randomly selected from [1, H], $randni$ and $randci$ randomly selected from normal and Cauchy distributions with mean μ_{CR,r_i} and μ_{F,r_i}, respectively, and variance of 0.1. If the values of Cr_i and F_i are outside [0,1], a repair mechanism is used as follows: if CR_i is outside the range, it is replaced by the limit value (0 or 1) closest to the generated value and if $F_i > 1$, it is replaced by 1, and if $F_i \leq 0$, equation 3.9 is repeatedly executed until a valid value is generated.

At the end of each generation, the (Cr_i) and (F_i) used by the successful individuals are recorded in S_{CR} and S_F, and then the contents of the historical memories are updated as:

$$\mu_{CR,h,g+1} = \begin{cases} meanw_L(S_{Cr}) & \text{if } SCR \neq \phi \\ \mu_{CR,h,g} & otherwise \end{cases} \tag{3.10}$$

$$\mu_{F,h,g+1} = \begin{cases} meanw_L(S_F) & \text{if } SF \neq \phi \\ \mu_{F,h,g} & otherwise \end{cases} \tag{3.11}$$

where $1 \leq h \leq H$ is the position in the memory to be updated. It is initialized to 1 and then incremented whenever a new element is inserted into the history, and if it is greater than H, it is set to 1 and $meanw_L(SF)$ the Lehmer mean computed by:

$$meanw_L(SF) = \frac{\sum_{h=1}^{|SF|} w_h.S_{F_h}^2}{\sum_{h=1}^{|SF|} w_h.S_{F_h}} \tag{3.12}$$

where w_h is the weight computed using equation 3.13

$$w_h = \frac{|(f(u_{h,g+1}) - f(x_{h,g}))|}{\sum_{h=1}^{|S_F|} |(f(u_{h,g+1}) - f(x_{h,g}))|} \tag{3.13}$$

3.4 Experimental Setup and Results

To judge the performance of the proposed algorithm, experiments were conducted by solving the set of benchmark problems presented in Section 3.2.

To verify the performance of the proposed algorithm, comparisons are carried out with three DE-based single operator algorithms (LSHADE [67], SHADE [148], and JADE [64]), four multioperator-based ones (CoDE [68], SaDE [155], EPSDE [156], and MPEDE [22]), and two powerful multi-method based ones (UMOEAs [104] and AMALGAM-SO [103]). Also, to test the effectiveness of the proposed selection mechanism LSAOS-DE was also compared with seven of its variants (AOSDE-LS, AOSDE-SR, AOSDE-FIR, AOSDE-PR, AOSDE-PBM, AOSDE-RD, and AOSDE-PAOC), in each of which only the selection mechanisms were different as will be described in Section Heuristics chapter3.

All the algorithms proposed in this thesis were coded in Matlab, and were run on a PC with a 3.4 GHz Core I7 processor, 16 GB RAM, and Windows 7. For the state-of-the-art algorithms, all the parameters were set at those values recommended in their corresponding papers, and for a fair comparison, all the algorithms started from the same seed. As the original JADE paper only reported population sizes for $10D$, $30D$, and $100D$, an approximate value of 185 individuals for $50D$ was used. For more comprehensive comparisons, each comparative algorithm was run 51 times for all the optimization functions with $10D$, $30D$, and $50D$, to a maximum number of fitness function evaluations of $10000D$, with the best, mean and standard deviation results recorded. Note that, for any run, if the deviation from the optimal solution of the best fitness value was less than or equal to $1.0e-8$, it was considered to be zero.

In this thesis, to compare different algorithms, two non-parametric tests (the Wilcoxon signed-rank [203] and Friedman's ranking [204]) were conducted. The Wilcoxon signed-rank test assigns one of three signs ($+$, $-$, and $\approx$) for the comparison of any two algorithms, where the "$+$" sign means the first algorithm was significantly better than the second, the "$-$" that it was significantly worse, and the "$\approx$" that there was no significant difference between the two algorithms. A 5% significance level was used to judge the difference between any pair of algorithms. A Friedman test was also used to rank the algorithms.

To compare the performance of the proposed algorithm graphically, performance profiles [205, 206], which are a tool to compare performance, of a number of algorithms (S) using a set of problems (P), and a comparison goal such as the computational time and the average number of fitness evaluations to obtain a certain level, of a performance indicator (such as optimal fitness), are plotted. For an algorithm (s), the performance profile Rho_s is given by

$$Rho_s(\tau) = \frac{1}{n_p} \times |p \in P : r_{p,s} \leq \tau| \tag{3.14}$$

$Rho_s(\tau)$ is the probability for an algorithm $s \in S$ that a performance ratio $r_{p,s}$ is within a factor $\tau \in R$ of the best possible ratio. The function Rho_s is the (cumulative) distribution function for the performance ratio.

3.4.1 Parameters analysis

In this section, analysis of the algorithm's components, to determine the final version of the proposed algorithm, are discussed. The effect of the following parameters: 1) the effect of the number of DE operators m; 2) the effect of reducing m; 3) the effect of NP^{init}; 4) the effect of CS; and 5) the effect of NP^{min}, are examined.

3.4.1.1 Default values

As for parameters settings, NP^{init} was set to $18D$, NP^{min} to 7. The CS parameter to 25. φ to 0.5 for DE/φbest/1 was used to maintain diversity, and 0.1 for all other variants, to speed up convergence rate. The archive rate (A) was 1.4, memory size (H) was 5, and *limit*, which was the limit for running the multi-operator, was $\frac{1}{2} \times MAX_{FES}$, after which the best-performing operator evolved the population until the end of the run.

3.4.1.2 Analyzing the number of operators in LSAOS-DE (m)

To design a multi-operator DE algorithm, one can first analyze and determine the mutation strategies that should be used. 11 variants of the standard DE algorithm (DE/φbest/1, DE/current-to-φbest/1/ with archive, DE/current-to-φbest/1/ without archive, DE/rand/1,DE/rand/2, DE/current-to-rand/1/ with

Table 3.3: Friedman's test results

Algorithm	Mean rank	Order
DE/current-to-φbest/1 without archive	4.22	1
DE/current-to-φbest/1 archive	4.63	2
DE/φbest/1	5.57	3
DE/rand-to-φbest/1 without archive	5.6	4
DE/rand/1	6.07	5
DE/rand-to-φbest/1 with archive	6.17	6
DE/current-to-rand/1	6.35	7
DE/rand-to-φbest/2 with archive	6.37	8
DE/current-to-rand/1 without archive	6.48	9
DE/φbest/2	7.02	10
DE/rand/2	7.53	11

archive, DE/current-to-rand/1/ without archive, DE/rand-to-φbest/1/ with archive, DE/rand-to-φbest/1 without archive, DE/φbest/2, and DE/rand-to-φbest/2/with archive) were tested by solving CEC2014 problems with $30D$, and the average results were recorded. Based on this, the variants were ranked by a Friedman test and their mean values are presented in Table 3.3.

The number of DE mutation strategies (m) required for LSAOS-DE was analyzed, by conducting different experiments using m =3, 4, 5, 6, and 7 (best variants based on results obtained in Table 3.3) to solve the $30D$ problems, with the remaining parameters set as mentioned in Section 3.4.1.1. Note that increasing the number of operators does not necessarily mean achieving better performance [24]. One reason for this, is that operators may not be complementary, which may cause loss of fitness evaluations and/or a biased search process.

Summary of the results are presented in Table 3.4. To understand it, for example, row 4, which compares $m = 3$ and $m = 7$, indicates that $m = 3$ was superior for 15 test problems, obtained the same results for 9, but was inferior for 6, which means that $m = 3$ was better than $m = 7$.

Regarding the Wilcoxon signed-rank test, it is clear that $m = 5$ is better than

Table 3.4: Summary of comparison of different variants of proposed algorithm with different m values (3, 4, 5, 6 and 7) is based on average results

	Algorithms	Better	Equal	Worse	Dec.
Mean	$m = 3$Vs. $m = 4$	12	7	11	$\approx$
	$m = 3$ vs. $m = 5$	10	8	12	$\approx$
	$m = 3$ vs. $m = 6$	13	8	9	$\approx$
	$m = 3$ vs. $m = 7$	15	9	6	$+$
	$m = 4$ vs. $m = 5$	7	11	12	$-$
	$m = 4$ vs. $m = 6$	16	7	7	$+$
	$m = 4$ vs. $m = 7$	15	7	8	$+$
	$m = 5$ vs. $m = 6$	17	7	6	$+$
	$m = 5$ vs. $m = 7$	15	9	6	$+$
	$m = 6$ vs. $m = 7$	13	8	9	$\approx$

* Note Dec. is based on the Wilcoxon signed test

the others, as confirmed by the Friedman test, the results of which are shown in Figure 3.2(a).

Furthermore, the performance profiles are depicted in Figure 3.3(a), in which, one can observe from the figure, that the variant with $m = 5$ outperforms other variants.

3.4.1.3 Analyzing the reduction mechanism

In this section, the effect of reducing m during a run was analyzed. To do this, four different variants of the proposed algorithm (Ver1, Ver2, Ver3 and Ver4) were run. In Ver1, m was reduced from 5 to 1; in Ver2 m was reduced from 5 to 4 to 3 to 2 to 1, m was reduced from 5 to 3 to 1 in Ver3 and reduced from 5 to 3 to 2 to 1 in Ver4. To show the benefit of reducing m over generations, another variant (Ver5) with no reduction was run. In Ver5, all the five search operators were used in each generation and the number of individuals evolved by each mutation strategy was updated based on the proposed selection mechanism. All the parameters setting are set as those in the previous section. The detailed results (average, St.d) are presented in Appendix A, Table A.11.

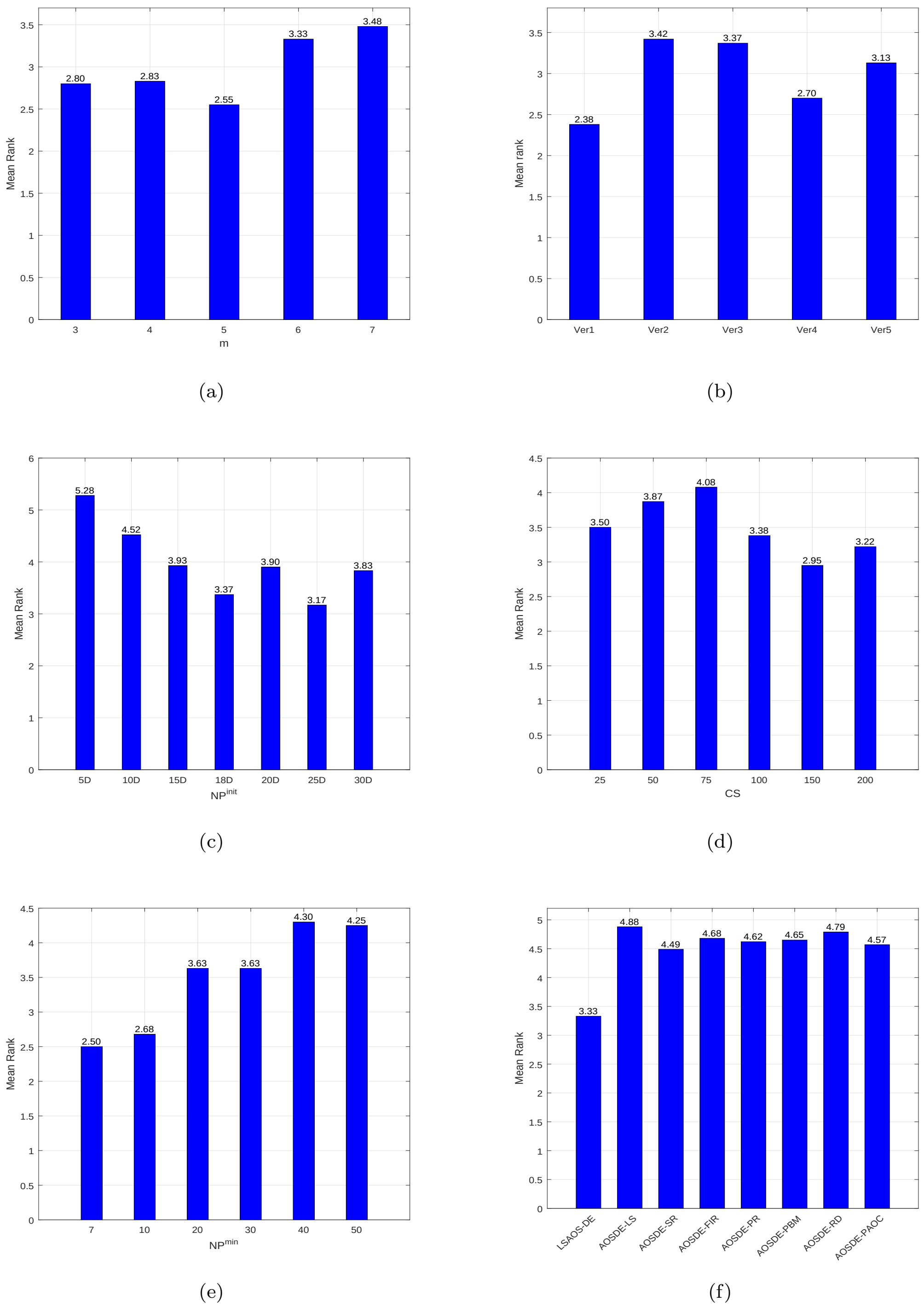


Figure 3.2: Average rankings of (a) different variants for $m = 3$, 4, 5, 6, and 7 mutation strategies (b) five different variants, (c) different values of NP^{init}, (d) different values of CS, (e) different values of NP^{min} and (f) LSAOS-DE and seven other variants

Table 3.5: Summary of comparisons of different variants of proposed algorithm with different variants for reducing m based on average results, where 'Dec.' statistical decision based on Wilcoxon signed-rank test results

Algorithms	Better	Equal	Worse	Prob.	Dec. $\alpha = 0.05$	Dec. $\alpha = 0.10$
$Ver1$ vs. $Ver2$	19	7	4	0.002	+	+
$Ver1$ vs. $Ver3$	17	8	5	0.026	+	+
$Ver1$ vs. $Ver4$	12	10	8	0.332	≈	≈
$Ver1$ vs. $Ver5$	14	8	8	0.042	+	+
$Ver2$ vs. $Ver3$	11	7	12	0.891	≈	≈
$Ver2$ vs. $Ver4$	8	7	15	0.097	≈	−
$Ver2$ vs. $Ver5$	11	6	13	0.909	≈	≈
$Ver3$ vs. $Ver4$	8	6	16	0.103	≈	≈
$Ver3$ vs. $Ver5$	10	7	13	0.761	≈	≈
$Ver4$ vs. $Ver5$	15	7	8	0.059	≈	+

Table 3.6: Friedman test results

Algorithm	Mean rank	Order
Ver1	2.38	1
Ver2	3.42	5
Ver3	3.37	4
Ver4	2.70	2
Ver5	3.13	3

Table 3.5 presents shows the summary of the obtained results, based on the average results. Regarding the Wilcoxon signed-rank test, Ver1 is the best. The overall rankings for the different five variants are presented in Table 3.6 and Figure 3.2(b), in which Ver1 has the best performance. These results are also confirmed by the plots of the performance profiles depicted in Figure 3.3(b). From the above comparison, it is clear that Ver1 is the best among the variants which used a reduction mechanism for m. Comparing Ver1 with Ver5, Ver1 is better, which means that reducing the number of DE mutations strategy is better than, fixing the number of DE mutation strategies during the search process.

3.4.1.4 Effect of NP^{init}

The influence of NP^{init} on the performance of LSAOS-DE was investigated by conducting several experiments with NP^{init} =5D, 10D, 18D, 20D, 25D, and 30D for solving the 30D problems. The remaining parameters were fixed as mentioned in Section 3.4.1.1, detailed results are shown in Appendix A, Table A.12. Table 3.7 shows the summary of comparisons for different NP^{init} values, based on the average obtained results. Regarding the Wilcoxon signed-rank test results, it is clear that $NP = 25D$ was the best for all variants. Further analysis based on the Friedman test for different values of NP^{init} was conducted. The mean ranks depicted in Figure 3.2(c) also show that $NP^{init} = 25D$ was the best.

Furthermore, the performance profiles graph is depicted in Figure 3.3(c). From which, it is clear that the variant with $NP^{init} = 25D$ was better than all the others.

3.4.1.5 Effect of CS

The effect of CS, which determines the timing of reductions in the number of mutation strategies, was analyzed. This is done by utilizing different numbers of generations, 25, 50, 75, 100, 150, and 200 for CS, while all other parameters were set as in previous sections (detailed results are shown in Appendix A, Table A.13). The overall rankings for all the different CS values, based on the Friedman test are depicted in Figure 3.2(d), in which it is clear that higher values of CS had good impacts on performance, with $CS = 150$ preferred.

Table 3.8 presents a summary of the comparisons of six different CS values. Regarding the statistical test, it was found that $CS = 150$ better than the others.

From the performance profiles of the six different CS values depicted in Figure

Table 3.7: Summary of comparisons of different variants of proposed algorithm with different NP^{init} values ($5D$, $10D$, $18D$, $20D$, $25D$ and $30D$) based on average results, where 'Dec.' statistical decision based on Wilcoxon signed-rank test results

Algorithms	Better	Equal	Worse	Prob.	Dec. $\alpha = 0.05$	Dec. $\alpha = 0.10$
$5D$ vs. $10D$	**4**	**8**	**18**	**0.006**	–	–
$5D$ vs. $15D$	**5**	**6**	**19**	**0.002**	–	–
$5D$ vs. $18D$	**4**	**6**	**20**	**0.001**	–	–
$5D$ vs. $20D$	**7**	**5**	**18**	**0.013**	–	–
$5D$ vs. $25D$	**6**	**4**	**20**	**0.005**	–	–
$5D$ vs. $30D$	**9**	**4**	**17**	**0.028**	–	–
$10D$ vs. $15D$	**6**	**7**	**17**	**0.001**	–	–
$10D$ vs. $18D$	**6**	**8**	**16**	**0.002**	–	–
$10D$ vs. $20D$	**9**	**6**	**15**	**0.034**	–	–
$10D$ vs. $25D$	**7**	**5**	**18**	**0.005**	–	–
$10D$ vs. $30D$	**9**	**5**	**16**	**0.048**	–	–
$15D$ vs. $18D$	**8**	**7**	**15**	**0.052**	$\approx$	–
$15D$ vs. $20D$	11	6	13	0.376	$\approx$	$\approx$
$15D$ vs. $25D$	**8**	**4**	**18**	**0.032**	–	–
$15D$ vs. $30D$	11	6	13	0.230	$\approx$	$\approx$
$18D$ vs. $20D$	16	5	9	0.122	$\approx$	$\approx$
$18D$ vs. $25D$	11	5	14	0.451	$\approx$	$\approx$
$18D$ vs. $30D$	13	5	12	0.936	$\approx$	$\approx$
$20D$ vs. $25D$	9	6	15	0.119	$\approx$	$\approx$
$20D$ vs. $30D$	11	8	11	0.709	$\approx$	$\approx$
$25D$ vs. $30D$	15	6	9	0.141	$\approx$	$\approx$

3.3(d), it is clear that the variant with $CS = 150$ was the first to reach a probability of 1, with τ between 4 and 5.

3.4.1.6 Effect of NP^{min}

As previously mentioned, the number of individuals in the whole population was initially NP^{init} which was then linearly reduced to NP^{min}. To analyze the effect of NP^{min}, different experiments were conducted using different NP^{min} values of 7, 10, 20, 30, 40, and 50, while all the other parameters were fixed, as previously discussed (detailed results are shown in Appendix A, Table A.14). All the variants were ranked based on the Friedman test and their results depicted in

Table 3.8: Summary of comparisons of different variants of proposed algorithm with different CS values (25, 50, 75, 100, 150 and 200) based on average results, where 'Dec.' statistical decision based on Wilcoxon signed-rank test results

Algorithms	Better	Equal	Worse	Prob.	Dec. $\alpha = 0.05$	Dec. $\alpha = 0.10$
$CS = 25$ vs. $CS = 50$	12	9	9	0.566	$\approx$	$\approx$
$CS = 25$ vs. $CS = 75$	13	9	8	0.313	$\approx$	$\approx$
$CS = 25$ vs. $CS = 100$	10	9	11	0.741	$\approx$	$\approx$
$CS = 25$ vs. $CS = 150$	**8**	**9**	**13**	**0.095**	$\approx$	$-$
$CS = 25$ vs. $CS = 200$	10	8	12	0.256	$\approx$	$\approx$
$CS = 50$ vs. $CS = 75$	9	10	11	0.709	$\approx$	$\approx$
$CS = 50$ vs. $CS = 100$	7	9	14	0.251	$\approx$	$\approx$
$CS = 50$ vs. $CS = 150$	**8**	**7**	**15**	**0.089**	$\approx$	$-$
$CS = 50$ vs. $CS = 200$	10	7	13	0.140	$\approx$	$\approx$
$CS = 75$ vs. $CS = 100$	8	8	14	0.168	$\approx$	$\approx$
$CS = 75$ vs. $CS = 150$	**3**	**8**	**19**	**0.004**	$-$	$-$
$CS = 75$ vs. $CS = 200$	**6**	**8**	**16**	**0.020**	$-$	$-$
$CS = 100$ vs. $CS = 150$	**8**	**9**	**13**	**0.076**	$\approx$	$-$
$CS = 100$ vs. $CS = 200$	9	10	11	0.232	$\approx$	$\approx$
$CS = 150$ vs. $CS = 200$	10	10	10	1	$\approx$	$\approx$

Table 3.9: Summary of comparisons of different variants of proposed algorithm with different NP^{min} values (7, 10, 20, 30, 40 and 50) based on average results, where 'Dec.' statistical decision based on Wilcoxon signed-rank test results

Algorithms	Better	Equal	Worse	Prob.	Dec. $\alpha = 0.05$	Dec. $\alpha = 0.10$
$NP^{min} = 7$ vs. $NP^{min} = 10$	14	8	8	0.961	$\approx$	$\approx$
$NP^{min} = 7$ vs. $NP^{min} = 20$	**19**	**8**	**3**	**0.002**	+	+
$NP^{min} = 7$ vs. $NP^{min} = 30$	**19**	**7**	**4**	**0.006**	+	+
$NP^{min} = 7$ vs. $NP^{min} = 40$	**18**	**6**	**6**	**0.063**	$\approx$	+
$NP^{min} = 7$ vs. $NP^{min} = 50$	**18**	**5**	**7**	**0.042**	+	+
$NP^{min} = 10$ vs. $NP^{min} = 20$	**18**	**7**	**5**	**0.006**	+	+
$NP^{min} = 10$ vs. $NP^{min} = 30$	**17**	**7**	**6**	**0.005**	+	+
$NP^{min} = 10$ vs. $NP^{min} = 40$	**21**	**7**	**2**	**0.000**	+	+
$NP^{min} = 10$ vs. $NP^{min} = 50$	**18**	**6**	**6**	**0.001**	+	+
$NP^{min} = 20$ vs. $NP^{min} = 30$	15	6	9	0.265	$\approx$	$\approx$
$NP^{min} = 20$ vs. $NP^{min} = 40$	16	7	7	0.191	$\approx$	$\approx$
$NP^{min} = 20$ vs. $NP^{min} = 50$	15	6	9	0.130	$\approx$	$\approx$
$NP^{min} = 30$ vs. $NP^{min} = 40$	**17**	**9**	**4**	**0.068**	$\approx$	+
$NP^{min} = 30$ vs. $NP^{min} = 50$	**18**	**5**	**7**	**0.013**	+	+
$NP^{min} = 40$ vs. $NP^{min} = 50$	**14**	**7**	**9**	**0.097**	$\approx$	+

Figure 3.2(e). Also, a Wilcoxon signed-rank test of the different variants of the proposed algorithm, with different NP^{min} values was conducted and the summary of the results presented in Table 3.9 indicates that the setting of $NP^{min} = 7$ was slightly better than those of other variants, with small values preferred.

The performance profiles for different values of NP^{min} are depicted in Figure 3.3(e), which shows that the variant with $NP^{min} = 7$ had the highest probability at the beginning, and was the first to reach a probability of 1, with τ between 70 and 80.

3.4.2 Comparisons of proposed algorithm and other heuristic-based selection methods

In this section, to assess the effect of the proposed selection mechanism, a comparison of LSAOS-DE and seven of its variants, namely AOSDE-LS, AOSDE-SR, AOSDE-FIR, AOSDE-PR, AOSDE-PBM, AOSDE-RD, and AOSDE-PAOC, in each of which only the selection mechanisms were different, and below is a description of the selection mechanism in every variant:

1. In AOSDE-LS, a landscape information is used to select the best-performing DE mutation strategy as in Equation 3.6.

2. In ASODE-SR, to select the best-performing operator, a success rate was used as in Equation 3.4.

3. In ASODE-FIR, fitness improvement rate, which is used to evaluate the difference in quality between the parent solution and its offspring in a normalized manner [108], is used and is computed by

$$FIR_{op} = \frac{pf - cf}{pf} \tag{3.15}$$

where pf and cf are the best fitness value for parent and child solution, respectively.

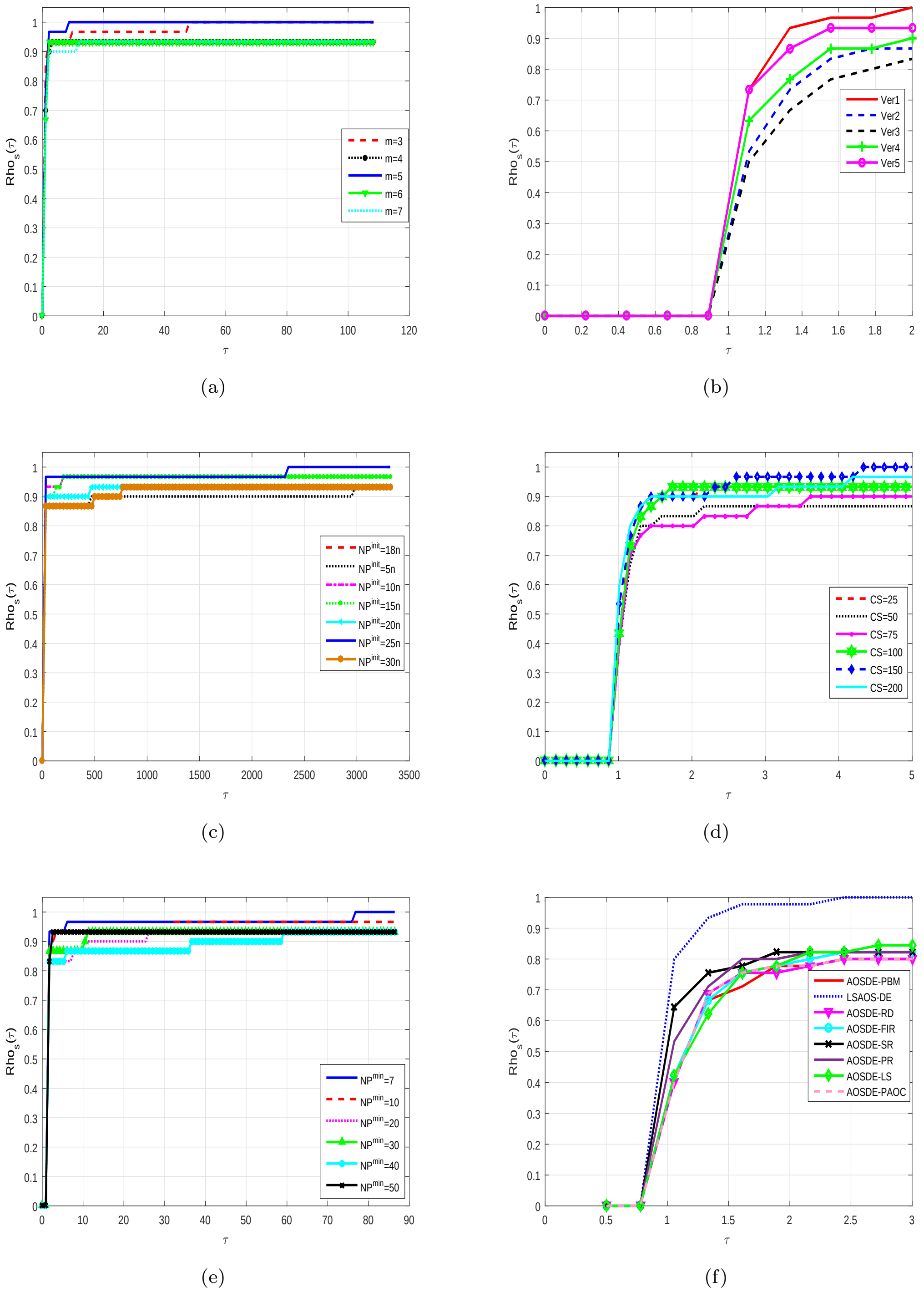


Figure 3.3: Performance profiles for (a) m with different values, (b) five different variants, (c) NP^{init} with different values, (d) CS with different values, (e) NP^{min} with different values and (f) LSAOS-DE and other seven variants, where $Rho_s(\tau)$ is the fraction of problems that a solver was able to solve within a factor of $\tau \geq 1$ of the best observed performance.

4. In AOSDE-PR, progress ratio [106], which is used to evaluate the ability of an operator to improve its best solution, is computed by

$$PR = |ln\sqrt{\frac{f_{min}(t-1)}{f_{min}(t)}}| \tag{3.16}$$

5. In AOSDE-PBM, in each generation the difference between the best fitness found by each operator, and the best known solution is calculated. An exponential function, (ae^{bx}), is fitted to these values. Then this model is used to calculate the expected absolute error after CS subsequent generations, and the DE mutation strategy that has the minimum expected absolute error is chosen as the best-performing operator [104].

6. In AOSDE-RD, one of the DE mutation strategies is randomly selected every CS generation to evolve the entire population for the subsequent CS generations.

7. In AOSDE-PAOC, DE mutation strategies are given probabilities in proportion to the improvement of fitness over two iterations [109]. This probability is computed by

$$Pb(t) = Pb(t-1) + \sum_{i=1}^{S_{op}} (f(x_i(t-1)) - f(x_i(t)) \tag{3.17}$$

where S_{op} is the number of individuals evolved by operator op and $Pb_{op}(0)$ is the initial probability and is calculated by:

$$Pb_{op}(0) = \frac{1}{S_{op}} \tag{3.18}$$

Detailed results are shown in Appendix A, Tables A.8, A.9, A.10. A Friedman test was carried out to rank all the algorithms based on their average results

Table 3.10: Friedman test results obtained from comparing LSAOS-DE and seven variants

	10D		30D		50D		Overall rank
Algorithm	Mean rank	Order	Mean rank	Order	Mean rank	Order	
LSAOS-DE	**3.42**	1	**3.38**	1	**3.18**	1	1
AOSDE-LS	4.86	6	4.74	6	5.03	5	8
AOSDE-SR	5.07	7	4.13	2	4.28	2	2
AOSDE-FIR	4.79	5	4.98	7	4.28	2	6
AOSDE-PR	4.50	4	4.66	5	4.69	4	4
AOSDE-PBM	4.27	3	4.47	3	5.22	6	5
AOSDE-RD	4.26	2	5.00	8	5.11	7	7
AOSDE-PAOC	4.86	6	4.64	4	4.21	3	3

* Overall rank is based on the sum of the orders for 10D, 30D, and 50D

as shown in Table 3.10 and Figure 3.2(e), in which LSAOS-DE was ranked first for the $10D$, $30D$ and $50D$ test problems, with AOSDE-RD second for the $10D$, AOSDE-SR second for the $30D$, and AOSDE-SR and AOSDE-FIR equal second for the $50D$ ones.

Considering the quality of solutions obtained, a comparison was carried out of LSAOS-DE and all the other variants, with the results shown in Table 3.11. It is clear that, LSAOS-DE is better than all other algorithms.

Regarding Wilcoxon signed-rank test, for the $10D$, LSAOS-DE is significantly better than all other algorithms and it is significantly better than all other algorithms, except AOSDE-SR and AOSDE-PAOC, for $30D$ and $50D$ problems, respectively. Generally speaking, it is evident that the proposed algorithm was better than its variants.

Also, the average total computational time and FEs is presented in Table 3.12, which shows that the proposed LSAOS-DE requires the least average computational time and FEs. That means the propos LSAOS-DE saves computational time by 15.45%, 14.87%, 15.14%, 15.20%, 15.40%, 14.57% and 13.94%, and FEs by 13.81%, 14.34%, 15.73%, 14.28%, 13.29%, 13.55% and 15.45% in comparison to AOSDE-LS, AOSDE-SR, AOSDE-FIR, AOSDE-PR, AOSDE-PBM, AOSDE-RD

Table 3.11: Summary of comparisons of proposed algorithm and seven variants, where 'Dec.' statistical decision based on Wilcoxon signed-rank test results

	Algorithms	Better	Equal	Worse	p-value	Dec.
10D	LSAOS-DE vs. AOSDE-LS	24	11	10	0.004	+
	LSAOS-DE vs. AOSDE-SR	27	8	10	0.017	+
	LSAOS-DE vs. AOSDE-FIR	25	9	11	0.010	+
	LSAOS-DE vs. AOSDE-PR	27	8	10	0.006	+
	LSAOS-DE vs. AOSDE-PBM	24	10	11	0.007	+
	LSAOS-DE vs. AOSDE-RD	21	10	14	0.047	+
	LSAOS-DE vs. AOSDE-PAOC	25	10	10	0.004	+
30D	LSAOS-DE vs. AOSDE-LS	32	3	10	0.003	+
	LSAOS-DE vs. AOSDE-SR	25	4	16	0.141	$\approx$
	LSAOS-DE vs. AOSDE-FIR	29	2	14	0.038	+
	LSAOS-DE vs. AOSDE-PR	28	4	13	0.035	+
	LSAOS-DE vs. AOSDE-PBM	31	2	12	0.009	+
	LSAOS-DE vs. AOSDE-RD	28	3	14	0.037	+
	LSAOS-DE vs. AOSDE-PAOC	28	4	13	0.015	+
50D	LSAOS-DE vs. AOSDE-LS	35	1	9	0.000	+
	LSAOS-DE vs. AOSDE-SR	29	3	13	0.043	+
	LSAOS-DE vs. AOSDE-FIR	28	2	15	0.036	+
	LSAOS-DE vs. AOSDE-PR	29	2	14	0.033	+
	LSAOS-DE vs. AOSDE-PBM	32	1	12	0.001	+
	LSAOS-DE vs. AOSDE-RD	37	1	7	0.000	+
	LSAOS-DE vs. AOSDE-PAOC	28	2	15	0.070	$\approx$

Table 3.12: Comparison among LSAOS-DE and seven of its variants based on the average computational time and FEs

{Algorithms}	**Total average time**	**Total average FEs**
LSAOS-DE	**231.28**	**7567184**
AOSDE-LS	273.56	8779867
AOSDE-SR	271.68	8833511
AOSDE-FIR	272.56	8979867
AOSDE-PR	272.72	8827440
AOSDE-PBM	273.39	8726557
AOSDE-RD	270.73	8752952
AOSDE-PAOC	268.75	8950416

and AOSDE-PAOC, respectively.

Furthermore, from the graphs of the performance profiles depicted in Figure 3.3(f), it is clear that LSAOS-DE had the highest probability at the start, and reached a probability of 1first at $\tau = 2.5$, while AOSDE-SR was second.

As a further illustration, the convergence plots of the best results obtained during 51 runs of LSAOS-DE, AOSDE-SR, AOSDE-LS for the $30D$ problems, for four different functions, F01, F06, F20 and F30, are shown in Figure 3.4, in which it is clear that the convergence speed of LSAOS-DE was the best. It can be concluded that, a balance between the success rate and landscape information leads to good performances, in comparison to the other cases.

3.4.3 Comparisons of proposed and state-of-the-art algorithms

Experiments were conducted to compare the performance of the proposed algorithm with those of state-of-the-art ones using the best set of previously determined parameter values, that is, $NP^{init} = 25D$, $NP^{min} = 7$, $CS = 150$ and NP updated using equation 3.1.

Details of the computational results $(f(x_{best}) - f(x^*))$ obtained from the proposed algorithm for $10D$, $30D$, and $50D$ problems are shown in Appendix A, (Tables A.1, A.2, A.3, A.4, A.5, A.6, A.7), in which the average values and standard deviations of the solutions are compared with those obtained from the state-of-the-art algorithms.

The proposed method obtained better results than all the other algorithms for more $10D$ test problems. For the $30D$ ones, LSAOS-DE obtained better results than all the other algorithms although AMALGAM-SO was competitive in terms of its average results. For the $50D$ problems, the proposed method performed better than all the other algorithms except LSHADE, which was competitive in terms of its average values but there was a bias towards LSAOS-DE.

The results based on the average fitness values obtained from the Wilcoxon

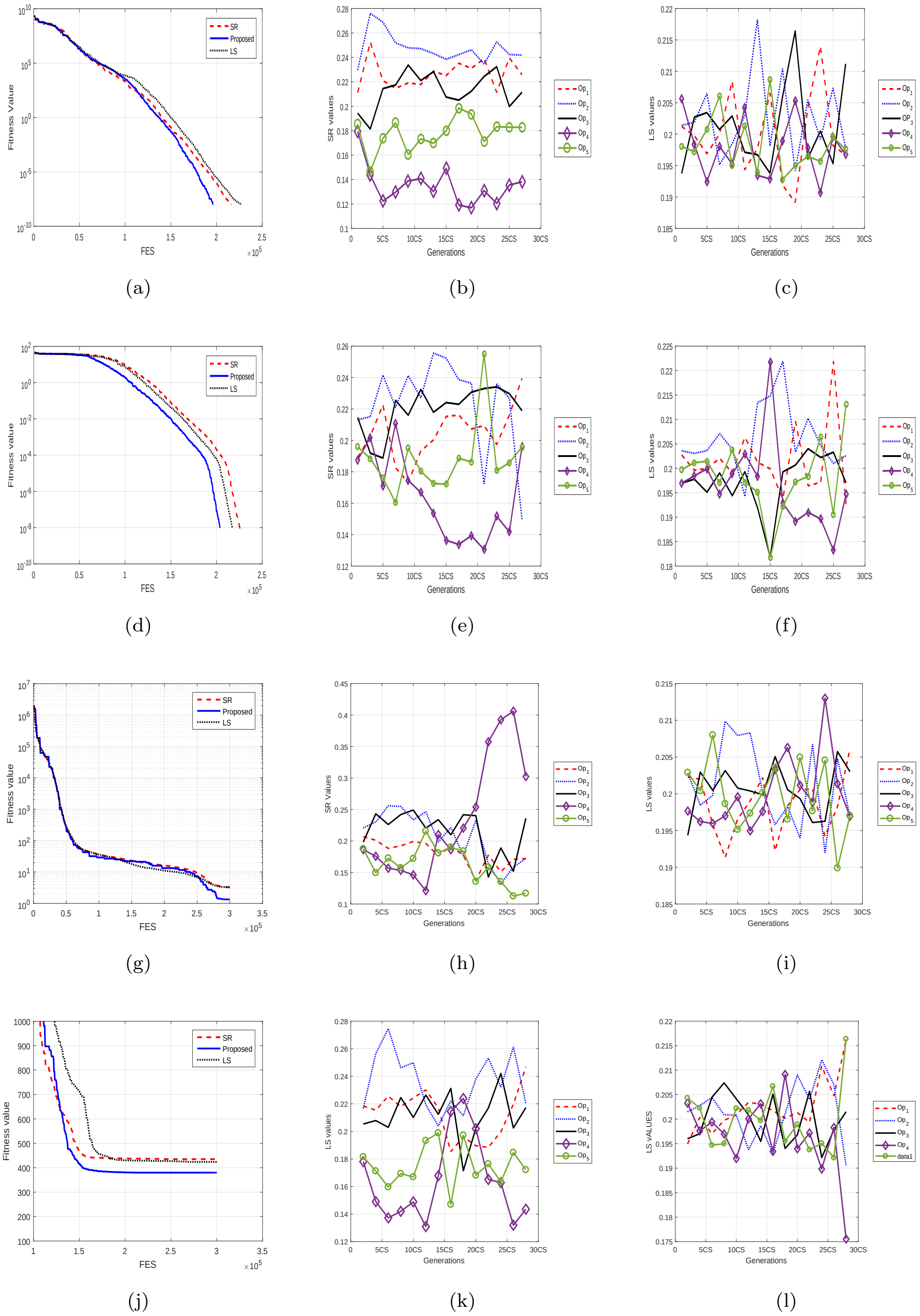


Figure 3.4: Convergence graphs which compares LSAOS-DE with both AOSDE-LS and AOSDE-SR, with the values of SR and LS (a,b,c) F01, (d,e,f) F06, (g,h,i) F20 and (j,k,l) F30

Table 3.13: Summary of comparisons of proposed algorithm and nine state-of-the-art algorithms based on mean results, where 'Dec.' statistical decision based on Wilcoxon signed-rank test results.

	Algorithms	Better	Equal	Worse	p-value	Dec.
10D	LSAOS-DE vs. LSHADE	20	12	13	0.057	≈
	LSAOS-DE vs. JADE	35	8	2	0.000	+
	LSAOS-DE vs. SHADE	33	9	3	0.000	+
	LSAOS-DE vs. CoDE	25	8	12	0.022	+
	LSAOS-DE vs. SaDE	32	8	5	0.000	+
	LSAOS-DE vs. EPSDE	29	8	8	0.011	+
	LSAOS-DE vs. UMOEAs	28	8	9	0.001	+
	LSAOS-DE vs. MPEDE	28	9	8	0.001	+
	LSAOS-DE vs. AMALGAM-SO	31	2	12	0.006	+
	LSAOS-DE vs. LSHADE	23	5	17	0.125	≈
	LSAOS-DE vs. JADE	33	4	8	0.000	+
	LSAOS-DE vs. SHADE	36	3	6	0.000	+
	LSAOS-DE vs. CoDE	35	1	9	0.000	+
	LSAOS-DE vs. SaDE	40	1	4	0.000	+
	LSAOS-DE vs. EPSDE	36	0	9	0.001	+
	LSAOS-DE vs. UMOEAs	31	4	10	0.001	+
	LSAOS-DE vs. MPEDE	32	4	9	0.000	+
	LSAOS-DE vs. AMALGAM-SO	31	0	14	0.019	+
50D	LSAOS-DE vs. LSHADE	23	5	17	0.125	≈
	LSAOS-DE vs. JADE	33	4	8	0.000	+
	LSAOS-DE vs. SHADE	36	3	6	0.000	+
	LSAOS-DE vs. CoDE	35	1	9	0.000	+
	LSAOS-DE vs. SaDE	40	1	4	0.000	+
	LSAOS-DE vs. EPSDE	36	0	9	0.001	+
	LSAOS-DE vs. UMOEAs	31	4	10	0.001	+
	LSAOS-DE vs. MPEDE	32	4	9	0.000	+
	LSAOS-DE vs. AMALGAM-SO	31	0	14	0.019	+

signed-rank test presented in Table 3.13 demonstrate that the proposed algorithm was statistically better than all the state-of-the-art ones in terms of both the average and best results for the $10D$, $30D$ and $50D$ problems,except for LSHADE and AMALGAM-SO for the means of the $50D$ and $30D$ ones, respectively, where there was no significant difference.

For further analysis, a Friedman test was performed to rank all algorithms based on the average fitness values they achieved. The results shown in Table 3.14

Table 3.14: Friedman test results obtained from comparisons of LSAOS-DE and state-of-the-art algorithms

	10D		30D		50D		
Algorithm	Mean rank	Order	Mean rank	Order	Mean rank	order	Overall rank
LSAOS-DE	**3.38**	1	**3.12**	1	**3.01**	1	1
LSHADE	4.02	2	3.90	2	3.91	2	2
JADE	5.96	5	6.59	8	6.34	7	8
SHADE	5.49	4	5.72	5	5.64	6	4
CoDE	4.87	3	5.88	7	6.52	8	7
SaDE	6.40	7	7.27	10	7.41	10	9.5
EPSDE	6.93	9	7.17	9	7.27	9	9.5
UMOEAs	6.37	6	5.84	6	5.47	5	6
MPEDE	4.87	3	4.11	3	4.53	3	3
AMALGAM-SO	6.72	8	5.40	4	4.89	4	5

* Overall rank is based on the sum of the orders for 10D, 30D, and 50D

indicate that, generally, LSAOS-DE ranked 1^{st} followed by LSHADE, MPEDE, SHADE, AMALGAM-SO, UMOEAs, JADE, CoDE, SaDE and EPSDE.

As a further illustration, the convergence plots of the average results obtained during 51 runs of all the algorithms for the $30D$ problems are shown in Figure 3.5, in which it is evident that the convergence speed of LSAOS-DE was the best.

3.4.4 Relative time complexity

This section describes the algorithmic complexity of the proposed LSAOS-DE code in which, as defined in [198, 199]. As defined in [198, 199], T_0 is the time calculated for running the code :

for $i = 1 : 1000000$

$x = 0.55+$(double)i;$x = x + x$; $x = x/2$; $x = x * x$;

$x = sqrt(x)$; $x = log(x)$; $x = exp(x)$; $x = x/(x + 2)$;

end

where T_1 is the time to execute 200,000 evaluations of the benchmark functions

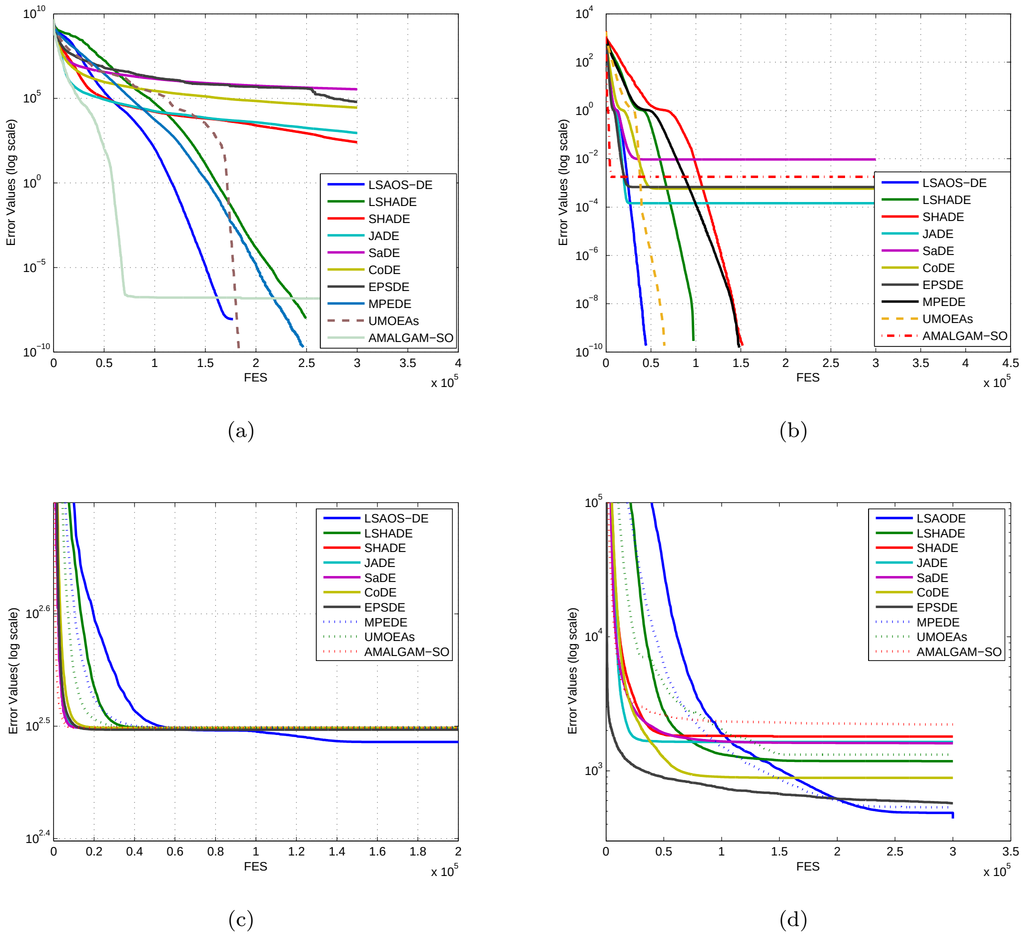


Figure 3.5: Convergence plots of proposed algorithm and state-of-the-art-algorithm for (a) F01, (b) F07, (c) F23 and (d) F30.

$F18$ for CEC2014 and $F31$ for CEC2015, respectively, with D dimensions, and T_2 is the time to execute LSAOS-DE with 200,000 evaluations of both functions in D dimensions. $\hat{T}_2$ is the mean of the T_2 values of 5 runs. Table 3.15 shows the time complexity of LSAOS-DE computed for $10D$, $30D$, and $50D$ in CEC2014 [198] and CEC2015 [199].

The correlation coefficient (R) in Table 3.15 is computed to observe the relationship between dimensions and $\frac{(\hat{T}_2 - T_1)}{T_0}$, which for both CEC2014 and CEC2015, is a strong positive linear one (i.e., $\frac{(\hat{T}_2 - T_1)}{T_0}$ scaled linearly with the number of dimensions).

Table 3.15: Algorithm's complexity

		T_0	T_1	$\hat{T_2}$	$\frac{(\hat{T_2}-T_1)}{T_0}$	R
CEC2014	$D = 10$	0.110568	0.307946	0.926645	5.59565064	0.981004
	$D = 30$		0.288033	1.176284	8.033201288	
	$D = 50$		0.559683	1.943288	12.51798893	
	$D = 100$		2.55944	4.424032	16.86744809	
CEC2015	$D = 10$	0.110568	0.117537	0.932886	7.374186021	0.946601
	$D = 30$		0.382173	1.276088	8.084753274	
	$D = 50$		0.789205	2.218334	12.92534006	
	$D = 100$		2.463938	4.172859	15.45583713	

3.5 Chapter Summary

During the last few decades, several EAs have been introduced to solve unconstrained optimization problems. However, no single EA is able to solve a wide range of these problems. As a result, the concept of multi-method and multi-operator has been emerged to tackle this issue. An AOS mechanisms are used in these designs, to choose the best-performing operator and/or algorithm. Utilizing a problem's landscape to design an efficient AOS mechanism has not been comprehensively explored. In this chapter, a novel multi-operator DE algorithm was designed. In its process, more than one DE mutation strategies with diverse characteristics are used. Also, to put more emphasis on the best-performing operator during the evolutionary process, a problem's landscape information is adopted within an AOS.

The performance of the proposed algorithms were tested by solving 45 bound constrained numerical optimization problems, taken from CEC2014 and CEC2015. LSAOS-DE was compared to seven of its variants, with the results showing that it was better in the quality of solutions obtained.

Then, based on the quality of solutions obtained, the results from LSAOS-DE were compared with those from existing state-of-the-art algorithms such as three well-known DE-based single operator algorithms (LSHADE, SHADE, and JADE), four multi-operator-based algorithms (CoDE, SaDE, EPSDE, and MPEDE); and two multi-method based algorithms (UMOEAs and AMALGAM-SO). The experimental results show that the proposed algorithm is able to obtain better solutions, and its overall performance outperforms the state-of-the-art algorithms.

Motivated by the encouraging results presented in this chapter on unconstrained optimization problems, the next chapter attempts to also use the landscape information to efficiently solve COPs.

Chapter 4

Landscape-based Algorithm for Constrained Problems

In this chapter, a modified landscape measure is used in AOS to select the best-performing DE mutation strategy from a pool of ones to solve constrained optimization problems (COPs). After a brief introduction, this chapter provides a description of the used benchmark problems. A proposed framework and the modified landscape method is presented. After that, an analysis of algorithm components is provided. Finally comparisons with state-of-art algorithms are provided.

4.1 Introduction

Constrained optimization problems play an important role in many scientific areas: such as computer science and operations research. As stated in chapter 2, the solution of COPs using EAs requires the satisfaction of all of a problem's constraints. To do this, many constrained handling techniques (CHTs) have been proposed and integrated with EAs, as reviewed in Chapter 2.

During the past few decades, many EAs and CHTs have been integrated to solve constrained optimization problems. However, no single algorithm is the best

Part of this work has previously appeared in
1. K. M. Sallam, S. M. Elsayed, R. A. Sarker, and D. L. Essam, "Landscape-based Differential Evolution for Constrained Optimization Problems," submitted, in in Computational Intelligence (SSCI), 2017 IEEE Symposium Series on. IEEE, 2017.

for all kind of problems. As stated in the previous chapters, researchers and practitioners have developed many different methods that use different algorithms or search operators in a single framework. In Chapter 3, landscape measures were used in an adaptive operator selection mechanism, to put emphasis on the best-performing DE operators from a pool of operators. The results were very promising in solving unconstrained optimization problems.

Motivated by the encouraging results obtained in chapter 3, for solving unconstrained optimization problem, in this chapter two landscape measures are modified to deal with COPs. This is because landscape measures were originally designed for unconstrained optimization problems. Moreover, a multi-operator DE algorithm is proposed, in which these modified landscape measures are used to put more emphasis on the best-performing operator, with the aim of solving a wide range of COPs.

The performance of the proposed algorithm is judged by solving a set of constrained optimization problems, 24 from CEC2006 [207], 36 test problems from CEC2010 [186] (18 with $10D$ and 18 with $30D$) and 10 problems from CEC2011 [181]. Several experiments were designed and carried out to analyze the effects of different parameters on its performance, and the results from the final variant of the algorithm is compared with different variants of the same algorithm with different selection criteria. Subsequently, the best variant found after analyzing the algorithm's components, is compared to a number of state-of-the-art algorithms.

This chapter is organized as follows: the next section presents the benchmark problems; the proposed algorithm and the landscape modifications are discussed in Section 3.3; the experimental results and analysis are discussed in Section 3.4; and the chapter summary is presented in Section 3.5.

4.2 Benchmark Problems

To judge the performance of the proposed method, three sets of problems are used. These problems were introduced in the CEC2006, and the CEC2010 special competition sessions on constrained optimization problems, and CEC2011 special completion on real-world problems, they are known as the CEC2006 [207], CEC2010 [186], and CEC2011 benchmark problems.

The CEC2006 benchmark problems [207] contains 24 test problems, as detailed in Table 4.1 presents the details of these 24 test functions. In it $\rho = \frac{NF}{NP}$ is the feasibility ratio, where NF is the number of feasible solutions in the initial population, generated by the test case developers, LI, NI, LE and NE are the number of linear inequality, nonlinear inequality, linear equality and nonlinear equality constraints, respectively, and α is the number of active constraints in the optimal solution. These test problems have different types of objective functions, such as linear, nonlinear, polynomial, quadratic and cubic. They have different numbers of decision variables, D, as seen in column 2 of the table.

CEC2010 contains 36 test problems(18 with 10 and 18 with 30 dimensions), as detailed in Table 4.2. Similar to the CEC2006 test problems, they have different mathematical properties, such as either linear or non-linear objective functions and/or constraints, either equality or inequality constraints and either uni-modal or multi-modal objective functions, with the feasible space possibly very tiny compared with the search space.

Real-application Test Problems: These problems come from CEC2011, in that benchmark, there are a number of real-world engineering problems. This chapter considers the 10 of them that are constrained problems. They have different mathematical characteristics, such as having different types and numbers of constraints

Table 4.1: Details of CEC2006 benchmark constrained test functions

Problem	Variables	$\rho(\%)$	LI	NI	LE	NE	α
g01	13	0.0111	9	0	0	0	6
g02	20	99.9971	0	2	0	0	1
g03	10	0.0000	0	0	0	1	1
g04	5	52.1230	0	6	0	0	2
g05	4	0.0000	2	0	0	3	3
g06	2	0.0066	0	2	0	0	2
g07	10	0.0003	3	5	0	0	6
g08	2	0.8560	0	2	0	0	0
g09	7	0.5121	0	4	0	0	2
g10	8	0.0010	3	3	0	0	6
g11	2	0.0000	0	0	0	1	1
g12	3	4.7713	0	1	0	0	0
g13	5	0.0000	0	0	0	3	3
g14	10	0.0000	0	0	3	0	3
g15	3	0.0000	0	0	1	1	2
g16	5	0.0204	4	34	0	0	4
g17	6	0.0000	0	0	0	4	4
g18	9	0.0000	0	13	0	0	6
g19	15	33.4761	0	5	0	0	0
g21	7	0.0000	0	1	0	5	6
g23	9	0.0000	0	2	3	1	6
g24	2	79.6556	0	2	0	0	2

and variables, being static or dynamic in nature, and of different modalities. rf_1-rf_5, are static economic load dispatch (ELD) ones with dimensions of 6, 13, 15, 40 and 140, respectively, problems rf_6 and rf_7 are dynamic economic dispatch (DED) ones with dimensions of 120 and 216, respectively, and problems rf_8- rf_{10} are Hydrothermal Scheduling Problems, all with 96 dimensions. Detailed descriptions of these problems can be found in [181].

Table 4.2: Details of CEC2010 (N-S indicates a function is non-separable, I-S that a function is separable, I- the number of inequality constraints, E- the number of equality constraints and feasibility ratio is estimated ratio between feasible region and the search space)

Problem	Search range	Objective type	Number of constraints		Feasibility rate	
			E	I	10D	30D
C01	$[0, 10]^D$	N-S	0	2 N-S	0.997689	1.0000
C02	$[5.12, 5.12]^D$	I-S	1 I-S	2 I-S	0.0000	0.0000
C03	$[-1000, 1000]^D$	N-S	1 I-S	0	0.0000	0.0000
C04	$[-50, 50]^D$	I-S	2 N-S, 2 I-S	0	0.0000	0.0000
C05	$[-600, 600]^D$	I-S	2 I-S	0	0.0000	0.0000
C06	$[-600, 600]^D$	I-S	2 Rotated	0	0.0000	0.0000
C07	$[-140, 140]^D$	N-S	0	1 I-S	0.505123	0.503725
C08	$[-140140]^D$	N-S	0	1 Rotated	0.505123	0.503725
C09	$[-500, 500]^D$	N-S	1 I-S	0	0.0000	0.0000
C10	$[-500, 500]^D$	N-S	1 Rotated	0	0.0000	0.0000
C11	$[-100, 100]^D$	Rotated	1 N-S	0	0.0000	0.0000
C12	$[-1000, 1000]^D$	I-S	1 N-S	1 I-S	0.0000	0.0000
C13	$[-500, 500]^D$	I-S	0	2 I-S, 1 N-S	0.0000	0.0000
C14	$[-1000, 1000]^D$	N-S	0	3 I-S	0.003112	0.006123
C15	$[-1000, 1000]^D$	N-S	0	3 Rotated	0.003210	0.006023
C16	$[-10, 10]^D$	N-S	2 I-S	1 I-S, 1 N-S	0.0000	0.0000
C17	$[-10, 10]^D$	N-S	1 I-S	2 N-S	0.0000	0.0000
C18	$[-50, 50]^D$	N-S	1 I-S	1 I-S	0.00001	0.0000

4.3 Modified Landscape Measure Combined with MODE

As previously described, existing multi-operator algorithms use an adaptive operator selection mechanism, which is usually based on the success of generating new offspring. In this section, the MODE algorithm, which uses modified problem's landscape information, to adaptively choose the most suitable DE operator, is proposed to solve a wide range of COPs. This section presents modified landscape measures, the general framework, and it components.

4.3.1 Modified landscape measures

In the previous chapters, the Information Landscape Negative Searchability (ILNS) measure was used in an AOS, to put more emphasis on the best-performing operator, from a pool of m operators, and was used to solve unconstrained optimization problems. Dealing with constrained optimization problems is different from unconstrained ones, as all the functional constraints must be satisfied. Therefore, ILNS must be modified to be able to deal with COPs.

4.3.1.1 Modified ILNS (MILNS)

This section proposes a modified ILNS.

First, an information matrix $M = [a_{i,j}]$ for a minimization problem is constructed using:

$$a_{ij} = \begin{cases} 1 & \text{if } (\psi(x_i) = \psi(x_j) = 0 \text{ and } f(x_i) < f(x_j)) \text{ or} \\ & (\text{if } (\psi(x_i) \neq 0 \text{ and } \psi(x_j) \neq 0 \text{ and } \psi(x_i) < \psi(x_j)) \text{ or} \\ & (\psi(x_i) = 0 \text{ and } \psi(x_j) \neq 0 \\ \\ 0 & \text{if } (\psi(x_i) = \psi(x_j) = 0 \text{ and } f(x_i) > f(x_j)) \text{ or} \\ & (\text{if } (\psi(x_i) \neq 0 \text{ and } \psi(x_j) \neq 0 \text{ and } \psi(x_i) > \psi(x_j)) \text{ or} \\ & \psi(x_i) \neq 0 \text{ and } \psi(x_j) = 0 \\ \\ 0.5 & \text{otherwise} \end{cases} \tag{4.1}$$

where $\psi(x_i)$ is the total degree of constrain violations for individual x_i. To construct the information landscape, as described in the previous chapter, not all entries in the information matrix M are necessary. As, there is a duplicate in the entries due to symmetry (the lower triangle should be omitted), the entries on the diagonal are always 0.5 (and should be omitted), and the row and column of the optimum solution should be omitted. So the information matrix can be reduced to a vector $LS = (ls1, ls_2, ..., ls_{|LS|})$, where the number of elements in LS, $|LS| = \frac{(NP-1)\times(NP-2)}{2}$, and NP is the population size.

Here, an example is given to understand how LS is constructed for COPs. Consider the following COP:

$$\begin{aligned} \text{Minimize } f(x,y) &= -(x-10)^3 + (y-20)^3 \qquad (4.2) \\ \text{subject to: } g_1(x,y) &= -(x-5)^2 - (y-5)^2 + 100 \leq 0 \\ g_2(x,y) &= (x-6)^2 + (y-5)^2 - 82.81 \leq 0 \\ 13 \leq x &\leq 100 \text{ and } 0 \leq y \leq 100 \end{aligned}$$

and the values of (x,y)=$\{(15.9997, 4.1173), (14.4503, 2.1789), (14.4354, 1.4401), (15.0003, 4.3179)\}$, then $f_1 = f(x_1, y_1) = -6321.1$, $f_2 = f(x_2, y_2) = -5571.7$, $f_3 = f(x_3, y_3) = -6310.8$, and $f_4 = f(x_4, y_4) = -3731.7$, $\psi_1 = \psi(x_1, y_1) = 2.6024$, $\psi_2 = \psi(x_2, y_2) = 2.7325$, $\psi_3 = \psi(x_3, y_3) = 0$ and $\psi_4 = \psi(x_4, y_4) = 0$. Then the pairwise matrix M is:

$$M = \begin{array}{c|cccc} & f_1 & f_2 & f_3 & f_4 \\ \hline f_1 & 0.5 & 1 & 0 & 0 \\ f_2 & 0 & 0.5 & 0 & 0 \\ f_3 & 1 & 1 & 0.5 & 1 \\ f_4 & 1 & 1 & 0 & 0.5 \end{array}$$

To construct the vector LS from matrix M, the lower triangular, main diagonal, third row and third column are deleted. So, $LS = (a_{12}, a_{14}, a_{24}) = (1, 0, 0)$. Given two landscape $LS_f = (ls_{f,1}, ls_{f,2}, ..., ls_{f,|LS_f|})$ and $LS_{ref} = (ls_{ref,1}, ls_{ref,2}, ...,$ $ls_{ref,|LS_{ref}|})$, the difference value between the two given landscapes is computed using equation 4.3.

$$LD = \frac{1}{|LS_f|} \times \sum_{z=1}^{|LS_f|} |ls_{f,z} - ls_{ref,z}| \tag{4.3}$$

where z = 1, 2, ..., $|LS_f|$. When LD is close to 0 or 1, the problem is considered easy or difficult, respectively.

4.3.1.2 Modified FDC

In this section, a modified FDC is proposed to deal with COPs. As previously stated in chapter 2, FDC measures the correlation between objective function of $\overrightarrow{x}$ and its distance to its nearest best solution (x^{opt}). As this chapter deals with COPs, the (x^{opt}) is determined based on both the objective value and sum of constraint violations using the feasibility rule, that is between wo individuals $\overrightarrow{x}_1$ and $\overrightarrow{x}_2$, the fittest one is determined by:

1. if $\psi(\overrightarrow{x}_1) = \psi(\overrightarrow{x}_2) = 0$ and $f(\overrightarrow{x}_1) < f(\overrightarrow{x}_2)$, then $\overrightarrow{x}_1$ is better;

2. if $\psi(\overrightarrow{x}_1) = 0$ and $\psi(\overrightarrow{x}_1) \neq 0$, then $\overrightarrow{x}_1$ is better, or if $\psi(\overrightarrow{x}_1) \neq 0$ and $\psi(\overrightarrow{x}_1) = 0$, then $\overrightarrow{x}_2$ is better; and

3. if $\psi(\overrightarrow{x}_1) \neq 0$ and $\psi(\overrightarrow{x}_2) \neq 0$ and $\psi(\overrightarrow{x}_1) < \psi(\overrightarrow{x}_2)$, then $\overrightarrow{x}_1$ is better.

4.3.2 MODE combined with modified landscape for COPs

In this section, details of the proposed modified landscape-based multi-operator DE (MLS-MODE) are discussed.

The general framework of the proposed algorithm is presented in Algorithm 4. It starts with a random initial population of size NP, generated using LHD. Then, each individual is evaluated, and the number of current fitness evaluations (FES) is updated. Subsequently, each DE operator generates the same number of individuals. At the end of each generation, the information landscape metric is calculated for each operator using Algorithm 3. After a predefined number of generations (CS), the best-performing DE operator is selected based on the normalized landscape value (NLD) (see Section 4.3.3.1), and then used to evolve the entire population for the subsequent CS generations. Then, as the performance of one operator may be good at the later generations, all the DE operators are reused to evolve the current entire population. This process is continued until a pre-defined number of fitness evaluations (FEs_{pre}) is reached and then the best-performing operator found so far is used to evolve the entire population until a stopping criterion is reached.

4.3.3 The selection phase

In this section, the selection phase, which uses the the modified landscape to select the best-performing operator, is described.

Algorithm 4 Proposed algorithm

$Counter \leftarrow 0$; Generate an initial population (X) of size NP using Latin Hypercube Design; $FES \leftarrow 0$;
Calculate the fitness values of X, and total constraint violation ψ;
$FES \leftarrow FES + NP$;
while $FES \leq MAX_{FES}$ **do**
  $C \leftarrow C + 1$;
  if $FES \leq FEs_{pre}$ **then**
    $C \leftarrow C + 1$;
    if $C < CS$ **then**
      Randomly assign each operator to ($[\frac{NP}{m}]$) solutions;
      Generate new population using the assigned operators.
      Calculate LD_{op} by considering those individuals updated by operator *op*;
    end if
    if mod$(C, CS == 0)$ **then**
      Measure the normalized landscape value of every m using Equation 4.5;
      Determine the best-performing operator, i.e., the one with the maximum value of $ON_{LD_{op}}$;
    end if
    if $C > CS$ and $C < 2CS$ **then**
      Evolve the current entire population using the best DE mutation strategy;
    end if
    if $C == 2CS$ **then**
      $Counter \leftarrow 0$, reset m, $LD_{op} = [\ ]$;
    end if
  else
    Evolve the population using the best DE operator;
  end if
  $FES \leftarrow FES + NP$;
  Update NP using equation 3.1;
end while

4.3.3.1 Normalized average Landscape measure

In this section, the overall normalized average landscape measure ($ON_{LD_{op}}$) of each operator *op* is computed, by utilizing the MILNS measure (see Section 4.3.1.1).

The normalized value $N_{LD_{op}}$ metrics of operator *op* is calculated using by:

$$N_{LD_{op}} = \frac{1 - M_{LD_{op}}}{\sum_{OP=1}^{m}(1 - M_{LD_{op}})} \tag{4.4}$$

where $\{op = 1, 2, ..., m\}$, M_{LD} is the mean value of the landscape information measure.

Subsequently, the overall normalized performance $ON_{LD_{op}}$ of operator op is computed by equation 4.5:

$$ON_{LD_{op}} = \frac{N_{LD_{op}}}{\sum_{OP=1}^{m}(N_{LD_{op}})} \tag{4.5}$$

4.3.4 Adaptation of F and Cr

As previously mentioned, a DE's performance depends on its search operators and control parameters. However, selecting them is not an easy task. Therefore, a self-adaptive mechanism is used in this chapter [67, 208].

A historical memory of length H for both F and Cr is used. The parameter values in this historical memory are denoted as μ_F and μ_{Cr}, and initially were set to 0.5. Each individual $\overrightarrow{x}_z$ is associated with its own (F_z) and (Cr_z), and their values are created using the following equations:

$$Cr_z = randni(\mu_{Cr,r_z}, 0.1) \tag{4.6}$$

$$F_z = randci(\mu_{F,r_z}, 0.1) \tag{4.7}$$

where r_z is randomly selected from $[1, H]$, $randni$ and $randci$ are randomly selected from normal and Cauchy distributions with mean μ_{Cr,r_z} and μ_{F,r_z} respectively, with variance 0.1. A repair mechanism is used to handle any values of Cr_z and F_z, if their values are outside [0,1]. They are repaired as follows. If Cr_z is out of the range, it is replaced by the limit value (0 or 1) closest to the generated value. If $F_z > 1$, it is replaced by 1, and if $F_z \leq 0$, equation 4.7 is repeatedly executed until a valid value is generated.

At the end of each generation, the (Cr_z) and (F_z) used by the successful individuals are recorded in S_{Cr} and S_F, and then the contents of the historical memory are updated as follows:

$$\mu_{CR,h,G+1} = \begin{cases} meanw_L(S_{Cr}) & \text{if } SCR \neq \phi \\ \mu_{CR,h,G} & otherwise \end{cases} \tag{4.8}$$

$$\mu_{F,h,G+1} = \begin{cases} meanw_L(S_F) & \text{if } SF \neq \phi \\ \mu_{F,h,G} & otherwise \end{cases} \tag{4.9}$$

$$meanw_L(SF) = \frac{\sum\limits_{h=1}^{|SF|} \omega_h.S_{F_h}^2}{\sum\limits_{h=1}^{|SF|} \omega_h.S_{F_h}} \tag{4.10}$$

where ω_h is the weight computed using:

$$\omega_h = \frac{\beta_h}{\sum_{h=1}^{|S_{Cr}|} \beta_h} \tag{4.11}$$

where $1 \leq h \leq H$ is the position of the memory to update. It is initialized with a value of 1 and is consequently incremented whenever a new element is inserted into the history. If $h > H$, h is reset to 1. The value of β_h is computed, based on one of the following cases [208]:

1. Infeasible to infeasible: the best solution in the population is infeasible in both generations $G-1$ and G;

2. Infeasible to feasible: the best solution in the population is infeasible at generation $G-1$ and becomes feasible at G; and

3. Feasible to feasible: the best solution is feasible at both generations $G-1$ and G.

Firstly, for every successful solution $(h \in 1, 2, ..., |S_{Cr}|)^2$ which falls in case 1, its β_h is computed as:

$$\beta_h = I_h = max(0, \frac{\psi_{h,G-1} - \psi_{h,G}}{\psi_{h,G-1}}) + max(0, \frac{f_{h,G-1} - f_{h,G}}{f_{h,G-1}}) \tag{4.12}$$

Then, for every successful solution $(h \in 1, 2, ..., |S_{Cr}|)^2$ which exist in case 2 or 3, its β_h is calculated as:

$$\beta_h = max(0,\ I_h) + \frac{\psi_{h,G-1} - \psi_{h,G}}{\psi_{h,G-1}} + max(0, \frac{f_{h,G-1} - f_{h,G}}{f_{h,G-1}}) \tag{4.13}$$

4.3.5 Constraints handling

In this method, the feasibility rule is used to select between any individual and its parent, as: 1) from two feasible solutions, the one with the lowest fitness value is selected; 2) from two infeasible solutions, the one with the smallest sum of constraint violation (ψ) is chosen, where ψ is calculated using Equation 4.14; and 3) a feasible solution as always better than infeasible one.

$$\psi(\overrightarrow{x}_z) = \sum_{k=1}^{s} max(0, g_k(\overrightarrow{x}_z)) + \sum_{e=1}^{q} max(0, |h_e(\overrightarrow{x}_z)| - \delta_e) \tag{4.14}$$

where $g_k(\overrightarrow{x}_z)$ and $h_e(\overrightarrow{x}_z)$ are the k^{th} inequality and e^{th} equality constraints, respectively. For every equality constraint h_e, δ_e is initialized with a large value and then reduced gradualy to 0.0001, and its initial value is problem dependent [208–210].

4.4 Experimental Setup and Results

In this section, the performance of the proposed algorithm is tested by solving the set of benchmark problems presented in Section 4.2. For more comprehensive comparisons, each comparative algorithm was run 25 times for all the optimization functions to a maximum number of fitness function evaluations of 200000 for CEC2006, $20000 \times D$ for CEC2010 and 150000 for CEC2011, with the best, mean and standard deviation results recorded.

Considering parameters settings, the CS parameter was set to 25, φ to 0.5 for DE/φbest/1 to maintain diversity, and 0.1 for all other variants, to speed up the convergence rate. The archive rate (A) to 1.4, the memory size (H) to 5, and FEs_{pre}, which was the limit for running the multi-operator, to $\frac{1}{2} \times MAX_{FES}$, after which the best-performing operator evolved the population until the end of the run. A linear population size reduction mechanism [67] is used as previously described in the previous chapter in which NP^{init} set at a value of 180 and NP^{min} was set at a value of 4.

In the following sections, the effect of the parameters and each component of the proposed algorithm, is examined. After which,the comparison between the best variants of the proposed algorithm and the other state-of-the-art algorithms is conducted.

Table 4.3: Computational time and average FEs of different variants for adapting F and Cr

Algorithms	Total average time	Total average FEs
V1	2.65	163861
V2	1.93	123725
V3	**1.07**	**67066**

4.4.1 Parameter analysis

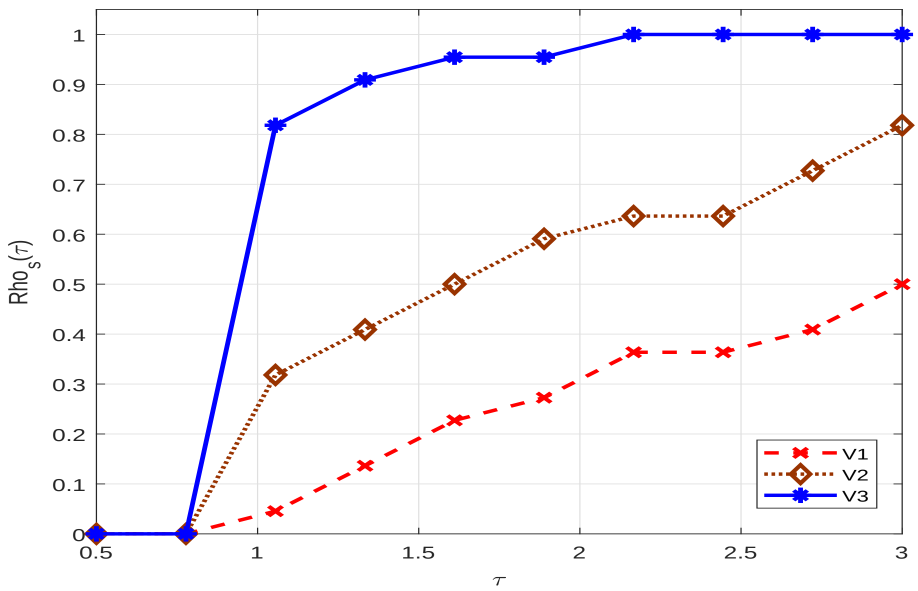
In this section, the effect of the following parameters: 1) the effect of different adaptation rates of F and Cr; 2) the effect of cycle (CS); 3) the effect of linear reduction; 4) the effect of the NP^{init} that are used as the maximum population size; 5) the effect of NP^{min} that is the minimum population size; and 6) the effect of using landscape method in the selection method; on the performance of the proposed algorithm are examined.

4.4.1.1 Effect of Self-adaptation

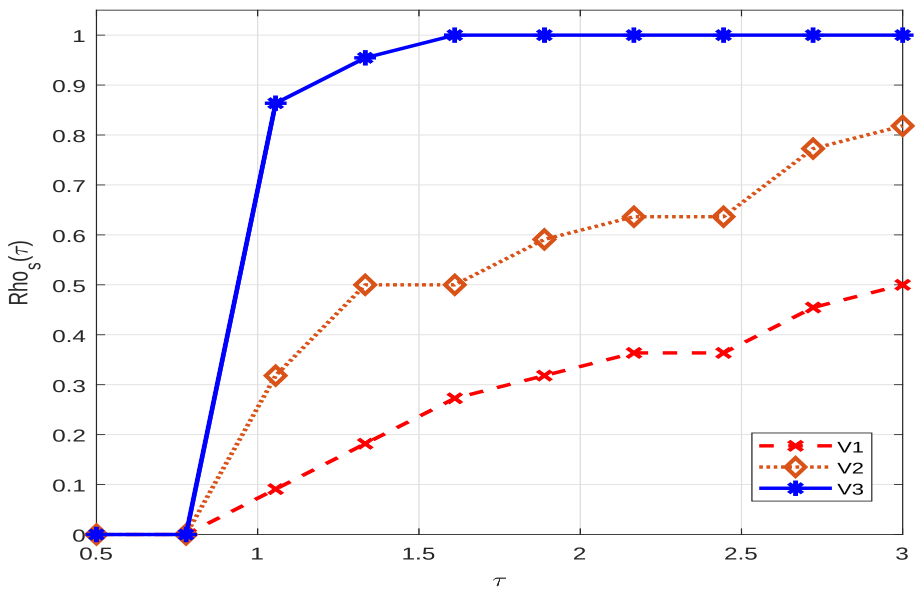
In this section, to see the effect of the self-adaptation strategy used in this chapter, two variants of the proposed algorithm, with two different mutation and crossover rates, are considered. Their results are compared with those obtained from the proposed algorithm with the self-adaptive mechanism. V1 uses $F = 0.9$ and $Cr = 0.1$, while V2 uses $F = 0.5$ and $Cr = 0.5$ uses, and V3 uses the self-adaptive mechanism described in Section 4.3.4. The best, average and standard deviation results, obtained from 25 runs for each variant, are presented in Appendix B, Table B.1.

When comparing the computational time and FEs of the different variants, V3 which uses the self-adaptive approach, is the best. This is shown in Table 4.3.

Also, the performance profiles for both computational time and FEs are depicted in Figure 4.1. From these graphs, the variant with self-adaptation is the



(a)



(b)

Figure 4.1: Comparison of performance profiles of different variants for adapting F and Cr (a) based on computational time; and (b) based on FEs

best, for both average computational time and average FEs.

Regarding the quality of solutions, Table 4.4 shows a comparison summary

Table 4.4: Comparison summary between different variants for adapting F and Cr

Variants	Best Results					Average results				
	Better	Equal	Worse	Dec.	P-value	Better	Equal	Worse	Dec.	P-value
V3 vs. V1	13	8	1	+	0.005	15	6	1	+	0.003
V3 vs. V2	10	11	1	+	0.049	11	10	1	+	0.001

Table 4.5: Computational time and average FEs of different CS values

Variants	Total average time	Total average FEs
$CS = 25$	1.065	67066
$CS = 50$	1.109	68570
$CS = 75$	0.978	68407
$CS = 100$	0.996	66988
$CS = 125$	**0.963**	**66074**
$CS = 150$	0.981	66469

between V3 and the other two variants (V1 and V2). From the results, the performance of V3 with self-adapting F and Cr is better than the two other variants for both the best and average results obtained. In regards to the Wilcoxon test, V3 is statistically better than both V1 and V2 for the both best and average results.

4.4.1.2 Effect of CS

In this section, MLS-MODE is tested while using the self-adaptation of F and Cr, as it was better than both of the fixed values, but with different values of CS. CS represents the number of generations at which MLS-MODE selects the best-performing operator. A number of experiments are conducted by using $CS = 25, 50, 75, 100, 125$ and 150 generations, with detailed results presented in Appendix B, Tables B.2, B.3.

Table 4.5 presents the total average time and total average number of fitness evaluations. From the results, the variant with $CS = 125$ is slightly better than all other variants in both average time and FEs. Also, the overall rank based on the Friedman test is presented in Table 4.6, from which it is clear, that the variant with $CS = 125$ has the best rank, for both the best and average results.

Table 4.6: Ranks of different values for CS, based on Friedman rank test, based on both best and average results

Variants	**Average rank based on best results obtained**	**Average rank based on average results obtained**
$CS = 25$	3.52	3.48
$CS = 50$	3.61	3.70
$CS = 75$	3.48	3.61
$CS = 100$	3.57	3.48
$CS = 125$	**3.39**	**3.25**
$CS = 150$	3.43	3.48

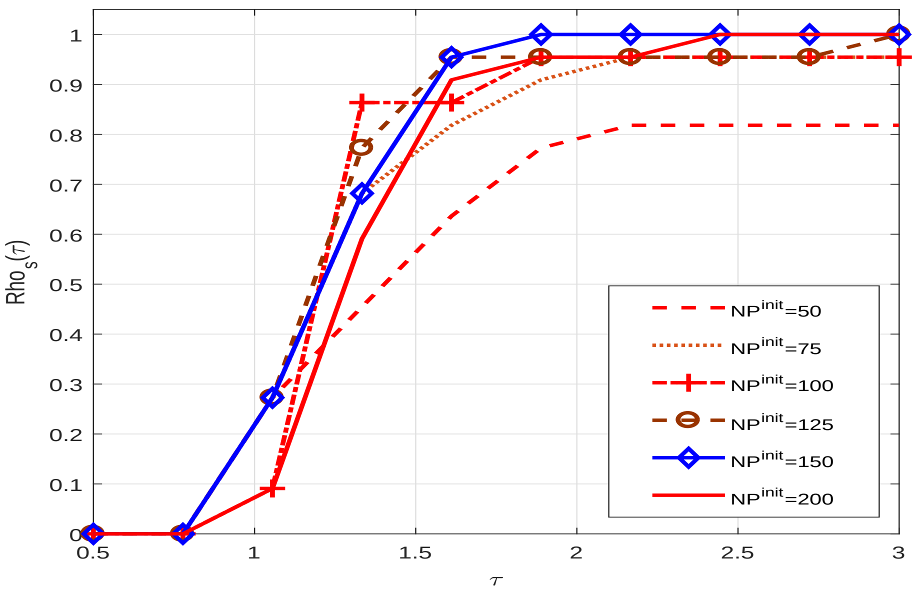
Table 4.7: Computational time and average FEs of different NP^{init} values

Variants	**Total average time**	**Total average FEs**
$NP^{init} = 50$	1.513	51912
$NP^{init} = 75$	1.068	57134
$NP^{init} = 100$	0.977	55279
$NP^{init} = 125$	0.942	61032
$NP^{init} = 150$	**0.913**	**50475**
$NP^{init} = 200$	0.963	68297

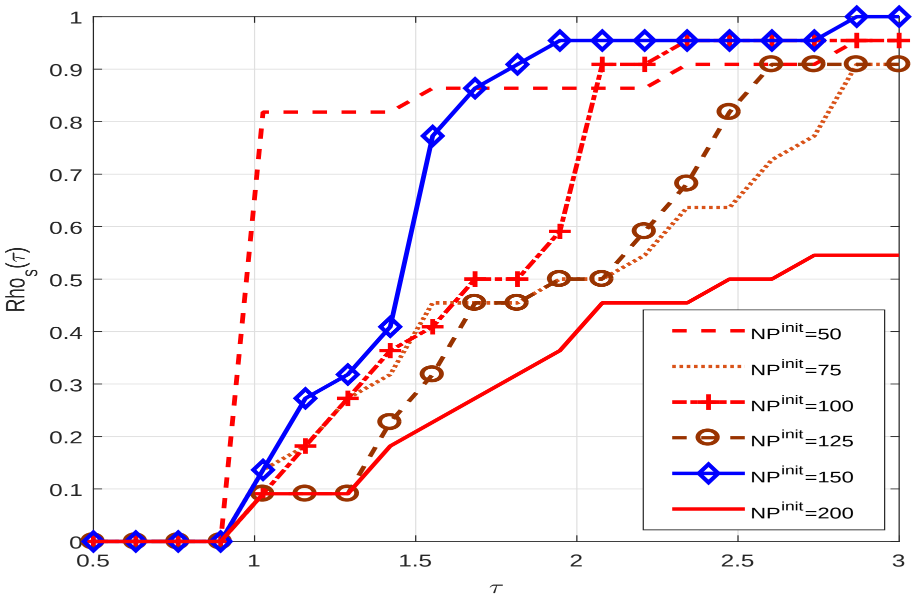
4.4.1.3 Effect of NP^{init}

To see the effect of the initial population size NP^{init}, different experiments with $NP^{init} = 50, 75, 100, 125, 150$ and 200 were conducted. The detailed results are presented in the Appendix B, Tables B.4, B.5. The total average time and FEs are presented in Tables 4.7, from which it is clear that the variant with $NP = 150D$ is slightly better. This is proofed by the performance profiles graphs depicted in Figure 4.2, from which it is clear that the variant with $NP^{init} = 150$ is the best for both average computational time and FEs.

Regarding the quality of solutions, Table 4.8 shows the summary of the obtained results, based on the average results. It is clear that $NP = 150D$ was better than all of the other variants. Regarding the Wilcoxon test, there is no significant difference between the variants. However, there is a bias towards the variant of $NP = 150D$.



(a)



(b)

Figure 4.2: Comparison of performance profiles of proposed algorithm with different values of NP^{init} different variants for adapting F and Cr (a) based on computational time; and (b) based on FEs

4.4.1.4 Effect of NP^{min}

For this analysis, the proposed algorithm was run with different NP^{min} values, such that $NP^{min} = 4, 10, 20, 30, 40$ and 50 individuals. The detailed results are

Table 4.8: Comparison summary between different variants of the proposed algorithm with different NP^{init} values

Variants	Better	Equal	Worse	Dec.
$NP^{init} = 150$ vs. $NP^{init} = 50$	5	16	1	$\approx$
$NP^{init} = 150$ vs. $NP^{init} = 75$	4	17	1	$\approx$
$NP^{init} = 150$ vs. $NP^{init} = 100$	4	17	1	$\approx$
$NP^{init} = 150$ vs. $NP^{init} = 125$	4	17	1	$\approx$
$NP^{init} = 150$ vs. $NP^{init} = 200$	3	17	2	$\approx$

Table 4.9: Computational time and average FEs of different NP^{min} values

Variants	Total average time	Total average FE
$NP^{min} = 4$	1.038	**55515**
$NP^{min} = 10$	1.176	57173
$NP^{min} = 20$	0.961	56496
$NP^{min} = 30$	0.936	57678
$NP^{min} = 40$	**0.880**	56384
$NP^{min} = 50$	0.935	57240

Table 4.10: Average ranks of proposed algorithm with different NP^{min} values

Variants	Mean rank
$NP^{min} = 4$	3.73
$NP^{min} = 10$	3.77
$NP^{min} = 20$	3.73
$NP^{min} = 30$	3.32
$NP^{min} = 40$	**3.07**
$NP^{min} = 50$	3.39

shown in Appendix B, Tables B.6, B.7. The average total computational time and FEs is presented in Table 4.9, which shows that the variant with $NP^{min} = 40$ requires the least average computational time, and the second least in average FEs.

A Friedman test was carried out to rank all the variants. From the results shown in Table 4.10, it was found that the variant with $NP^{min} = 40$ was ranked first, while the one with $NP^{min} = 30$ and $NP^{min} = 50$ came second and third, respectively.

Table 4.11: Comparison summary between different variants of the proposed algorithm with different NP^{min} values

Variants	Better	Equal	Worse	Dec.
$NP^{min} = 40$ vs. $NP^{min} = 4$	**5**	**17**	**0**	+
$NP^{min} = 40$ vs. $NP^{min} = 10$	**5**	**17**	**0**	+
$NP^{min} = 40$ vs. $NP^{min} = 20$	4	17	1	≈
$NP^{min} = 40$ vs. $NP^{min} = 30$	4	17	1	≈
$NP^{min} = 40$ vs. $NP^{min} = 50$	4	17	1	≈

As a further step in comparison, a Wilcoxon test was carried out with the summary of its results are presented in Table 4.11. It is clear that, $NP^{min} = 40$ is significantly better than $NP^{min} = 4$ and $NP^{min} = 10$, while there is no significant difference with the others.

4.4.1.5 Comparing MLS-MODE with different selection mechanisms

In this section, the effect of the modified landscape method in the selection mechanism is analyzed. To do this, a comparison of MLS-MODE is compared with 4 of its variants (Q-MODE, D-MODE, QD-MODE and MFDC-MODE), in each of which only the selection mechanisms were different, and below is a description of the selection mechanism in every variant:

- In D-MODE, population diversity is used to select the best-performing DE operator, as:

$$D_{op,G} = \frac{\sum_{i=1}^{NP_{op}} dis(\overrightarrow{x^{op}}_i, \overrightarrow{x^{op}}_b)}{NP_{op}}, \forall op = 1, 2, ..., m \tag{4.15}$$

  where $dis(\overrightarrow{x^{op}}_i, \overrightarrow{x^{op}}_b)$ is the distance between the i^{th} individual and the best individual generated by operator op, NP_{op} is the number of individuals evolved by operator op, $\overrightarrow{x}_b$ the best solution in the individuals evolved by operators op, and m is the number of operators.

Table 4.12: Ranks of 5 different variants, based on Friedman rank test

Algorithm	Mean rank	Order
MLS-MODE	**2.59**	**1**
D-MODE	3.29	5
Q-MODE	3.07	3
QD-MODE	3.09	4
MFDC-MODE	2.96	2

- In Q-MODE, the selection mechanism is based on the improvement rates in objective function values [163], as:

$$Q_{op,G} = \frac{\sum_{z=1}^{NP_{op}} max(0, f_{G+1,z} - f_{G,z})}{\sum_{z=1,op}^{NP_{op}} f_{G,z}}, \forall op = 1, 2, ..., m \quad (4.16)$$

  where $f_{G+1,z}$ and $f_{G,z}$ are the new and old objective FVs, respectively.

- QD-MODE: the selection mechanism is based on both the population diversity and the quality of solutions, as:

$$QD_{op,G} = D_{op,G} + Q_{op,G} \quad (4.17)$$

- MFDC-MODE: the selection mechanism is based on the modified FDC, as previously described in section 4.3.1.2.

The detailed results obtained from these five variants are presented in Appendix B, Tables B.8, B.9, B.10, B.11, B.12.

A Friedman's ranks test was carried out to rank all the variants for all problems in CEC2006 and CEC2011. From the results shown in Table 4.12, it was found that MLS-MODE was ranked first, while MFDC-MODE came second.

Table 4.13: Comparison summary between MLS-MODE, D-MODE, Q-MODE, QD-MODE and MFDC-MODE

Variants	Better	Equal	Worse	Dec.
MLS-MODE vs. D-MODE	**11**	**17**	**4**	$\approx$
MLS-MODE vs. Q-MODE	**11**	**15**	**6**	$\approx$
MLS-MODE vs. QD-MODE	12	16	4	$\approx$
MLS-MODE vs. MFDC-MODE	10	16	6	$\approx$

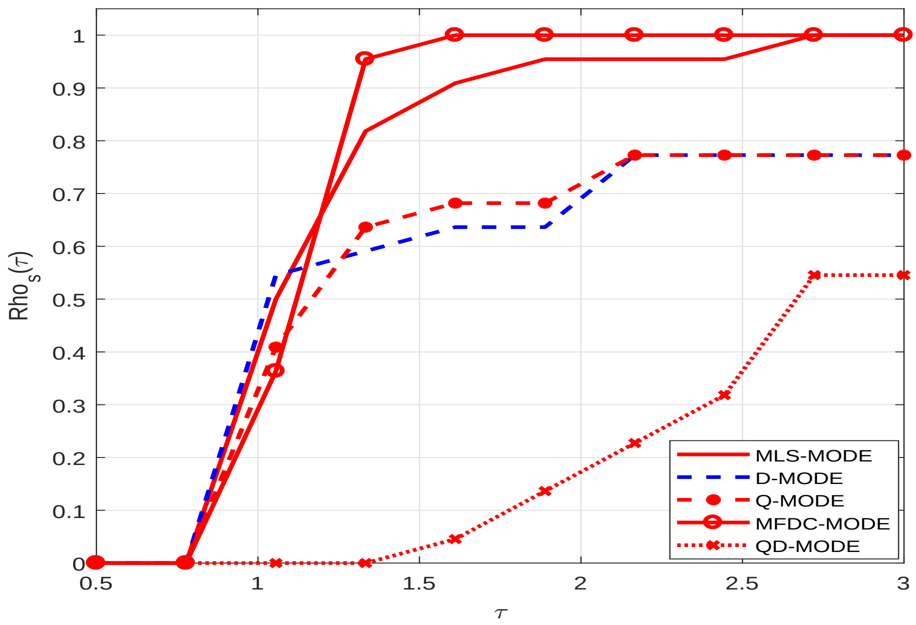
Table 4.14: Comparison among MLS-MODE, Q-MODE, D-MODE, QD-MODE and MFDC-MODE based on the average computational time and FEs

Variants	Total average time	Total average FE
MLS-MODE	**0.999**	**53377**
Q-MODE	1.519	76420
D-MODE	1.530	79659
QD-MODE	3.200	65906
MFDC-MODE	1.209	70939

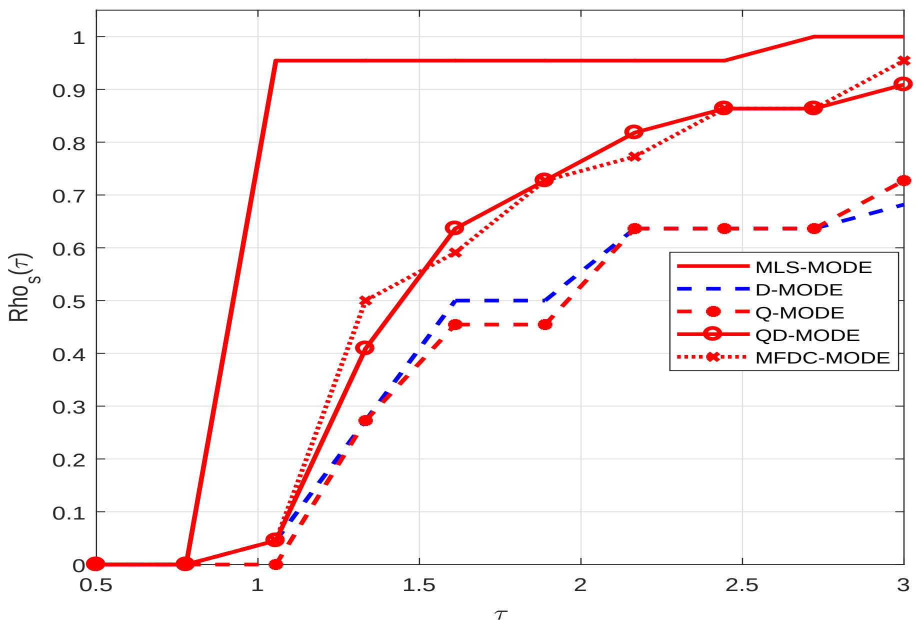
A comparison summary between MLS-MODE and the other four variants is presented in Table 4.13. MLS-MODE is better than D-MODE, Q-MODE, QD-MODE and MFDC-MODE for 11, 11, 12, and 10 test problems respectively, while MLS-MODE is inferior to D-MODE, Q-MODE, QD-MODE and MFDC-MODE in 4, 6, 4 and 6 test problems respectively.

In addition to judging the quality of solution, the average computational time and the number of fitness evaluations required to reach the optimal solution with an error of 0.0001, i.e., the stopping condition is $(f(x) - f(x^*) \leq 0.0001)$, where FEs $f(x^*)$ is the best known solution. Table 4.14 shows the comparison. From this table, it is clear that MLS-MODE saves computational time by 34%, 35%, 69% and 17% in comparison to Q-MODE, D-MODE, QD-MODE and MFDC-MODE, respectively. Regarding FEs, MLS-MODE saves it by 30%, 33%, 19% and 24% in comparison to Q-MODE, D-MODE, QD-MODE and MFDC-MODE, respectively.

Further more, performance profiles for both computational time and FEs are depicted in Figure 4.3. From these figures, the proposed MLS-MODE is the best,



(a)



(b)

Figure 4.3: Performance profiles comparing MLS-MODE, Q-MODE, D-MODE, QD-MODE and MFDC-MODE, based on (a) the average computational time and (b) the average number of FEs.

for both computational time and FEs.

4.4.2 Comparisons with state-of-the-art algorithms

From the above-mentioned analyzes, it was found that, MLS-MODE with $CS = 125$, $NP^{init} = 150$ and $NP^{min} = 40$ is the best variant. So, this variant will be compared to the state-of-the-art algorithms by solving the benchmark problems previously described in Section 4.1.

4.4.2.1 Comparison to the state-of-the-art algorithms for CEC2006

In this section, to verify the performance of the proposed (MLS-MODE) algorithm, comparisons are carried out with: an adaptive hybrid DE algorithm (AH-DEa) [211], a self-adaptive multi-operator genetic algorithm (SAMO-GA) [24], a self-adaptive algorithm with multi-operator strategy (SAMO-DE) [24], an improved version of SAMO-DE (ISAMODE-CMA) [47], the adaptive penalty formulation with GA (APF-GA) [130], an evolutionary programming based on ensemble of constraint-handling techniques ECHT-EP2 [146], a DE based on ensemble of constraint-handling techniques ECHT-DE [146], a rank based multi-operator DE algorithm (rank-iMDDE) [212], ϵDEg with Gradient-Based Mutation ϵDE[141] and an artificial immune system based approach for COPs (AIS-ZYH) [213], which they were used in solving CEC2006 test problems. The detailed results of MLS-MODE, based on 200000 FEs, along with those obtained from the state-of-the-art algorithms are presented in Appendix B, Tables B.13, B.14, B.15, B.16, B.17, B.18. The table shows the mean and standard deviation (Std.) results obtained from 25 runs.

It must be mention here that MLS-MODE used 200000 fitness evaluations (FEs), while ISAMODE-CMA, SAMO-DE, ECHT-EP2, ECHT-DE, SAMO-GA, rank-iMDDE, AH-DEa used 240000 FEs, and APF-GA, MDE, and ϵDEg used 500000 FEs. It should mentioned here that all algorithms solved 22 out 24 test

Table 4.15: Summary of comparisons of the proposed MLS-MODE against AH-DEa, SAMO-GA, SAMO-DE, AIS-ZHY, ECHT-EP2, ECHT-DE, APF-GA, ISAMODE-CMA, ϵDEg, and rank-iMDDE based on the average results, where 'Dec.' statistical decision is based on Wilcoxon signed-rank test results

Algorithms	Better	Equal	Worse	Dec. $\alpha = 0.05$	Dec. $\alpha = 0.10$
MLS-MODE vs. AH-DEa	7	15	0	+	+
MLS-MODE vs. SAMO-GA	11	9	2	+	+
MLS-MODE vs. SAMO-DE	8	14	0	+	+
MLS-MODE vs. AIS-ZHY	6	16	0	+	+
MLS-MODE vs. ECHT-EP2	6	16	0	+	+
MLS-MODE vs. ECHT-DE	6	16	0	+	+
MLS-MODE vs. APF-GA	10	11	1	$\approx$	+
MLS-MODE vs. ISAMODE-CMA	2	20	0	$\approx$	$\approx$
MLS-MODE vs. ϵDEg	1	21	0	$\approx$	$\approx$
MLS-MODE vs. rank-iMDDE	3	19	0	$\approx$	$\approx$

problems. Thus, the analysis is based on 22 test problems. The proposed MLS-MODE algorithm is able to obtain the optimal solutions for all test problems, with feasibility and success rates equal to 100%.

Table 4.15 presents a summary of the comparison between MLS-MODE and the state-of-the-art algorithms. MLS-MODE is better than AH-DEa, SAMO-GA, SAMO-DE, AIS-ZHY, ECHT-EP2, APF-GA, ECHT-DE, ISAMODE-CMA, ϵDE, and rank-iMDDE for 7, 11, 8, 6, 6, 10, 6, 2, 1 and 3 test problems, respectively, while MLS-MODE is inferior to SAMO-GA and APF-GA in 2 and 1 test problems, respectively. Regarding the Wilcoxon test results, MLS-MODE was significantly better than AH-DEa, SAMO-GA, SAMO-DE, AIS-ZHY, ECHT-EP2, ECHT-DE, and APF-GA. Although there is no significant difference between MLS-MODE and ISAMODE-CMA, ϵDEg and rank-iMDDE, there is a bias towards MLS-MODE in the number of better functions. One advantage of MLS-MODE, is its ability to reach optimal solution faster than ISAMODE-CMA, ϵDEg and rank-iMDDE. In summary, the average number of FEs that are consumed by MLS-MODE are 53377, while ISAMODE-CMA, ϵDEg and rank-iMDDE consumed 76420, 79659, and 65,906 FEs respectively, which means that MLS-MODE is able to save 30.15%,

Table 4.16: Average ranking of MLS-MODE, EHCT-DE, AIS-ZHY,ISMOADE-CMA, SAMO-DE, ECHT-EP2, ϵDEg, AH-DEa, SAMO-GA, APF-GA and rank-iMDDE by the Friedman test for the 22 functions in terms of mean value

Algorithm	Mean rank
MLS-MODE	**4.70**
EHCT-DE	6.45
AIS-ZHY	5.84
ISMOADE-CMA	5.14
SAMO-DE	6.59
ECHT-EP2	5.80
ϵDEg	7.75
AH-DEa	6.55
SAMO-GA	7.86
APF-GA	7.07
rank-iMDDE	5.25

32.99%, and 19.01% of FEs in comparison to ISAMODE-CMA, ϵDEg and rank-iMDDE, respectively.

Furthermore, a Friedman test was used to rank all the algorithms according to the mean results obtained, with the mean rank presented in Table 4.16 and Figure 4.4. From Table 4.16, MLS-MODE gets the first rank among the 11 algorithms on 22 test functions.

4.4.2.2 Comparison to the state-of-the-art algorithms for CEC2010

In this section, the proposed MLS-MODE is tested by solving CEC2010 [186] constrained optimization competition. MLS-MODE is compared with the state-of-the-art algorithms: constrained DE with an archive and gradient-based mutation (ϵDEag) [214], which won the CEC2010 COP competition, self-adaptive multi-operator DE (SAMODE) [39], DE combined with DE-DBmax (DE-DBmax) [215], co-evolutionary comprehensive learning particle swarm optimizer (Co-CLPSO) [216], adaptive ranking mutation operator-based DE (ECHT-ARMOR-DE) [179], elitist artificial bee colony (eABC) [217], multi-operator GA (SAMO-GA) [39]

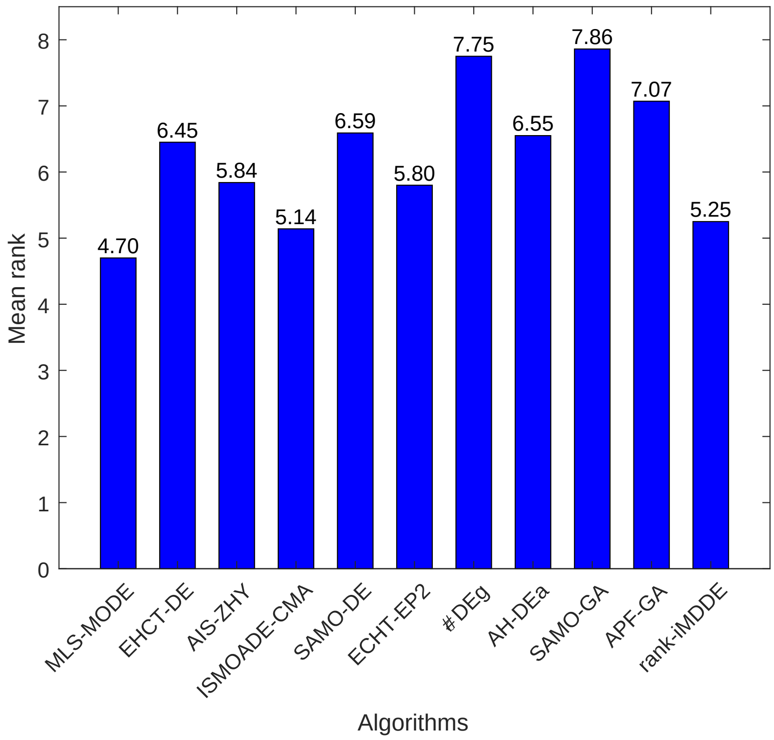


Figure 4.4: Average ranking of MLS-MODE, EHCT-DE, AIS-ZHY,ISMOADE-CMA, SAMO-DE, ECHT-EP2, ϵDEg, AH-DEa, SAMO-GA, APF-GA and rank-iMDDE by the Friedman test for the 22 functions in terms of mean value

and constraint-consensus mutation based DE (DEbavCC) [218]. Detailed results obtained from MLS-MODE and the state-of-the-art algorithms are presented in Appendix B, Tables B.19, B.20, B.21, B.22, B.23, B.24.

MLS-MODE, SAMODE, DEbavDBmax and DE-DBmax were able to achieve a 100% feasibility rate for the 10-D and 30-D test functions, while ϵDEag attained 100% feasibility ratio for only 35 out of 36 test instances, while it only obtained a 12% feasibility ratio for C12 with 30-D. Co-CLPSO was able to obtain a 94.4% feasibility ratio for the 10-D and 87.3% for the 30-D test problems, and the other algorithms were also not able to achieve the 100% feasibility rate

Table 4.17 shows the summary of the quality of solutions obtained. From

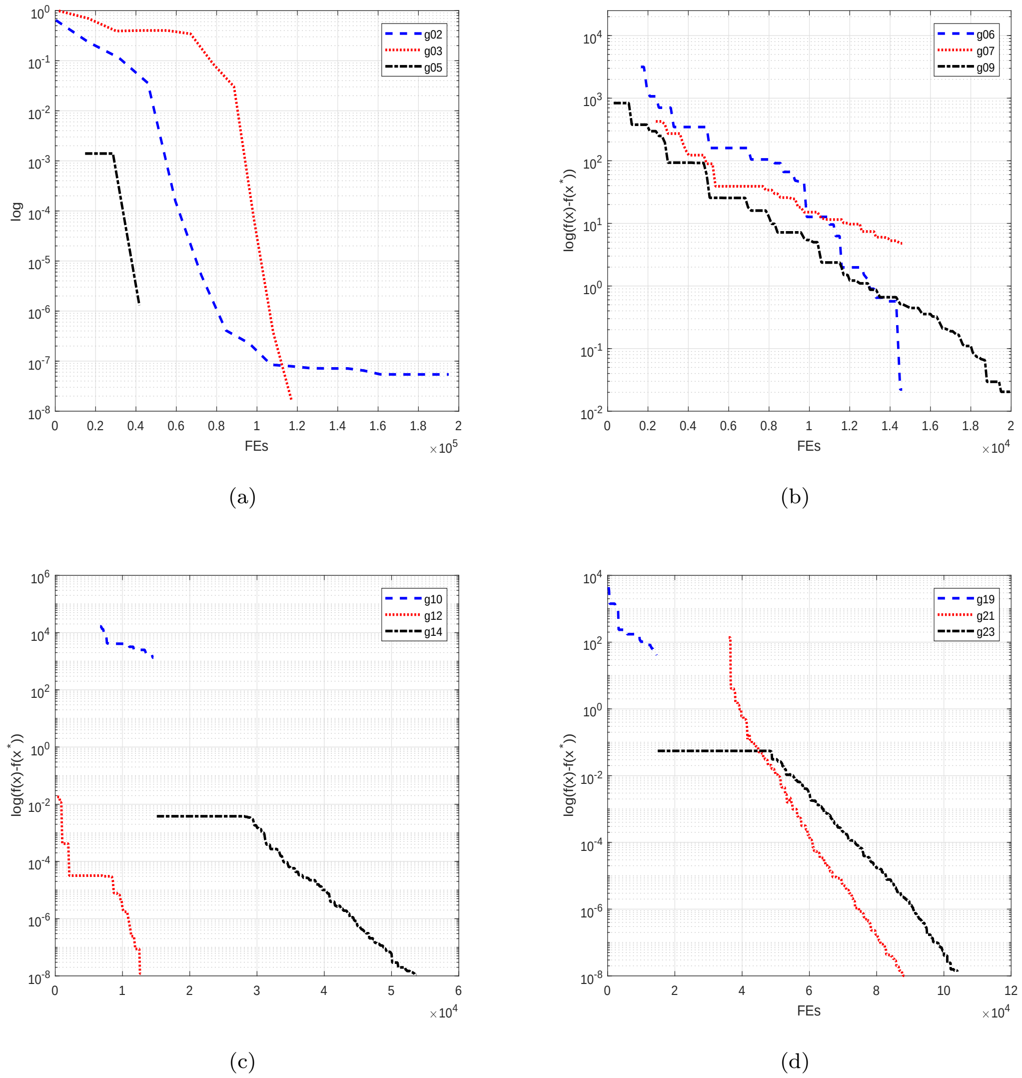


Figure 4.5: Convergence graph for (a) g02, g03 and g05, (b) g06, g07 and g09, (c) g10, g12 and g14 and (d) g19, g21 and g23.

this table, it is clear that MLS-MODE was able to obtain better results for many problems of both 10D and 30D test problems.

Based on the Wilcoxon test results, for 10D test problems, MLS-MODE is significantly better than DEbavDBmax, DE-DBmax, eABC, Co-CLPSO and SAMO-GA for the average results obtained, and better than eABC and Co-CLPSO for the best results obtained. Based on 30D test problems, MLS-MODE was found to be significantly better than all the algorithms for the best results obtained,

Table 4.17: Summary of comparison of MLS-MODE and state-of-the-art algorithms CEC2010

	Algorithms	Criteria	Better	Equal	Worse	Dec.
10D	MLS-MODE vs. DEbavDBmax	Best	3	12	3	≈
		Average	13	5	0	+
	MLS-MODE vs. SAMODE	Best	2	12	4	≈
		Average	12	2	4	≈
	MLS-MODE vs. ϵDEag	Best	5	10	3	≈
		Average	8	6	4	≈
	MLS-MODE vs. DE-DBmax	Best	5	10	3	≈
		Average	16	2	0	+
	MLS-MODE vs. eABC	Best	16	1	1	+
		Average	17	0	1	+
	MLS-MODE vs. Co-CLPSO	Best	14	3	1	+
		Average	16	0	2	+
	MLS-MODE vs. SAMO-GA	Best	10	5	3	≈
		Average	14	0	4	+
	MLS-MODE vs. ECHT-ARMOR-DE	Best	6	10	2	≈
		Average	8	5	5	≈
30D	MLS-MODE vs. DEbavDBmax	Best	7	9	2	≈
		Average	11	3	4	≈
	MLS-MODE vs. SAMODE	Best	13	2	3	+
		Average	12	1	5	≈
	MLS-MODE vs. ϵDEag	Best	15	1	2	+
		Average	15	0	3	+
	MLS-MODE vs. DE-DBmax	Best	8	8	2	≈
		Average	11	3	4	≈
	MLS-MODE vs. eABC	Best	16	0	2	+
		Average	17	0	1	+
	MLS-MODE vs. Co-CLPSO	Best	15	1	2	+
		Average	16	0	2	+
	MLS-MODE vs. SAMO-GA	Best	13	1	4	+
		Average	13	0	5	+
	MLS-MODE vs. ECHT-ARMOR-DE	Best	11	4	3	+
		Average	14	1	3	+

except DEbavDBmax and DE-DBmax which were statistically similar. Regarding the average fitness values, MLS-MODE was statistically better than ϵDEag, eABC, Co-CLPSO, SAMO-GA and ECHT-ARMOR-DE, and statistically similar to DEbavDBmax, SAMODE, and DE-DBmax.

Furthermore, a Friedman test was conducted to rank all algorithms based on the best and average results. Table 4.18 and Figure 4.6 shows the results, from which it is clear that MLS-MODE is ranked first for 10D and 30D.

Table 4.18: Average Ranking achieved by Friedman test for CEC2010

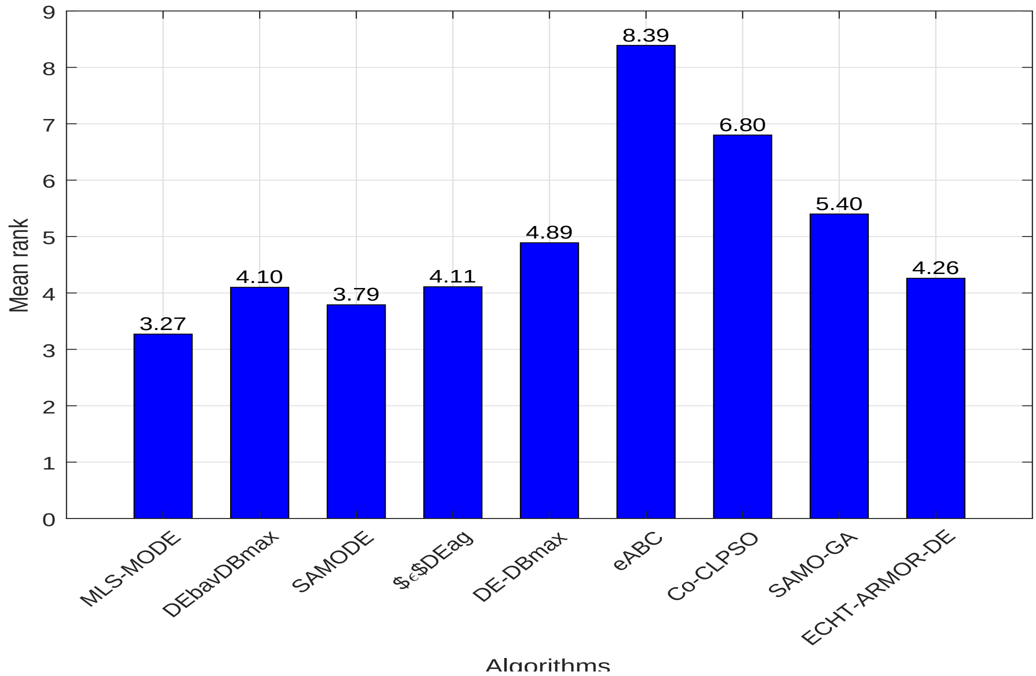
		Average rank based on best results obtained	Average rank based on average results obtained	Over all rank
10D	MLS-MODE	3.86	**2.67**	**3.27**
	DEbavDBmax	3.81	4.39	4.10
	SAMODE	**3.47**	4.11	3.79
	ϵDEag	4.22	4.00	4.11
	DE-DBmax	4.31	5.47	4.89
	eABC	8.36	8.42	8.39
	Co-CLPSO	6.81	6.78	6.80
	SAMO-GA	5.47	5.33	5.40
	ECHT-ARMOR-DE	4.69	3.83	4.26
30D	MLS-MODE	**2.83**	**2.72**	**2.78**
	DEbavDBmax	2.97	2.94	2.96
	SAMODE	4.47	3.81	4.14
	ϵDEag	6.81	5.47	6.14
	DE-DBmax	3.11	3.67	3.39
	eABC	7.94	8.39	8.17
	Co-CLPSO	6.89	7.08	6.99
	SAMO-GA	5.50	5.00	5.25
	ECHT-ARMOR-DE	4.47	5.92	5.20

4.4.2.3 Comparison to the state-of-the-art algorithms for CEC2011

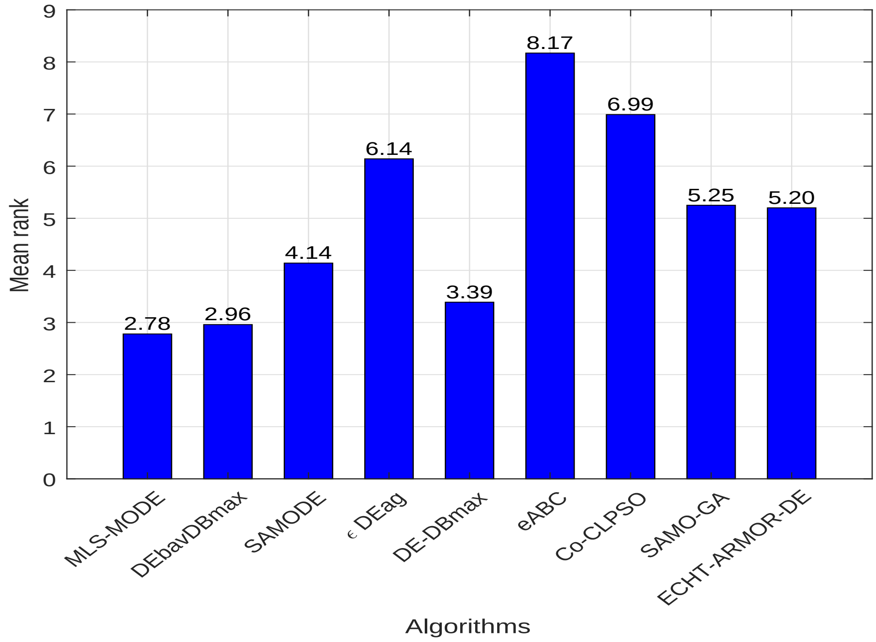
In this section the proposed MLS-MODE is judged by solving 10 real-world application problems taken from the CEC2011 [181] competition on real-world optimization problems.

The performance of the proposed MLS-MODE is compared with the state-of-the-art algorithms, known as: continuous DE ant-stigmergy algorithm (CDASA) [219], an adaptive DE algorithm (ADE) [220], ensemble DE algorithm (EPSDE) [221], SAMODE [23], DE with adaptive crossover rate (DE-Acr) and a competitive DE with local search [222], (CDELS) [223], with their detailed results are presented in Appendix B, Tables B.25, B.26. All these algorithm solve these test problems as unconstrained optimization problems, and there is no information about whether the optimal solution obtained is feasible or not.

In order to compare the performance of the proposed MLS-MODE with the state-of-the-art algorithms, a Friedman test is conducted. Table 4.19 and Figure



(a)



(b)

Figure 4.6: Average Ranking achieved by Friedman test for CEC2010 (a) 10D and (b) 30D.

4.7 shows the average ranking of the seven algorithms. The highest ranking is shown in boldface. As seen, MLS-MODE and DE-Acr obtained the best ranking.

Table 4.19: Average Ranking achieved by Friedman test for CEC2011

Algorithm	Mean rank
MLS-MODE	**2.20**
ADE	4.40
EPSDE	3.70
SAMODE	3.10
DE-Acr	**2.20**
CDELS	6.20
CDASA	6.20

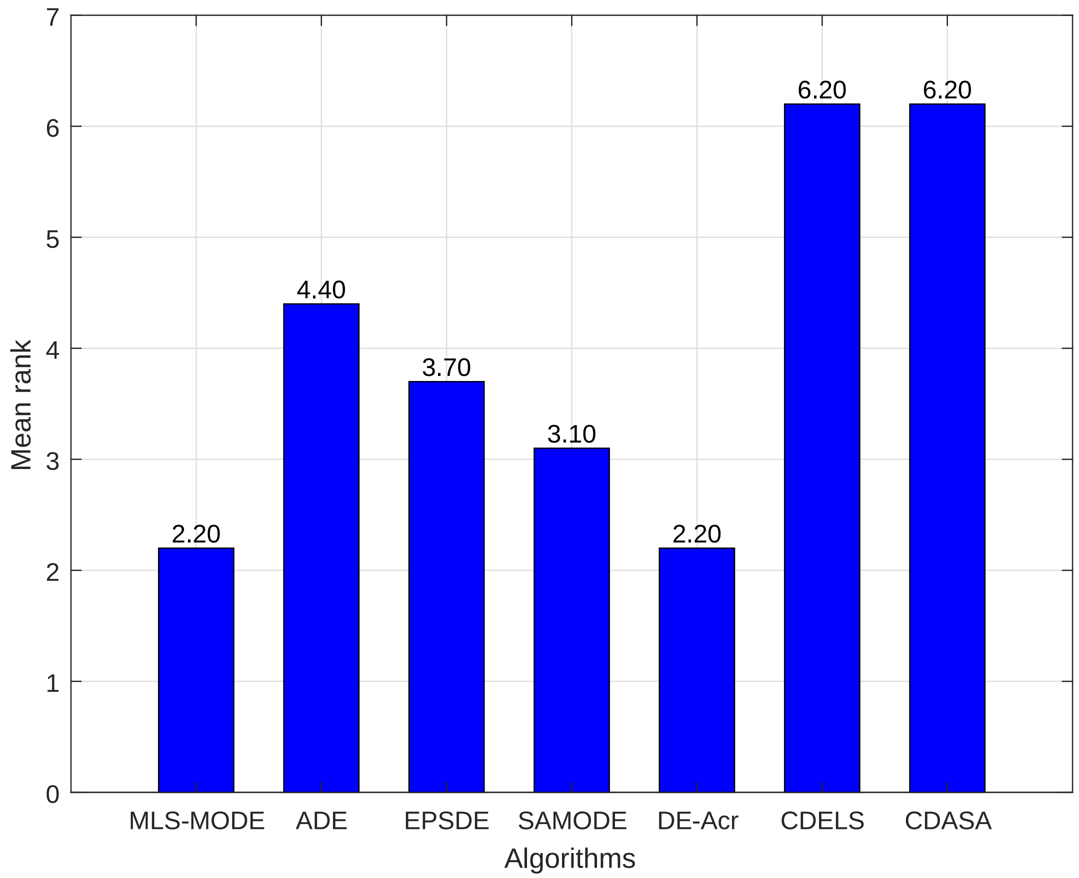


Figure 4.7: Average ranking of MLS-MODE against ADE, EPSDE, SAMODE, DE-ACr, CDELS and CDASA, by the Friedman test for the 10 functions in terms of mean value

Table 4.20: Comparison summary of MLS-MODE against ADE, EPSDE, SAMODE, DE-ACr, CDELS and CDASA for CEC2011

Algorithm	Better	Equal	Worse	p-value	Dec.
MLS-MODE vs. ADE	8	0	2	0.037	+
MLS-MODE vs. EPSDE	8	0	2	0.017	+
MLS-MODE vs. SAMODE	7	0	3	0.203	≈
MLS-MODE vs. DE-Acr	5	0	5	0.575	≈
MLS-MODE vs. CDELS	10	0	0	0.05	+
MLS-MODE vs. CDASA	10	0	0	0.05	+

Regarding the quality of solutions, a summary has been reported in Table 4.20. MLS-MODE was better than ADE, EPSDE, SAMODE, DE-ACr, CDELS and CDASA for 8, 8, 7, 5, 10 and 10 test problems, respectively, While ADE, EPSDE, SAMODE, DE-ACr, CDELS and CDASA are better than MLS-MODE in 2, 2, 3, 5, 0, and 0 test problems, respectively. Based on the Wilcoxon test, it was found that MLS-MODE is significantly better than ADE, EPSDE, CDELS and CDASA, while there is no significance difference with SAMODE and DE-Acr.

4.5 Chapter Summary

Motivated by the promising results obtained from LSAOS-DE in Chapter 3, for solving unconstrained optimization problems, in this chapter, a modified landscape method, called MLS-MODE, is proposed, and is used in an AOS to put more weight to the best-performing DE mutation strategy. MLS-MODE is used to solve constrained optimization problems.

MLS-MODE was tested by solving 22 test problems taken from CEC2006, 36 test problems taken from CEC2010, and 10 test problems taken from CEC2011. MLS-MODE was compared to four of its variants, which all share the same structure with the only difference being in the selection mechanism. The impact of MLS-MODE was significant, as it also consumed 34.0%, 35.0%, 69.0% and 17.0% less computational time than Q-MODE, D-MODE, QD-MODE and

MFDC-MODE, respectively, and regarding to the number of fitness evaluation, MLS-MODE consumed 30.0%, 33.0%, 19.0% and 25.0% less than Q-MODE, D-MODE, QD-MODE and MFDC-MODE, respectively.

The results from MLS-MODE were then compared with those from state-of-the-art algorithms, and based on the quality of solutions obtained and non-parametric statistical testing, the results showed the superiority of MLS-MODE.

In solving COPs some constraints, known as active constraints, are more important than others. As of the optimization literature, the optimal solution often lies at the intersection of active constraints. This has served as motivation to develop a mechanism that may efficiently be able to to find that intersection, or a nearer point, as will be discussed in Chapter 5.

Chapter 5

Reduced Search Space Mechanism for Constrained Problems

In this chapter, a reduced search space mechanism with an active constraint determination procedure is combined with several state-of-the-art algorithms to solve constrained optimization problems (COPs). The experimental results from, and analysis of, the proposed method are presented and discussed by solving benchmark problems. Finally, detailed results obtained from the proposed algorithms and comparisons with those from state-of-the-art algorithms are presented.

5.1 Introduction

In solving COPs, some constraints are more important than others, they are known as active constraints. As of the optimization literature [35–37, 195], the optimal solution often lies at the intersection of active constraints. So the purpose of the search process can be find that intersection, or a nearer point. If the active constraints are known in some way, finding the optimal solution will almost always be easier.

The aim of this chapter is to identify promising reduced search space for COPs. To achieve this objective, new mechanism has been proposed, which identifies the

The following articles have been published based on this Chapter
1. K. M. Sallam, R. A. Sarker, and D. L. Essam, "Reduced search space mechanism for solving constrained optimization problems," Engineering Applications of Artificial Intelligence, vol. 65, pp. 147-158, 2017

promising boundary region, based on an analysis of the individuals in the current population. To do that, a mechanism for automatically identifying the active constraints, based on the current solutions, is proposed. After that, using the information of the active constraints, the promising boundary zone for searching is determined. However, for effective searching using DE, this approach incorporates small regions from both feasible and infeasible spaces around the identified boundary region. As a simple example, suppose the promising boundary is a ring (circle) in a two dimensional search space. Then a smaller and a bigger ring are created for search space, that is the area between the smaller and bigger rings. The distance between the two rings can be varied depending on the nature of the problems and the stage of the evolution process. Note that the promising boundary can also be a part of the ring (an arc) in which the reduced search space (R2S) is created by joining the inner and outer arcs.

In order to evaluate the performance of the proposed R2S mechanism, it is incorporated with a number of state-of-the-art algorithms. The mechanism was tested by solving a set of COPs, 24 from CEC2006 [207], 36 from CEC2010 [186] (18 with $10D$ and 18 with $30D$) and 10 from CEC2011 [181] defined in the previous chapter. Also, several experiments have been designed and carried out to analyze the effects of different parameters on the performance of the proposed algorithm. Moreover, the proposed algorithm is also able to handle the case where the optimal solution lies well inside the feasible region, by using the whole search space. The proposed concept can also easily be applied with other evolutionary search algorithms. This chapter also uses the same test problems as Chapter 4 (Section 4.2).

The chapter is organized as follows: Section 5.2 discusses the proposed framework; the simulation results for the benchmark problems are provided in Section

5.3; and finally, section 5.4 presents the chapter summary.

5.2 Proposed Algorithm

In this section, the proposed active constraints determination (ACD) and R2S mechanisms are described. Then, the general framework of the proposed algorithm that combines R2S with the basic DE algorithm, is discussed.

5.2.1 Active constraint determination (ACD) mechanism

As previously mentioned, some constraints are active at the optimum points in many real-world COPs. According to the definition, at any point $\overrightarrow{x}$, the constraints $g_k(\overrightarrow{x})$ that satisfy $g_k(\overrightarrow{x}) = 0$ are considered active at $\overrightarrow{x}$, as equality constraints are active for all points of decision space. Note that if $\overrightarrow{x}$ is feasible and at least one of the constraints is active at $\overrightarrow{x}$, then $\overrightarrow{x}$ lies on the boundary between the feasible and infeasible regions of the search space. Algorithm 5 describes the main steps of the ACD mechanism, in which a predefined number of fitness evaluations (FES_{ACD}) is spent in every CS (which is the number of generation for evaluating whether or not constraints are active) generations, to check if the value of $g_k(\overrightarrow{x}) \leq Val$ for a randomly chosen individual from the m best individuals. In such a case, $g_k(\overrightarrow{x})$ is an active constraint, otherwise it is not.

Algorithm 5 ACD mechanism

```
Input: iter, cfe, FES_ACD, g_j is the constraint values of individual j;
if mod(iter, CS==0) AND cfe ≤ FES_ACD then
    if g_j ≤ Val then
        constraint j is an active constraint.
    else
        constraint j is not an active constraint.
    end if
end if
```

5.2.2 R2S

As of the literature, equality constraints are always converted to inequality ones using

$$g_k(\overrightarrow{x}) = |h_k(\overrightarrow{x})| - \mathit{\Delta} \leq 0, \text{ for } k = s+1, ..., q \tag{5.1}$$

where $\mathit{\Delta}$ is a small positive tolerance parameter for relaxing the equality constraints. Therefore, the degree of constraint violation of solution $\overrightarrow{x}$ on the j^{th} constraint can be calculated by

$$G_k(\overrightarrow{x}) = max\{0, g_k(\overrightarrow{x})\}, \ k = 1, ..., q \tag{5.2}$$

Then $\psi(\overrightarrow{x}) = \sum_{k=1}^{q} G_k(\overrightarrow{x})$ is the total degree of constraint violation of solution x.

For each individual active constraint, the width or thickness of the boundary area is given by

$$-\delta_{in} \leq g_k(\overrightarrow{x}) \leq \delta_{out} \tag{5.3}$$

where $g_k(\overrightarrow{x})$ is the k^{th} constraint, δ_{in} and δ_{out} are the inner-delta and outer-delta values, respectively.

Equation (5.3) is used to define the new search space, R2S, that is smaller than the search space defined by the variable bounds, and is also different from the feasible space.

δ_{out} is used to include the infeasible regions around the selected boundary. For a problem when its number of feasible solutions is less than $0.2 * NP$ individuals, the initial δ_{out} value $\delta_{out}(0)$ is set to the total constraint violation (ψ) of the top

$0.2 * NP$ individuals in the population, and for other problems, to 1. The δ_{out} value is updated when the number of cfe is less than FES_c , after which it is set to zero to obtain solutions with minimum constraint violations, as:

$$\delta_{out}(t) = \begin{cases} \delta_{out}(0) \times (1 - (\frac{cfe}{FES_c}))^z & 0 < cfe \leq FES_c \\ 0 & cfe > FES_c \end{cases} \tag{5.4}$$

The δ_{out} value is controlled by parameter z, the value of which is updated using

$$z = \frac{-5 - ln(\delta_{out}(0))}{ln(0.05)} \tag{5.5}$$

To avoid a too-small value, a predefined value of $z_{min} = 3$ is assigned and is then controlled as

$$z = 0.3 \times z + 0.7 \times z_{min} \tag{5.6}$$

δ_{in} is used to reduce a feasible region from the inside to be closer to the boundary. $\delta_{in}(0)$ is set to a large value to initially consider all of the feasible region, and is then updated every consecutive generation using the linear reduction mechanism computed by:

$$\delta_{in}(t) = \delta_{in}^{init} - cfe \times \frac{\delta_{in}^{init} - \delta_{in}^{min}}{MAX_{FES}} \tag{5.7}$$

where $\delta_{in}^{init} = abs(max(min(g)))$ is its initial value, $\delta_{in}^{min} = 0.002 \times \delta_{in}^{init}$ its minimum value, and cfe is the current number of fitness evaluations and and MAX_{FES} is the maximum numbers of fitness evaluations.

As previously described, the degree of constraint violation of solution $\overrightarrow{x}$ on the k^{th} constraint is computed using equation 5.2, and the total degree of constraint violation is computed by

$$\psi(\overrightarrow{x}) = \sum_{k=1}^{q} G_k(\overrightarrow{x}). \tag{5.8}$$

In this chapter, if the problem has an active constraint and $\overrightarrow{x}$ satisfies both L and K in equations 6.8 and 6.9, the violation is zero. Otherwise, the total degree of constraint violation is calculated using equations 6.8-6.7.

$$L = -\delta_{in} - g_k(\overrightarrow{x}) \leq 0 \tag{5.9}$$

$$R = g_k(\overrightarrow{x}) - \delta_{out} \leq 0 \tag{5.10}$$

The total degree of constraint violation is calculated as

$$\psi_{R2S}(\overrightarrow{x}) = \sum_{i=1}^{m} max(\,0,\, max(\,L,\, R)) \tag{5.11}$$

By using equation 6.7, any individual which lies inside the space formed by the intersection of the active constraint's boundary region is feasible, and ones outside this region are infeasible. On the other hand, if the problem does not have any active constraints, the total degree of constraint violation is computed using equation 5.8.

Figure 5.1 shows an example of how R2S is constructed, Figure 5.1(a) shows the original search space and feasible region. At the start of the initial evolutionary process, a boundary is constructed for each individual constraint ($g_k(\overrightarrow{x})$) by using equation 5.3, as depicted in Figure 5.1(b), where the shaded area represents the new reduced search space for all constraints, i.e., if any solution, $\overrightarrow{x}_z$, lies inside the shaded region and satisfies all the constraints boundaries, it is feasible and its violation is zero, while if it lies outside, or inside this region and did not satisfy all constraints, it is considered infeasible and its violation is calculated using equation 6.7. After the active constraints are determined by Algorithm 5.1(c), the search space is reduced to cover the boundary regions for only the active constraints, that is, g_1 and g_2 in the given example, with 5.1(c) showing the new reduced search space. Figure 5.1(d) shows the area of interest (intersection of the active constraints), which is the reduced feasible region.

The following example is used to show how the population of individuals is evolved, based on the R2S mechanism.

$$\text{minimize } f(y) = (y_1 - 2)^2 + (y_2 - 1)^2$$

$$\text{subject to: } g_1(y) = y_1 + y_2 - 2 \leq 0$$

$$g_2(y) = y_1^2 - y_2 + 2 \leq 0$$

$$-5 \leq y_1, y_2 \leq 5$$

The optimal solution of this problem is located at $y^* = (0,\, 2)$ with $f(y^*) = 5$. The feasible space of this problem occupies roughly 0.25% of the whole search space and both constraints are active at the optimal solution. Figure **??** shows the population progress of one run. As depicted in Figure 5.2, in Figure 5.2(a) it is observed that the initially created individuals are uniformly distributed in the search space, but none of them is feasible. While in Figure 5.2(b), it is observed that some of the individuals are considered as feasible with respect to the new feasible space constructed by R2S. From generation to generation, the constructed

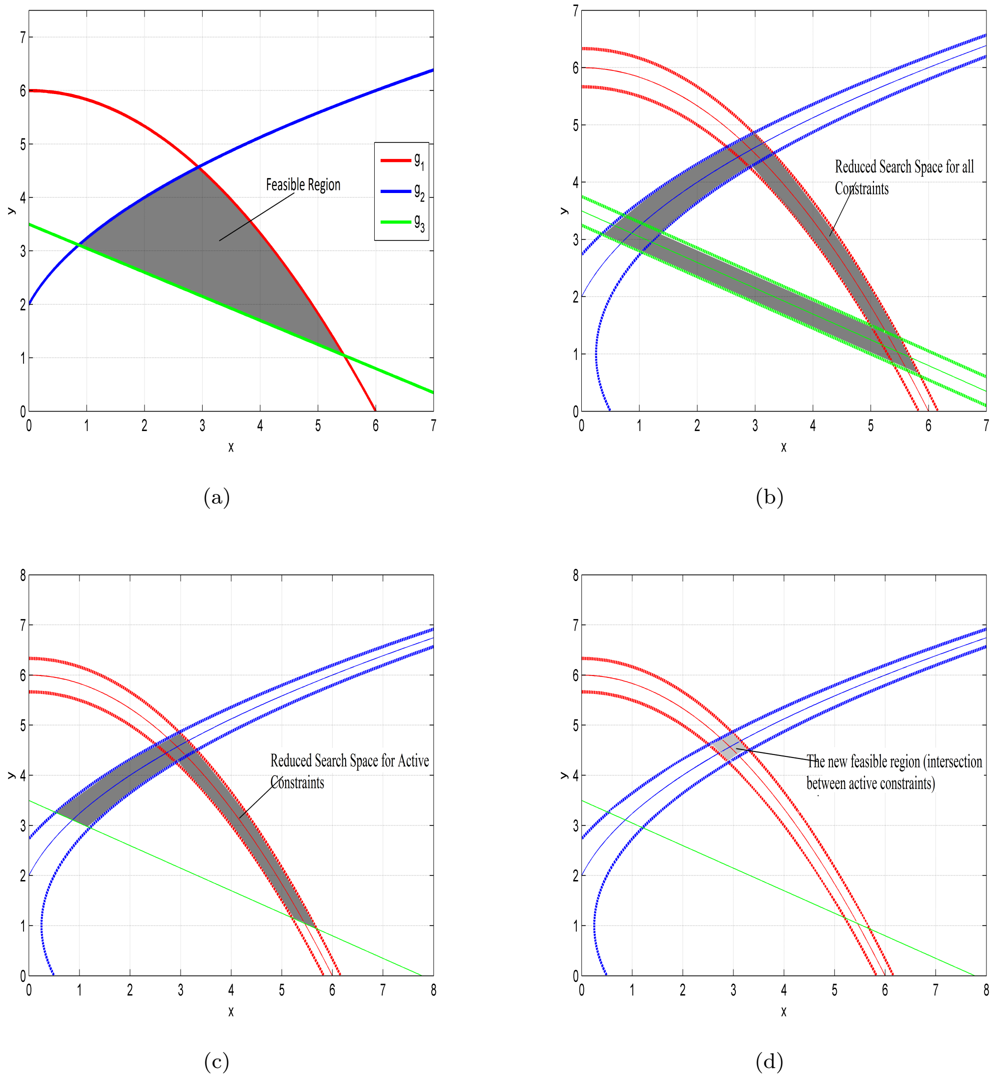


Figure 5.1: Example of Reduced Search Space for (a) Original Search Space, (b) All Constraints, (c) Active Constraints and (d) Feasible region, which is the intersection of the active constraints.

search space is reduced to reach the boundary of the feasible region, as noted in Figures 5.2(c), 5.2(d), 5.2(e), and 5.2(f). It is clear from this graphic illustration, that R2S is very effective in guiding the generation of solutions to the boundary of the feasible region.

To show the effectiveness of the proposed R2S mechanism, a comparison between MJADE, which is different from the original JADE [64] algorithm in only

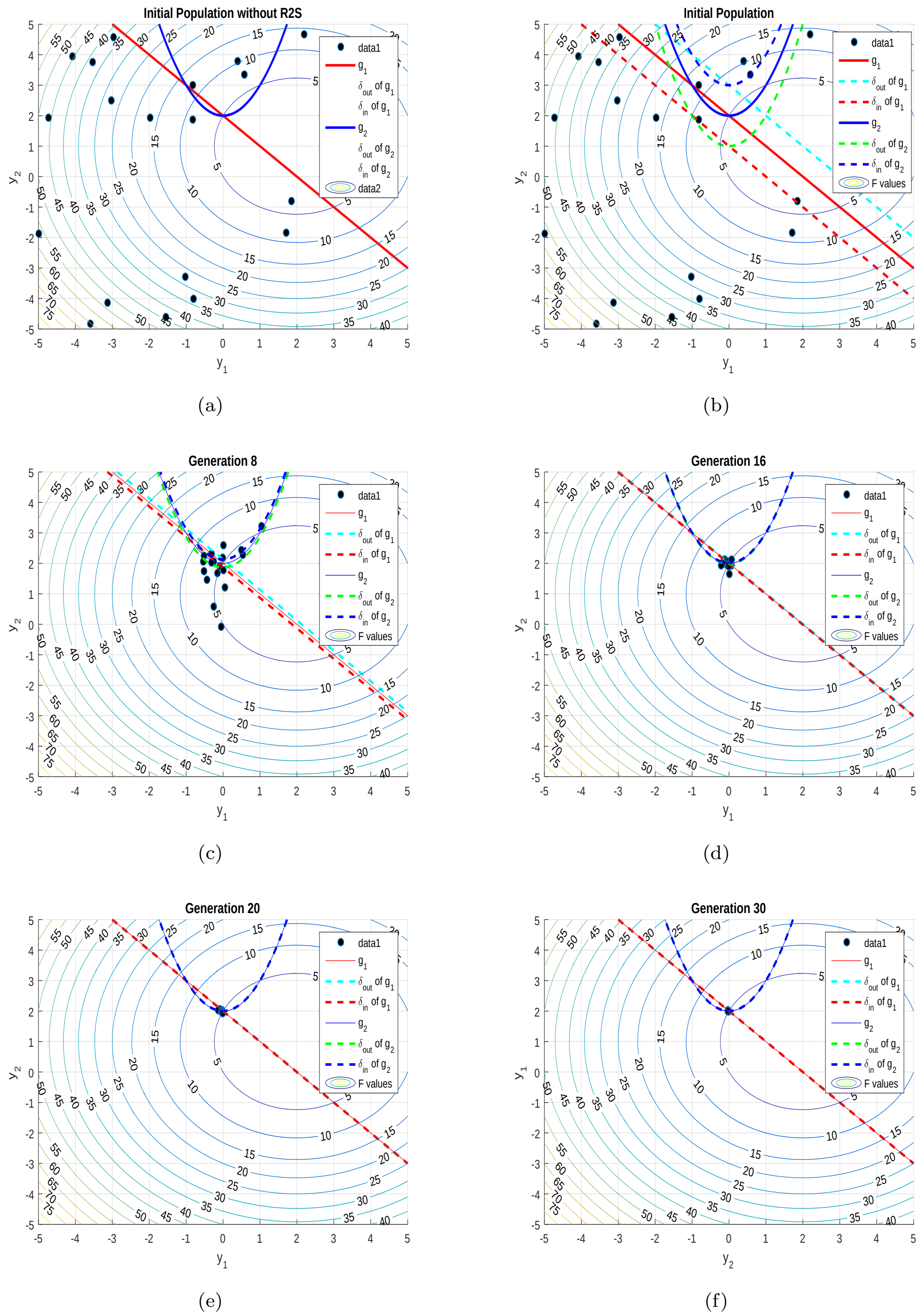


Figure 5.2: The distribution of (a) initial population without R2S mechanism, (b) initial population with R2S mechanism, (c) population at gen 8 with R2S mechanism, (d) population at gen 16 with R2S mechanism (e) population at gen 20 with R2S mechanism and (f) population at gen 30 with R2S mechanism

that a feasibility rule [224] is used to handle the constraint, and MJADE-R2S for g11, which has 2 variables and one equality constraint, are depicted in Figure 5.3. Plots 5.3(a), 5.3(c) and 5.3(e) in Figure 5.3, depict the distribution of population for MJADE without R2S at initialization, and at the 15th, and 40th generation, respectively, while, Plots 5.3(b), 5.3(d) and 5.3(f) in Figure 5.3, depict the distribution of population for MJADE-R2S at initialization, and at the 15th, and 40th generation, respectively. With R2S, the population converges to the global optimum (plot 5.3(f)). Therefore, Fig. 5.3 for problem g11 indicates that the proposed R2S can enhance an algorithm's search ability when incorporated with a state-of-the-art algorithm.

5.2.3 General framework

Algorithm 6 presents the general framework for the proposed algorithm using ACD and R2S with DE (DE-R2S).

This algorithm starts with an initial population of size NP, generated randomly within the bounds of each decision variable. Then, the objective value and (ψ) of each individual is computed using equation 5.8, and based on these (ψ), the initial values for δ_{in} and δ_{out} are automatically determined for each function. Equation 6.7 is then used to compute (ψ_{R2S}) of the boundary region. A boundary is constructed for every single constraint ($g_k(\overrightarrow{X})$), as explained in Figure **??**. Then, a trial population is generated via the processes of mutation and crossover, the two populations are compared point by point, based on their fitness values and constraint violations, and a new population is formed according to Deb's feasibility rule [224], whereby one of the following rules is used: 1) of two feasible solutions ($\overrightarrow{x}_1$, $\overrightarrow{x}_2$), the fittest is selected, i.e. $f(\overrightarrow{x}_1) \leq f(\overrightarrow{x}_2)$, 2) a feasible solution is always better than an infeasible one, 3) of two infeasible solutions, the one with the smaller sum of its constraint violations is selected, i.e. $\psi(\overrightarrow{x}_1) \leq \psi(\overrightarrow{x}_2)$. As

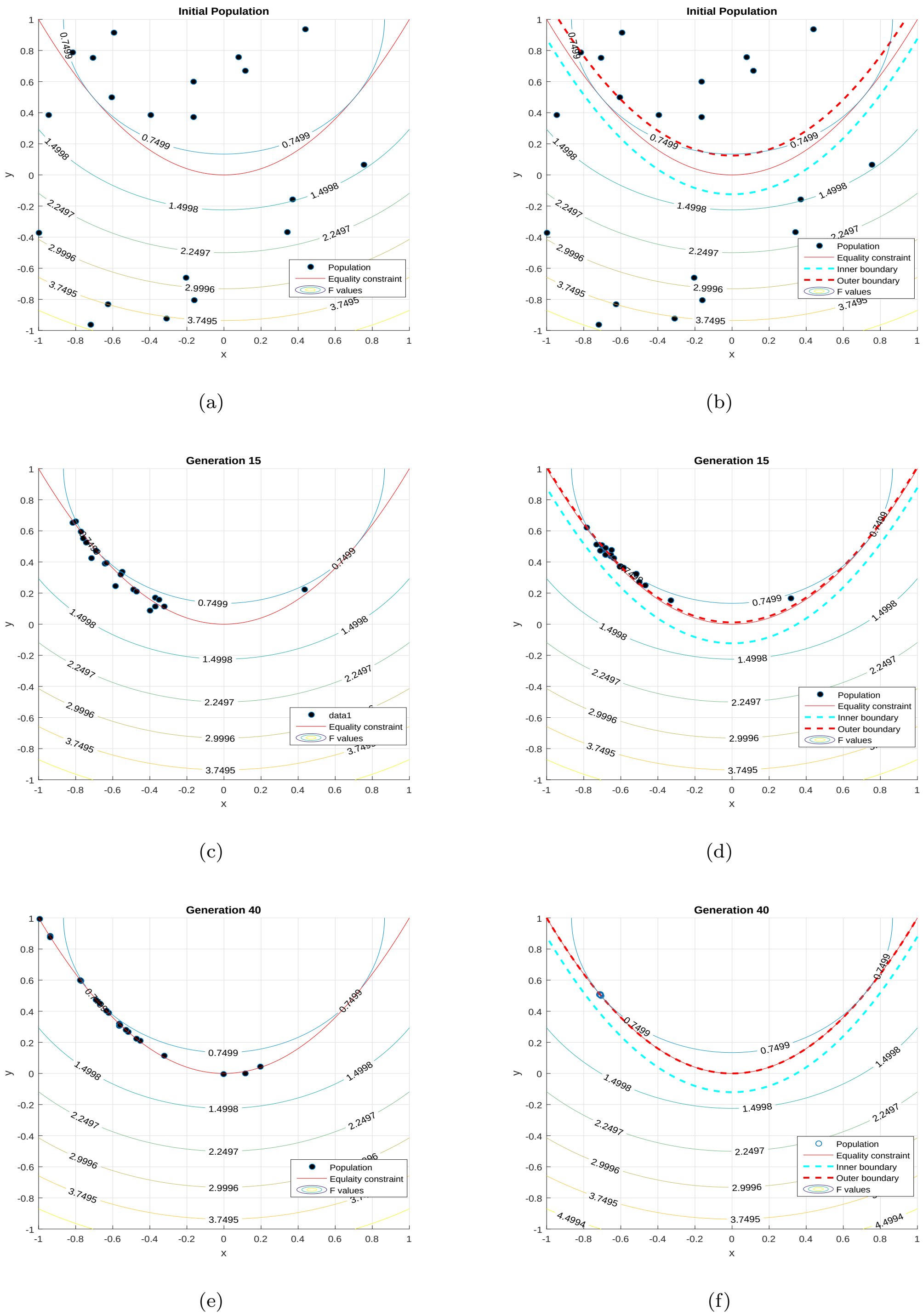


Figure 5.3: The distribution of the population of MJADE algorithm (a) initial population without R2S mechanism, (b) initial population with R2S mechanism, (c) population at gen 15 without R2S mechanism, (d) population at gen 15 with R2S mechanism (e) population at gen 40 without R2S mechanism and (f) population at gen 40 with R2S mechanism

---

Algorithm 6 The proposed framework

1: In generation $iter = 0$, generate an initial random population of size NP.
2: Evaluate objective/constraint function values and the ψ for each individual of the population NP using equation 5.2.
3: Determine the initial values of δ_{in} and δ_{out} based on original total constraint violation as described in Sec 5.2.2.
4: Construct a boundary region for every constraint based on the determined values δ_{in} and δ_{out} as described in Figure 5.1.
5: Update fitness evaluations $cfe \leftarrow NP$;
6: At each generation (iter): Evolve entire population using any algorithm.
7: Update fitness evaluations $cfe \leftarrow cfe + NP$;
8: Determine the active constraints using Algorithm 5.
9: **if** active constraint are present **then**
10:     Remove boundary for non-active constraints;
11:     Update values of δ_{in} and δ_{out} for active constraints by using equations 5.7 and 5.4, respectively.
12:     Update values of total constraint violations based on the updated values of δ_{in} and δ_{out} by using equation 6.7.
13: **else**
14:     Use entire search space
15: **end if**
16: Compare child solutions with parent solutions, and the fittest between them will enter the new population.
17: Stop if the termination criterion is met; else set $iter = iter + 1$, and go to STEP 6.

---

the ACD algorithm runs, at predefined numbers of fitness evaluations, whether or not there are active constraints is determined. If active constraints are present, the values of δ_{in} and δ_{out} are computed and updated, as described in section 5.2.2, otherwise δ_{in} is set to a maximum value (to search the whole inside feasible region), which is done by deleting the inner boundary. The processes of mutation, crossover and acceptance are repeated until a stopping condition is met.

5.3 Experimental Results and Analysis

This section reports comprehensive experiments that were conducted to evaluate the performances of the R2S and ACD methods when combined with modified JADE [64], modified CoDE [68], modified OXDE [225] (MJADE, MCoDE and

MOXDE all are similar to their original algorithm but to be able to solve COPs, Deb's feasibility rule was incorporated in each algorithm), GAMPC [180], SaDE [65] and FRORI [187]. The combinations are referred to as MJADE-R2S, MCoDE-R2S, GAMPC-R2S, SaDE-R2S, MOXDE-R2S and FRORI-R2S, respectively, and were used to optimize the 32 functions presented in CEC2006 [207] and CEC2011 [181].

5.3.1 Experimental setup

In this section, the obtained computational results are presented and analyzed.

To fairly compare the results obtained from the different algorithms, each problem is optimized over 25 independent runs, each algorithm starts from the same initial population for corresponding runs. The algorithms are implemented using Matlab and all experiments were performed on a computer with a 3.4 GHz Core (TM) i7 processor, 16.0G RAM and Windows 7. For all the test functions in CEC2006 and CEC2011, their maximum numbers of fitness evaluations (FES_{max}) are set to 200000 and 150000 , respectively. Val is set to small value. It was set to 0.01. m was set to a value of 10.

5.3.2 Comparison between state-of-the-art algorithms, with and without R2S, for solving CEC2006 and CEC2011

This subsection, briefly discusses the results obtained from six state-of-the-art algorithms, with and without, R2S. Unless otherwise stated, for each algorithm, the following parametric setup was used, based on the relevant guidelines in the literature:

- MJADE with $p = 0.05$, $c = 0.1$ and $NP = 100$.

Table 5.1: Summary of comparison of results obtained from different algorithms with and without R2S for CEC2006 and CEC2011

Algorithms	Criteria	R^+	R^-	p-value	$\alpha = 0.05$
MJADE-R2S vs. MJADE	Best	84	7	0.007	Yes
	Average	246	30	0.001	Yes
MCoDE-R2S vs MCoDE	Best	252	1	0.000	Yes
	Average	297	3	0.000	Yes
MOXDE-R2S vs MOXDE	Best	79	12	0.019	Yes
	Average	175	56	0.039	Yes
FRORI-R2S vs FRORI	Best	189	1	0.000	Yes
	Average	177	13	0.001	Yes
SaDE-R2S vs SaDE	Best	156	15	0.002	Yes
	Average	229	47	0.006	Yes
GAMPC-R2S vs GAMPC	Best	67	11	0.028	Yes
	Average	233	43	0.004	Yes

- MCoDE with $NP = 30$,
- MOXDE with $NP = 100$, $F = 0.9$ and $Cr = 0.9$.
- GAMPC with $NP = 90$, $cr = 1$, $p = 0.1$, maximum number of generations $T_{max} = 1667$, the number of individuals that are selected to fill the archive pool $m = 0.5 \times NP$, and the tournament pool size is selected randomly as 2 or 3.
- SaDE with $NP = 50$,
- FRORI with $NP = 80$, maximum replacement number $MRN = max(5, D/2)$, $F_{pool} = [0.6, 0.8, 0.1]$, and $Cr_{pool} = [0.1, 0.2, 1.0]$

The detailed results (best, mean and standard deviation values) are shown in Appendix C, Tables C.1, C.2. .

The comparison summaries among the algorithms, with and without R2S, are presented in Table 5.1. From this table, it is clear that the number of problems in which the average fitness values obtained by algorithms with R2S is higher

Table 5.2: Comparison of performances of algorithms with and without R2S regarding best and mean computational times and savings in FEs

Algorithms	Saving in Computational Time	Saving in FEs
MJADE-R2S vs. MJADE	17.64 %	8.34%
MCoDE-R2S vs. MCoDE	10.60%	10.66%
MOXDE-R2S vs. MOXDE	36.69%	11.88%
FRORI-R2S vs. FRORI	18.19%	2.37%
SaDE-R2S vs. SaDE	30.29%	5.04%
GAMPC-R2S vs. GAMPC	13.37%	11.12%

than those obtained by algorithms without R2S. In regards to best fitness values obtained, algorithms with R2S, are also better in terms of the number of best values obtained. To understand Table 5.1, for example the first row compares MJADE-R2S with MJADE, in regards to best values obtained. It shows that MJADE-R2S, is better than MJADE, as $R^+ = 84$, $R^- = 7$ and p-value=0.007, which means that MJADE-R2S is significantly better than MJADE.

In addition to the quality of solutions, the computational effort (the number of fitness evaluations (FES)) and times required by these algorithms to obtain optimal solutions within an error of 0.0001, i.e., the stopping condition is $(f(\overrightarrow{x}) - f(\overrightarrow{x^*}) \leq 0.0001)$, where $f(\overrightarrow{x^*})$ indicates the best-known solutions, are compared. A summary of the comparison of the algorithms, with and without R2S, is provided in Table 5.2, in which it is clear that those with R2S are the fastest. This means that this method is capable of improving the performances of the state-of-the-art algorithms, in terms of both computational time and effort. To understand these results, as an example, the second row shows that MCoDE-R2S takes 10.66% fewer FEs and 10.60% less computational time than MCoDE, to reach the known best solutions.

Regarding the Wilcoxon signed-rank test, for both the best and average results obtained, the algorithms with R2S are significantly better than those without it.

The reasons why algorithms with R2s perform better than algorithms without R2s, can be explained as follows. Firstly, ACD works well and is able to distinguish between active and in-active constraints, as will be shown later in Section 5.3.3.1. Secondly, the R2S mechanism helps the solutions to move very fast to the boundary between feasible and infeasible regions, which means it will reach the optimal solution very fast, as is clear from Figure 5.3.

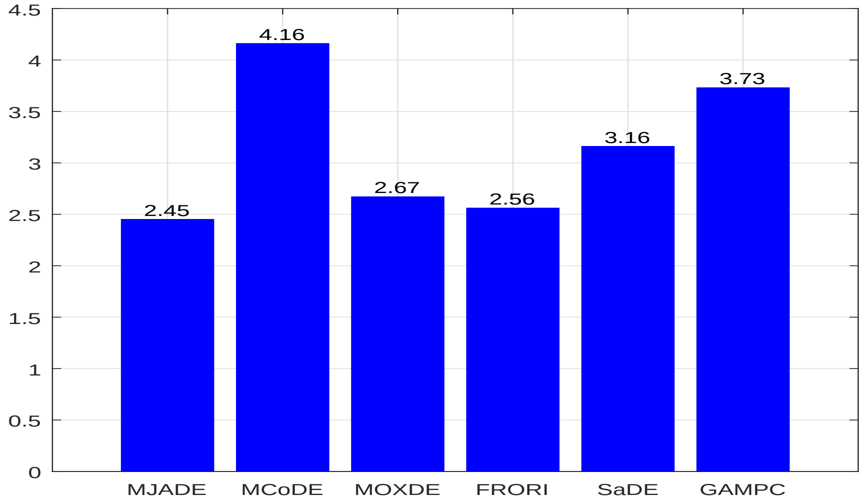
To determine the best-performing algorithm (to be used in the following parametric analysis) , MJADE, MCoDE, MOXDE, GAMPC, FRORI and SaDE are compared, and their rankings obtained from the Friedman test and their mean values, are depicted in Figure 5.4. The figure includes the mean rankings of MJADE-R2S, MCoDE-R2S, MOXDE-R2S, GAMPC-R2S, FRORI-R2S and SaDE-R2S. It is clear that MJADE, both with and without R2S, is the best.

Furthermore, the performance profiles for the average and best results obtained by the MJADE algorithms, with and without R2S, are depicted in Figure 5.5. It is clear that MJADE-R2S is better than MJADE in both cases.

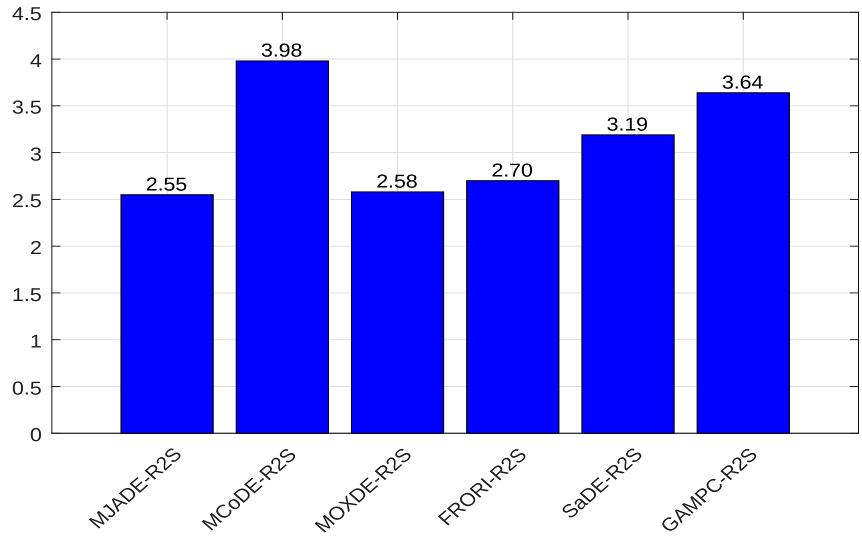
Finally, as a sample, a convergence plot for MJADE and MJADE-R2S, for problems g02, g03, g13, and r01, is presented in Figure 5.6. In the figure, the x-axis represents the number of fitness evaluations and the y-axis represents the fitness value. This figure shows that MJADE-R2S converges much faster than MJADE.

5.3.3 Parametric analysis

This section analyzes the R2S mechanism and its parameters.



(a)

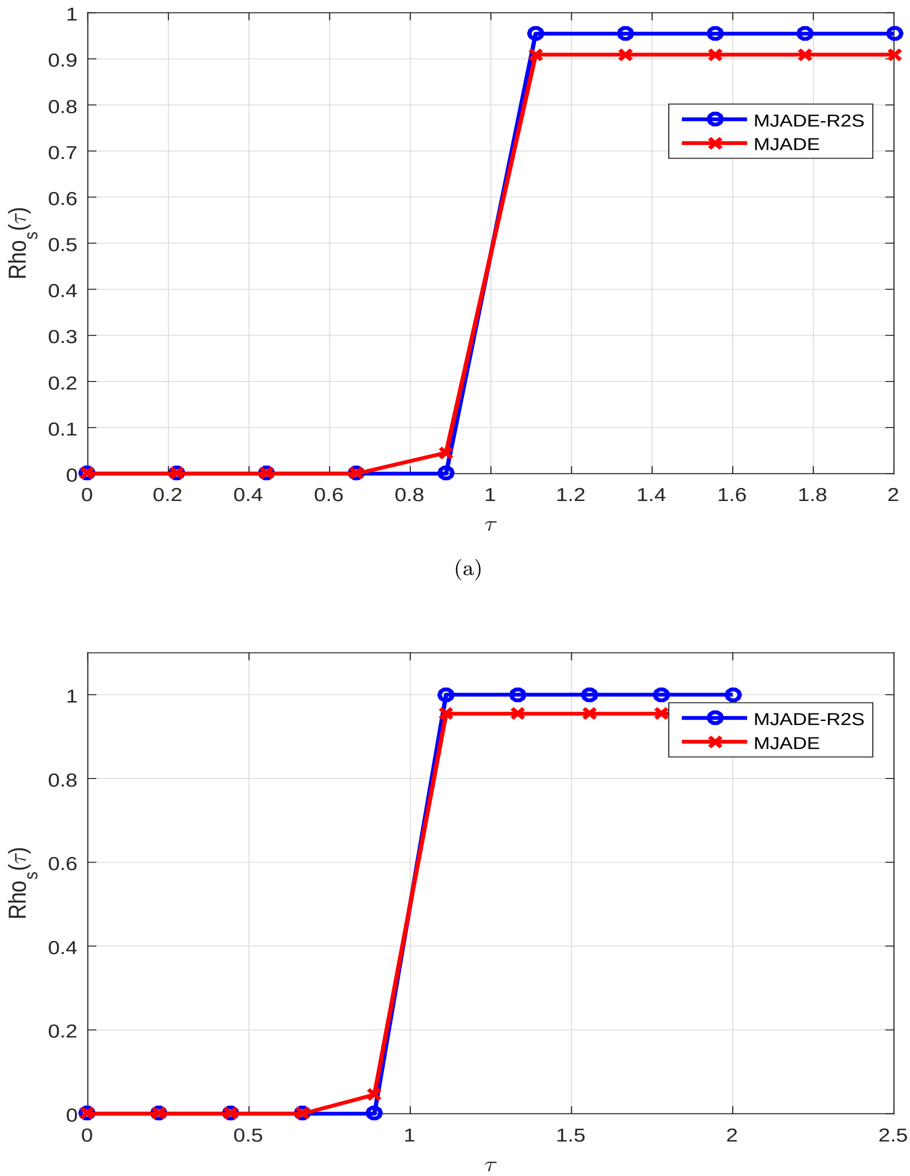


(b)

Figure 5.4: Average rankings for (a) MJADE, MCoDE, MOXDE, GAMPC, FRORI, and SaDE, and (b) MJADE-R2S, MCoDE-R2S, MOXDE-R2S, GAMPC-R2S, FRORI-R2S, and SaDE-R2S

5.3.3.1 Active constraints

As previously mentioned, by their nature equality constraints are active constraints, and so they do not require any mechanism for identification. So the ACD mechanism is only tested on the CEC2006 problems that have inequality



(a)

(b)

Figure 5.5: Plots of performance profiles for MJADE-R2S and MJADE based on (a) best results, and (b) average results

constraints. Table 5.3 shows the actual numbers of active constraints reported in [207], the predicted numbers of active constraints determined by the proposed ACD mechanism, and the number of FEs required for each problem to determine the active constraints. It is clear that, ACD was able to identify all the active constraints successfully, with the last column indicating its success rate (SRs),

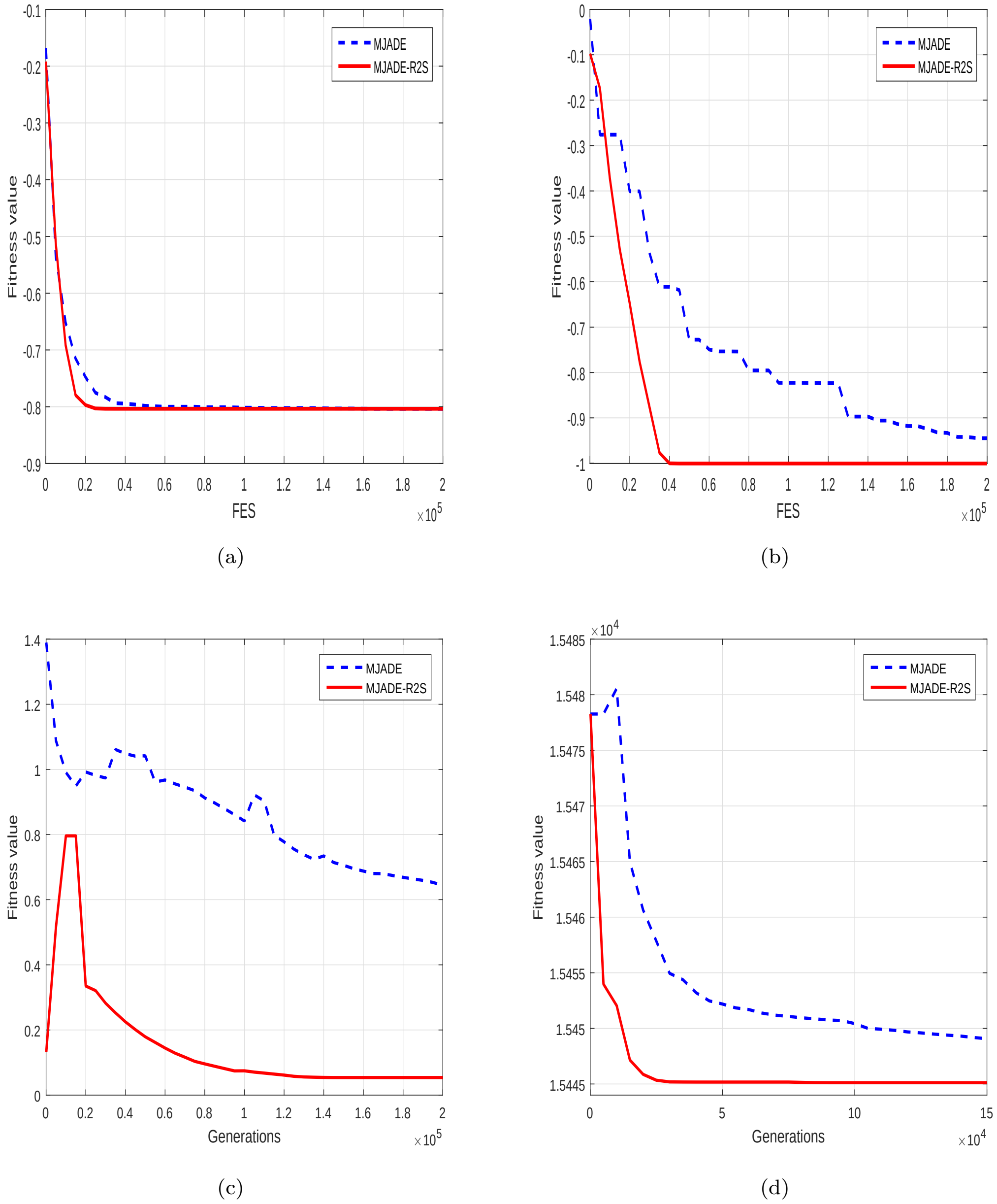


Figure 5.6: Convergence plots of MJADE-R2S and MJADE for (a) g02 (b) g03 (c) g13 and (d) r01

calculated by:

$$SR = \frac{\#\, of\, predicted}{\#\, of\, actual} \times 100 \tag{5.12}$$

Therefore, based on the FES_active_required column in Table 5.3, FES_{ACD} is set to the maximum value (35200). For every problem except g07, the ACD

Table 5.3: Number of active constraints determined by ACD

Problem	Actual	Predicted	FES_active_required	SR
g01	6	6	30200	100
g04	2	2	15200	100
g06	2	2	15200	100
g07	6	5	35200	83.3
g09	2	2	20200	100
g10	3	3	20200	100
g16	4	4	5200	100
g18	6	6	25200	100
g24	2	2	5200	100

mechanism was able to obtain the same number of active constraints as the actual number.

5.3.3.2 Effect of CS

The effect of CS, which is an interval that is the number of generations for evaluating whether constraints are active or not, is analyzed using different numbers of generations, namely 1, 25, 50, 75 and 100. The detailed results (mean and standard deviation values) are shown in Appendix C, Table C.7 . The overall rankings, based on the Friedman test, are presented in Table 5.4. The table shows that $CS = 25$ is the best. The Wilcoxon signed-rank test results are presented in Table 5.5, in which for example row 6, which compares $CS = 25$ and $CS = 75$, indicates that $CS = 25$ is superior for 2 test problems, obtains the same results for 20 and is inferior for 0, which means that it is slightly better than $CS = 75$, although the difference between them is not significant. Also, it is clear that CS values equal to or greater than 25 are significantly better than $CS = 1$, and while those of $CS = 25$, 50, 75, and 100 do not greatly affect the results, $CS = 25$ is slightly better.

Table 5.4: Mean ranks of different CS values based on Friedman test

Algorithms	Mean rank	Order
CS=1	3.55	5
CS=25	**2.75**	**1**
CS=50	2.89	2.5
CS=75	2.89	2.5
CS=100	2.93	4

Table 5.5: Summary of comparisons of different variants of proposed algorithm with different CS values (1, 25, 50, 75 and 100) based on average results, where 'Dec.' statistical decision is based on Wilcoxon rank-sum test

Algorithms	Better	Equal	Worse	Dec.
CS=1 vs. CS=25	0	16	6	-
CS=1 vs. CS=50	0	16	6	-
CS=1 vs. CS=75	0	16	6	-
CS=1 vs CS=100	0	16	6	-
CS=25 vs CS=50	1	21	0	$\approx$
CS=25 vs. CS=75	2	20	0	$\approx$
CS=25 vs. CS=100	2	20	0	$\approx$
CS=50 vs. CS=75	1	20	1	$\approx$
CS=50 vs. CS=100	1	20	1	$\approx$
CS=75 vs CS=100	1	21	0	$\approx$

5.3.3.3 Effect of FES_c

As previously mentioned, δ_{out} and δ_{in} are updated using equations (5.4) and (5.7), respectively up to a predefined number of FEs (FEs_c). To analyze the effect of FES_c, experiments were conducted using different values for it of $0.3 \times FES_{max}$, $0.5 \times FES_{max}$, $0.7 \times FES_{max}$, and $0.9 \times FES_{max}$ while all other parameters are fixed. The detailed results (mean and standard deviation values) are shown in Appendix C, Table C.8. The variants were ranked based on the Friedman test and their results are presented in Table 5.6, which shows that $FES_c = 0.7 \times FES_{max}$ is slightly better than the others. Also, a Wilcoxon rank-sum test of the different variants of the proposed algorithm with different values was conducted. A summary of its results is presented in Table 5.7. It indicates that $FES_c = 0.7 \times FES_{max}$ is slightly better than the other variants.

Table 5.6: Mean ranks of different FES_c values, based on Friedman test

Algorithms	Mean rank	Order
$FES_c = 0.3 \times FES_{max}$	2.66	3
$FES_c = 0.5 \times FES_{max}$	2.07	2
$FES_c = 0.7 \times FES_{max}$	**1.98**	**1**
$FES_c = 0.9 \times FES_{max}$	3.30	4

Table 5.7: Summary of comparisons of different variants of proposed algorithm with different values (0.3, 0.5, 0.7 and 0.9) based on average results, where 'Dec.' indicates statistical decisions based on Wilcoxon rank-sum test

Algorithms	Better	Equal	Worse	Dec.
$FES_c = 0.3FES$ vs. $FES_c = 0.5FES$	0	14	8	$-$
$FES_c = 0.3FES$ vs. $FES_c = 0.7FES$	1	14	7	$-$
$FES_c = 0.3FES$ vs. $FES_c = 0.9FES$	11	7	4	$\approx$
$FES_c = 0.5FES$ vs. $FES_c = 0.7FES$	1	18	3	$\approx$
$FES_c = 0.5FES$ vs. $FES_c = 0.9FES$	14	7	1	$+$
$FES_c = 0.7FES$ vs. $FES_c = 0.9FES$	15	7	0	$+$

5.3.4 Comparison between state-of-the-art algorithms, with and without R2S, for solving CEC2010

From the above-mentioned analysis, it was found that, FES_active_required = 125, $CS = 25$ and $FES_c = 0.7 \times FES_{max}$are the best parameters. So, these parameters will be used in the comparison in this section.

A set of 36 test instances (18 problems each with 10 and 30 dimensions) that were introduced in CEC2010 [186] has been solved. The algorithm was run 25 times for each test problem, where the stopping criterion was to run for up to $200K$ FEs for $10D$ instances, and $600K$ FEs for $30D$. The detailed results (best, mean and standard deviation values) are shown in Appendix C, Tables C.3, C.4, C.5, C.6.

Considering both the best and average results, a comparative analysis is presented in Table 5.8. To explain the results, for example for MJADE-R2S vs

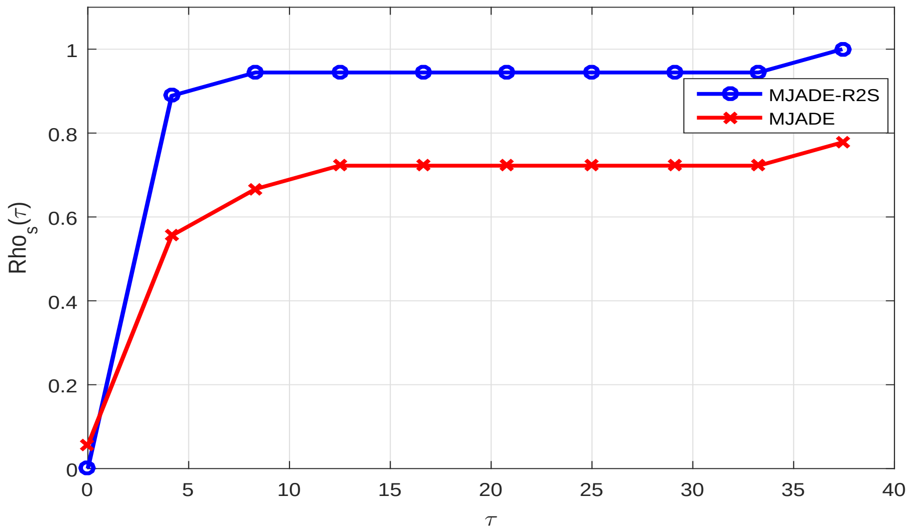
Table 5.8: Summary of comparison of results obtained from different algorithms with and without R2S for CEC2010

		10D				30D			
Algorithms	Criteria	R^+	R^-	p-value	Dec.	R^+	R^-	p-value	Dec.
MJADE-R2S vs. MJADE	Best	78	0	0.002	+	119	1	0.001	+
	Average	110	26	0.043	+	135	15	0.004	+
MCoDE-R2S vs MCoDE	Best	143	28	0.012	+	166	5	0.000	+
	Average	140	31	0.018	+	140	31	0.018	+
MOXDE-R2S vs MOXDE	Best	149	4	0.001	+	162	9	0.001	+
	Average	144	47	0.011	+	139	32	0.020	+
FRORI-R2S vs FRORI	Best	10	7	0.842	≈	35	1	0.017	+
	Average	48	30	0.480	≈	78	42	0.307	≈
SaDE-R2S vs SaDE	Best	104	1	0.001	+	151	20	0.004	+
	Average	153	18	0.003	+	149	22	0.006	+
GAMPC-R2S vs GAMPC	Best	43	2	0.015	+	66	12	0.034	+
	Average	87	4	0.004	+	88	17	0.026	+

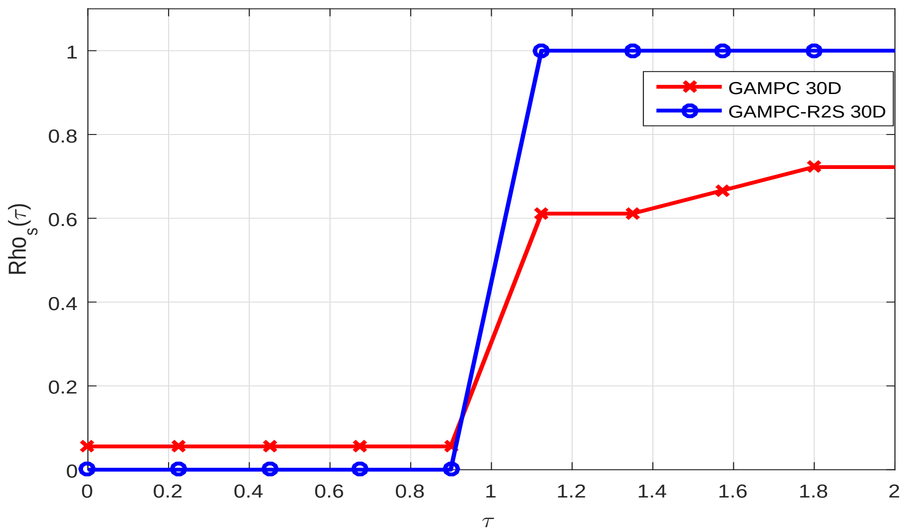
MJADE, it can be observed that MJADE-R2S provides higher R^+ values than R^- for MJADE, for both best and average values for $10D$ and $30D$. Moreover, the p-values of all the cases are less than 0.05, which indicates that algorithms with R2S exhibit statistically superior convergence performance against the algorithms without R2S, except for FRORI.

Furthermore, as an example, the performance profile comparing MJADE and GAMPC with and without R2S, is presented in Figure 5.7. This figure shows that MJADE-R2S and GAMPC-R2S are better than MJADE and GAMPC, as MJADE-R2S and GAMPC-R2S reach a probability of 1 (when $\tau = 37$ and $\tau = 1$, respectively) before their corresponding MJADE and GAMPC.

Since there is no significance different between FRORI with and without R2S, the complexity of these two algorithms is computed followed the guidelines in [186] for both 10D and 30D. This complexity is presented in Table 5.9. It can be concluded, that the time complexity of FRORI-R2S is less than the time complexity of FRORI, and that means R2S did not add any additional complexity to the original algorithms.



(a)



(b)

Figure 5.7: Performance profiles comparing MJADE and GAMPC, with and without R2S, for CEC2010 for 30D (a) MJADE and (b) GAMPC

5.4 Chapter Summary

As of the optimization literature, the optimal solution of COPs often lies on the intersection of the active constraints. Therefore, in this chapter, the R2S mechanism is proposed, which searches the boundary area between the feasible

Table 5.9: Comparison between FRORI and FRORI-R2S based on algorithm complexity

	FRORI				FRORI-R2S			
	T_0	T_1	$\widehat{T}_2$	$\frac{\widehat{T}_2-T_1}{T_0}$	T_0	T_1	$\widehat{T}_2$	$\frac{\widehat{T}_2-T_1}{T_0}$
10D	0.14293	0.121451	4.1496266	28.18286	0.14293	0.121451	3.60050308	**24.34095**
30D	0.14293	0.312842	5.28531388	34.78956	0.14293	0.312842	4.93899166	**32.36654**

and infeasible regions. R2S comprises two components. The first is an ACD mechanism which enables the algorithm to distinguish between active and non-active constraints. The second performs a boundary construction, in which after determining active constraints, the algorithm defines the thickness of the boundary area between the feasible and infeasible regions for active constraints, while this boundary area is removed for non-active ones. Also, it deals with the case in which the optimal solution lies inside the feasible region, by searching the original search space by increasing the value for the inner delta.

This new method is incorporated in six state-of-the-art algorithms that were tested on the CEC'2006 and CEC'2010 single-objective COPs, as well as a number of real-application constrained problems, with the results demonstrating that it is capable of improving the performances of these algorithms.

Motivated by the encouraging results obtained in this and the previous chapters, a new multi-method algorithm incorporating both landscape information and R2S, is proposed for solving COPs in the next chapter.

Chapter 6

Multi-EAs Framework for Constrained Problems

This chapter introduces a novel multi-method algorithm to solve COPs. It incorporates both landscape methods and the reduced search space mechanism proposed in Chapters 4 and 5. The experimental results from, and analysis of, the proposed method are presented and discussed by solving the previously used benchmark problems. Finally, detailed results obtained from the proposed algorithm, and comparisons with those from state-of-the-art algorithms, are presented.

6.1 Introduction

As previously mentioned, no single algorithm is able to solve a wide range of optimization problems. Also, in COPs, some constraints are much more important than others.

As a consequence, a new landscape-based DE algorithm was proposed in Chapter 4 for solving COPs. It used information about a problem's landscape to attempt to put more emphasis on the best DE operator, from a pool of them. It showed better performance than other state-of-the-art algorithms in both quality of solutions and computational time. Also, the proposed reduced search space

Part of this work has previously appeared in

1. K. M. Sallam, S. M. Elsayed, R. A. Sarker, and D. L. Essam, "Multi-method based orthogonal experimental design algorithm for solving cec2017 competition problems," in Evolutionary Computation (CEC), 2017 IEEE Congress on. IEEE, 2017, pp. 1350-1357.

mechanism in Chapter 5, demonstrated its capability to enhance state-of-the-art algorithms. So, a framework that can utilize both a problem's landscape and reduced search space in one single framework, may result in better performance.

Therefore, the main aim of this chapter is to propose a new multi-EA algorithm, that utilizes the benefits of both a problem's landscape and the search space reduction mechanism. In its entire process, information from the objective function and constraints landscapes are used to choose the best algorithm during the evolutionary process. Concurrently the reduced search space mechanism focuses the search process around the active constraints. The performance of the proposed algorithm has been tested by solving the CEC2006 [207], CEC2010 [186] and CEC2011 [181] problems defined in Chapter 3. Also, several experiments were designed and carried out to analyze the effects of different parameters on the performance of the proposed algorithm. The results from the final variant of the proposed algorithm are compared with those from well-known algorithm in the literature.

The organization of this chapter is as follows: the next section describes the proposed framework; the experimental results and analysis are reported in Section 6.3; and a chapter summary is presented in Section 6.4.

6.2 Proposed Algorithm

The general algorithmic framework of multi-EA, using a problem's landscape and reduced search space (MEA-LS-R2S) is described in this section.

6.2.1 MEA-LS-R2S

The main steps of MEA-LS-R2S are shown in Algorithm 7. Initially, a random population of size NP individuals is generated where each variable of each individual must be within the boundaries. Then, the initial population is divided into m sub-populations of sizes, NP_1, NP_2,..., NP_m respectively. The individuals in each population are evolved by different EAs (DE, GA and CMA-ES). Each algorithm is applied based on its probability ($prob_{EA}$). Initially, the probabilities of all EAs are equal to 1. Once a half cycle (CS generations) is finished, all $prob_{EA}$ are dynamically updated based on the landscape measure, as discussed in Section 6.2.1.2. In the subsequent cycle, in each generation, m random numbers, between 0 and 1 are generated, based on which the corresponding EA is used to evolve its own sub-population. After another CS generations, an information sharing mechanism is executed, and $prob_{EA}$ for every EA, is set to a value of 1. This cycle is repeated until the algorithm finishes.

Algorithm 7 Proposed MEA-LS-R2S algorithm

1: Define $NP \leftarrow NP_1 + NP_2 + ... + NP_m$, $count \leftarrow 0$, $prob_p \leftarrow 1, \forall p = 1, 2, ..., m$, $FES_{max} \leftarrow 10,000D$, and all other parameters required.
2: Generate an initial NP random individuals(X), with the variables of each one ($\overrightarrow{x}_z$) must be within their boundaries;
3: Assign random NP_1, NP_2, ..., NP_m individuals from X to X_p, $\forall p = 1, 2, ..., m$, respectively.
4: **while** $FES \leq FES_{max}$ **do**
5:     $count \leftarrow count + 1$
6:     **if** $count == CS$ **then**
7:         Update $prob_p$, $\forall p = 1, 2, ..., m$, as described in 6.2.1.2;
8:     **end if**
9:     **if** $count == 2 \times CS$ **then**
10:         Execute the information sharing mechanism as in section 6.2.1.3 and $prob_p \leftarrow 1, \forall p = 1, 2, ..., m$ and $count \leftarrow 1$;
11:     **end if**
12:     **if** $rand(p) \in [0, 1] \leq prob_p$ **then**
13:         Evolve the corresponding sub-population using the assigned EA, update FES and sort X_1;
14:     **end if**
15:     Go to step 4;
16: **end while**

6.2.1.1 The used Algorithms

In this section, the description of the EA algorithms used in the framework, are given.

Multi-operator DE (MODE): starts with NP_1 individuals which are randomly taken from the entire NP individuals. It uses the following three DE mutation strategies.

- DE/current-to-ϕbest with archive/1/bin

$$u_{i,j} = \begin{cases} x_{i,j} + F_i.(x_{\phi,j} - x_{i,j} + x_{r1,j} - x_{r2,j}) \\ \qquad \text{if } (rand \leq cr_i \text{ or } j = j_{rand}) \\ x_{i,j} \qquad \text{otherwise} \end{cases} \tag{6.1}$$

- DE/rand-to-ϕbest with archive/1/bin

$$u_{i,j} = \begin{cases} x_{r1,j} + F_i.(x_{\phi,j} - x_{r1,j} + x_{r3,j} - x_{r2,j}) \\ \qquad \text{if } (rand \leq cr_i \text{ or } j = j_{rand}) \\ x_{i,j} \qquad \text{otherwise} \end{cases} \tag{6.2}$$

- DE/ϕbest/1/bin

$$u_{i,j} = \begin{cases} x_{\phi,j} + F_i.(x_{r1,j} - x_{r3,j}) \\ \qquad \text{if } (rand \leq cr_i \text{ or } j = j_{rand}) \\ x_{i,j} \qquad \text{otherwise} \end{cases} \tag{6.3}$$

where $r_1 \neq r_2 \neq r_3 \neq i$ are random integer numbers, $\overrightarrow{x}_{r1}$ and $\overrightarrow{x}_{r3}$ are randomly selected from the entire population, $x_{\phi,j}$ was selected from the best 10% individuals from entire populations, while $x_{r2,j}$ was selected from the union of the entire population and an archive.

Each variant evolves the same number of individuals, and for each individual in the entire sub-population, a new offspring individual is generated using the assigned DE operator. Then, for each offspring the objective value and the total constraint violation is calculated (see Section 6.2.1.3). A selection process is carried out between every offspring and its parent, and the fittest enters the new population, as discussed in Section 6.2.1.3. Concurrently the landscape index for each DE is calculated. Once a half cycle (CS) is finished, the best-performing DE operator is selected based on the normalized measure, to evolve the entire whole sub-population for the subsequent half cycle. Again, this cycle is repeated while the algorithm runs.

Multi-operator GA (MOGA): like MODE, it starts with NP_2 individuals randomly selected from the whole population NP individuals. In its entire process, two effective GA variants are used, GA with simulated binary crossover (SBX) and a non-uniform mutation (NU-M), as well as MPC-GA as in [226]. The entire process of MOGA is similar to MODE, with the only difference being that the two GA algorithms are always used, based on the effectiveness of each one of them. To generate an offspring, a random number ($rand \in [0, 1]$) is generated, then if it is greater than a predefined probability ($P(1)$), two new individuals are generated by SBX-NU, otherwise three individuals are produced by MPC-GA. Then the number of successful individuals (entered the next population) generated by MPC-GA is saved in S_1, while the successful ones produced by SBX-NU are saved in S_2. Then ($P(1)$) is updated every generation by $P(1) = \frac{S_1}{S1+S_2}$.

CMA-ES: has shown its capability to solve different types of optimization problems [99] successfully. It originated from the concept of self-adaptation in ES, which adapts the covariance matrix of a multivariate normal search distribution. In it, the new individuals are created by a sampling of a Gaussian distribution, and rather than using a single step mutation; it depends on the path that the population takes over generations [227]. Algorithm 8 shows the main steps of CMA-ES.

Algorithm 8 CMA-ES algorithm

1: Generate an initial individual $\overrightarrow{x}_m$ and calculate its fitness function value;
2: Sample new solutions, such that $\overrightarrow{x}_{z,G+1} = \overrightarrow{x}_m + \sigma N(0,\, C_G)\, \forall z = 1 : NP$;
3: Evaluate and sort the new offspring.
4: Select μ individuals as a parental vector. Then calculate their center using $\overrightarrow{x}_{m,G+1} = \sum_{k=1}^{\mu} w_k \overrightarrow{x}_{k,G}$ where $\sum_{k=1}^{\mu} w_k = 1$ and $w_1 \geq w_2 \geq ... \geq w_\mu$.
5: Update the evolution path p^s_{G+1} and p^σ_{G+1}.
6: Adapt the covariance matrix (C_{G+1}).
7: Update the global step size (σ_{G+1}).
8: Go to steps 2, and repeat theses steps until a stopping criterion is met.

6.2.1.2 Normalized landscape measure

In this chapter, the overall normalized average landscape measure, for each algorithm is computed as in Chapter 4 (Section 3.3.1.2). The normalized value $N_{LD_{EA}}$ of algorithm EA is calculated by:

$$N_{LD_{EA}} = \frac{1 - M_{LD_{EA}}}{\sum_{OP=1}^{m}(1 - M_{LD_{EA}})} \tag{6.4}$$

where $\{EA = 1, 2, ..., m\}$, and M_{LD} is the mean value of the landscape information measure.

Subsequently, the overall normalized performance, $ON_{LD_{EA}}$, of each EA is computed by equation 4.5:

$$ON_{LD_{EA}} = \frac{N_{LD_{EA}}}{\sum_{EA=1}^{m}(N_{LD_{EA}})} \tag{6.5}$$

Then, each probability is updated as:

$$prob_{EA} = max(0.1,\ \min(0.9, \frac{ON_{LD_{EA}}}{\sum_{EA=1}^{m}(ON_{LD_{EA}})}), \forall EA = 1, 2, ..., m \tag{6.6}$$

Note: Equation 6.5 is used to determine the best-performing operator in MODE.

6.2.1.3 Selection process

The selection process between any offspring and its parent according to a modified form of Deb's feasibility rule [224], whereby one of the following rules is used: 1) of two feasible solutions ($\overrightarrow{x}_1$, $\overrightarrow{x}_2$), the fittest is selected, i.e. $f(\overrightarrow{x}_1) \leq f(\overrightarrow{x}_2)$, 2) a feasible solution is always better than an infeasible one, 3) of two infeasible solutions, the one that has a smaller value of ψ_{R2S} is preferred, i.e. $\psi_{R2S}(\overrightarrow{x}_1) \leq \psi_{R2S}(\overrightarrow{x}_2)$. ψ_{R2S} of an individual ($\overrightarrow{x_z}$) is calculated based on the reduced search space mechanism proposed in Chapter 5, as:

$$\psi_{R2S}(\overrightarrow{x}) = \sum_{i=1}^{m} max(\,0,\, max(\,L,\, R)) \tag{6.7}$$

where L and R are computed the following equations as:

$$L = -\delta_{in} - g_k(\overrightarrow{x}) \leq 0 \tag{6.8}$$

$$R = g_k(\overrightarrow{x}) - \delta_{out} \leq 0 \tag{6.9}$$

where δ_{in} and δ_{out} are are the inner-delta and outer-delta values, respectively, and their values are updated using the mechanism developed in Chapter 5 (Section 5).

6.2.1.4 Information sharing

The information sharing method in MEA-LS-R2S is simple. At the end of every cycle ($count = 2 \times CS$), all the m sub-populations are grouped together in the whole entire population. Then the sup-populations are filled with (NP_1, NP_2,...,NP_m) random individuals from the whole entire population. Then, the parameters of CMA-ES are reset to their initial values, except for σ, which is updated as $\sigma = \sigma_{initial} \times (1 - \frac{FES}{FES_{max}})$.

6.3 Experimental Results and Analysis

This section discusses the performances of the proposed algorithm when tested by solving the sets of benchmark problems presented in Chapter 4 (Section 4.2). To obtain comprehensive comparisons, each algorithm was run 25 times for all the optimization functions to a maximum number of FEs of 200000 for CEC2006, $20000 \times D$ for CEC2010 and 150000 for CEC2011, with the best, mean and standard deviation results recorded.

Regarding parameter settings, the values of φ, A, H were set to the same values as in Chapter 4, Section 4.4. The linear population size reduction mechanism [67] previously described in Chapter 3 is used, in which NP_{EA}^{init} was set at a value of $(NP - NP_3)/2$ and NP_{EA}^{min} was set at a value of 4. Note, the linear reduction mechanism is used for only DE and GA. The population size used for CMA-ES is set to 10. The CS parameter was set to 25, while the values of δ_{in} and δ_{out} were set to the same values as in Chapter 5, Section 5.3.

6.3.1 Parameter analysis

In this subsection, the effects of the following parameters on the performance of the proposed algorithm are examined: 1) the number of EAs used; 2) CS; 3)

Table 6.1: Comparisons of summary of Var1, Var2 and Var3

Algorithms	Best results				Average results			
	Better	Equal	Worse	Dec.	Better	Equal	Worse	Dec.
Var1 vs. Var2	10	7	1	+	13	1	4	≈
Var1 vs. Var3	2	7	9	≈	3	3	12	≈
Var2 vs. Var3	2	6	10	≈	3	1	14	−

Table 6.2: Average ranks of Var1, Var2 and Var3, obtained from Friedman test based on both best and average results

Algorithms	Average rank based on best results obtained	Average rank based on average results obtained
Var1	2.00	1.94
Var2	2.56	2.47
Var3	**1.44**	**1.58**

linear reduction 4) NP_{EA}^{init}; and 5) NP_{EA}^{min}. To do this, the $10D$ test problems from CEC2010 are considered.

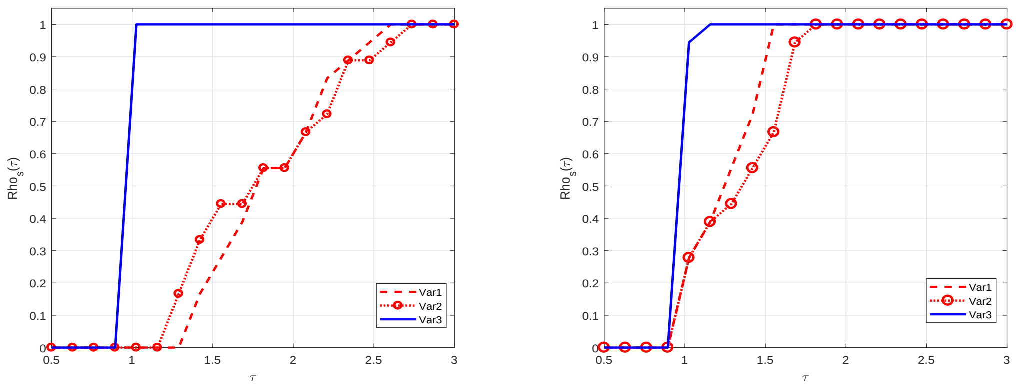
6.3.1.1 Effect of the number of the EAs, m

The number of EA algorithms (m) required for MEA-LS-R2S was analyzed. To do this, three different variants of the proposed algorithm (Var1, Var2 and Var3) were run. Var1 uses MODE, MOGA and CMA-EA, while for Var2 MODE and MOGA are used and Var3 utilizes MODE and CMA-ES. All the parameter settings are set as those in the previous section. The detailed results (best, average, St.d) are presented in Appendix D, Table D.1.

Regarding the quality of solutions, Table 6.1 shows of a summary of comparisons of these variants. It indicates that there was a bias towards Var3. Based on Wilcoxon signed-rank test results, Var1 was significantly better than Var2 for the best results, while Var3 was significantly better than Var2 for the average results. Also, a Friedman test was conducted to rank these variants, with the results presented in Table 6.2 showing that Var3 was the best variant.

Table 6.3: Computational times and average FEs for Var1, Var2 and Var3

Algorithms	Total average time	Total average FEs
Var1	9.54	155974
Var2	8.84	163799
Var3	**5.25**	**128841**



(a) (b)

Figure 6.1: Comparison of performance profiles of proposed algorithm with different m values, based on average (a) computational time; and (b) FEs

Also, the total average time and number of FEs needed to reach the optimal solution with an error of 0.0001 were compared. Table 6.3 presents the total average time and number of FEs for each variant, for both of which it is clear that Var3 was better than all the others.

Furthermore, the performance profiles depicted in Figure 6.1 show that Var3 was the best for both average computational time and FEs.

6.3.1.2 Effect of CS

To determine the effect of the cycle half CS, MEA-LS-R2S was tested using Var3 (MODE and CMA-ES), which was better than the other two variants from the previous section, but with different values of CS of 25, 50, 75 and 100 generations, with the detailed results are presented in Appendix D, Table D.2. CS specifies how often MEA-LS-R2S selects the best algorithm, and how often MODE

Table 6.4: Comparisons of summary of different variants of the proposed algorithm with different CS values

Algorithms	Best results				Average results			
	Better	Equal	Worse	Dec.	Better	Equal	Worse	Dec.
$CS = 25$ vs. $CS = 50$	1	15	2	$\approx$	11	7	0	$+$
$CS = 25$ vs. $CS = 75$	3	14	1	$\approx$	12	6	0	$+$
$CS = 25$ vs. $CS = 100$	4	13	1	$\approx$	11	6	1	$+$

Table 6.5: Average ranks of different values of CS, obtained from Friedman test based on both best and average results

Algorithms	Average rank based on best results obtained	Average rank based on average results obtained
$CS = 25$	2.39	**1.58**
$CS = 50$	**2.36**	2.58
$CS = 75$	2.58	2.67
$CS = 100$	2.67	3.17

selects the best DE operator.

Table 6.4 presents a comparison of different variants of the proposed MEA-LS-R2S with different CS values. Regarding the best results, the variant with $CS = 25$ was better than, inferior to, and obtained similar results with, the variants $CS = 50$, $CS = 75$ and $CS = 100$ for 1, 3 and 4, and 2, 1 and 1, and 15, 14 and 13, respectively, while for the average results obtained MEA-LS-R2S with $CS = 25$ was superior to $CS = 50$, $CS = 75$ and $CS = 100$ in 11, 12 and 11, test problems respectively, but inferior to these variants for 0, 0, and 1 test problems.

Regarding the Wilcoxon signed-rank test, for the average results obtained, the proposed MEA-LS-R2S with $CS = 25$ was significantly better than the others, while there is no significance difference between these variants for the best results.

For further analysis, a Friedman test was conducted with the results presented in Table 6.5 and Figure 6.2, from which it is clear the overall rank of $CS = 25$ performed best.

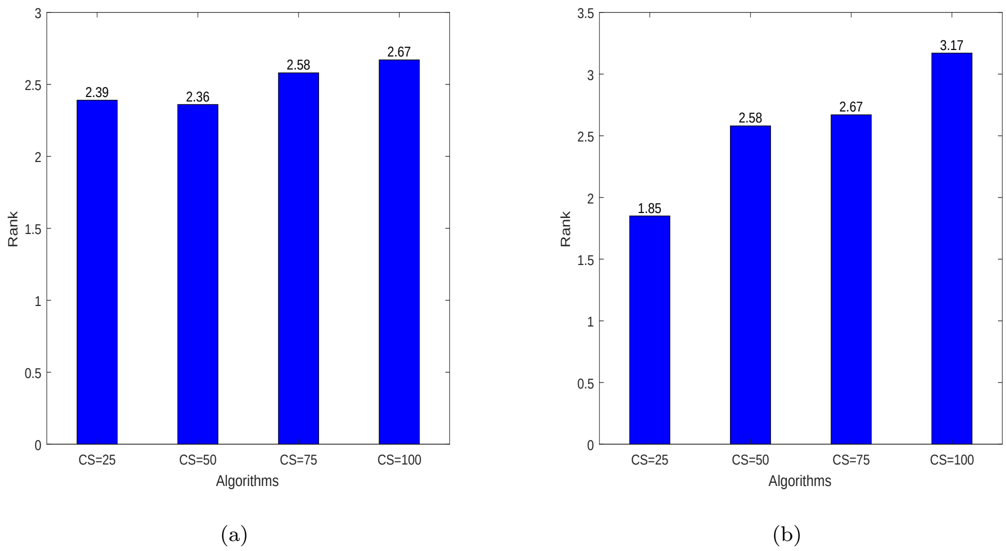


Figure 6.2: Average ranks of different values of CS, obtained from Friedman test based on (a) best results; and (b) average results

6.3.1.3 Effect of linear population size reduction mechanism

To judge the effect of the linear population size mechanism used, two variants of the proposed MEA-LS-R2S algorithm, with linear reduction V1, and without linear reduction V2, were tested. The detailed results are presented in Appendix D, Table D.3.

Regarding the quality of solutions, Table 6.6 shows comparisons of these two variants, which indicate that V1 with the population size linear reduction mechanism was best. Based on a Wilcoxon singed-rank test, V1 was statistically better than V2 in terms of best and average results obtained. The average computational time and FEs consumes by V1 to reach the optimal solution with an error equal to 0.0001 was 7.33s and 128318, respectively, while for V2 it was 8.32s and 140189, respectively.

Table 6.6: Comparisons of summary between the proposed algorithm with and without linear reduction

Algorithms	Criteria	Better	Equal	Worse	Dec
V1 vs. V2	Best	12	6	0	+
	Mean	14	4	0	+

Table 6.7: Comparisons of summary of different variants of the proposed algorithm with different NP^{init} values

Algorithms	Best results				Average results			
	Better	Equal	Worse	Dec.	Better	Equal	Worse	Dec.
$NP^{init} = 50$ vs. $NP^{init} = 75$	0	17	1	$\approx$	4	6	8	$\approx$
$NP^{init} = 50$ vs. $NP^{init} = 100$	1	16	1	$\approx$	3	6	9	$-$
$NP^{init} = 50$ vs. $NP^{init} = 125$	2	15	1	$\approx$	3	6	9	$-$
$NP^{init} = 50$ vs. $NP^{init} = 150$	3	14	1	$\approx$	5	5	8	$-$
$NP^{init} = 75$ vs. $NP^{init} = 100$	2	16	0	$\approx$	3	10	5	$\approx$
$NP^{init} = 75$ vs. $NP^{init} = 125$	3	15	0	$\approx$	4	11	3	$\approx$
$NP^{init} = 75$ vs. $NP^{init} = 150$	4	14	0	$\approx$	9	7	2	$\approx$
$NP^{init} = 100$ vs. $NP^{init} = 125$	3	15	0	$\approx$	7	11	0	+
$NP^{init} = 100$ vs. $NP^{init} = 125$	4	14	0	$\approx$	9	9	0	+
$NP^{init} = 125$ vs. $NP^{init} = 150$	4	14	0	$\approx$	8	8	2	$\approx$

6.3.1.4 Effect of NP^{init}

To observe the effect of the initial population size of MODE, different experiments with NP^{init} = 50, 75, 100, 125 and 150 were conducted, with the detailed results presented in Appendix D, Table D.4.

Regarding the quality of solutions in terms of the best and average results obtained (Table 6.7), it is clear that $NP^{init} = 100$ was better than all the other variants. The Wilcoxon signed-rank test showed that $NP^{init} = 100$ was significantly better than $NP^{init} = 50$, $NP^{init} = 125$ and $NP^{init} = 150$, but there was no significance difference with $NP^{init} = 75$.

Also, a Friedman test was carried out to rank all the variants. From the results shown in Table 6.8, it is clear that those with $NP^{init} = 100$ and $NP^{init} = 75$, ranked first and second respectively.

Furthermore, the performance profiles is presented in Figure 6.3, to compare

Table 6.8: Average ranks of different values of NP^{init}, obtained from Friedman test based on both best and average results

Algorithms	Average rank based on best results obtained	Average rank based on average results obtained	Overall rank
$NP^{init} = 50$	2.94	3.35	6.29
$NP^{init} = 75$	**2.72**	2.72	5.44
$NP^{init} = 100$	2.86	**2.33**	**5.19**
$NP^{init} = 125$	3.08	2.89	5.97
$NP^{init} = 150$	3.39	3.53	6.92

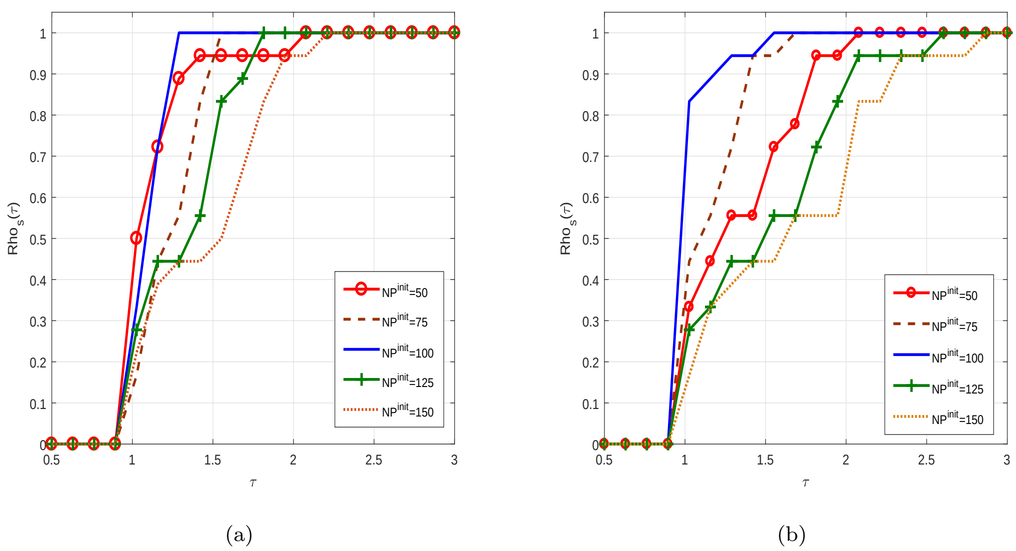


Figure 6.3: Comparison of performance profiles of proposed algorithm with different NP^{init} values, based on average (a) computational time; and (b) FEs

the different variants based on both the average computational time and FEs, from which it is clear that the variant with NP^{init}=100 is the best.

6.3.1.5 Effect of NP^{min}

For this analysis, the proposed MEA-LS-R2S was run with different NP^{min} values of 4, 10, 20, 30 and 40 individuals, with the detailed results shown in Appendix D, Table D.5. The average total computational times and FEs presented in Table 6.9 show that $NP^{min} = 4$ required the least average computational time and the fewest average FEs.

Table 6.9: Computational times and average FEs of different values for NP^{min}

Algorithms	Total average time	Total average FEs
$NP^{min} = 4$	**7.38**	**110549**
$NP^{min} = 10$	7.68	112180
$NP^{min} = 20$	7.75	111082
$NP^{min} = 30$	7.47	112196
$NP^{min} = 40$	7.40	112920

Table 6.10: Average ranks of different values of NP^{min}, obtained from Friedman test based on both best and average results

Algorithms	Average rank based on best results obtained	Average rank based on average results obtained
$NP^{min} = 4$	**2.92**	**2.44**
$NP^{min} = 10$	3.06	3.06
$NP^{min} = 20$	3.06	3.11
$NP^{min} = 30$	**2.92**	3.31
$NP^{min} = 40$	3.06	3.08

Table 6.11: Comparisons of summary of different variants of the proposed algorithm with different NP^{min} values

Algorithms	Best results				Average results			
	Better	Equal	Worse	Dec.	Better	Equal	Worse	Dec.
$NP^{min} = 4$ vs. $NP^{min} = 10$	1	17	0	$\approx$	7	9	2	+
$NP^{min} = 4$ vs. $NP^{min} = 20$	1	17	0	$\approx$	5	11	2	$\approx$
$NP^{min} = 4$ vs. $NP^{min} = 30$	0	18	0	$\approx$	7	11	0	+
$NP^{min} = 4$ vs. $NP^{min} = 40$	1	17	0	$\approx$	6	11	1	$\approx$

For further analysis, a Friedman test was done to rank all the different variants. The results are shown in Table 6.10. From which it is clear that $NP^{min} = 4$ ranked first.

Considering the quality of solutions, Table 6.11 presents a comparison between the proposed MEA-LS-R2S with $NP^{min} = 4$ and all other variants, in which it is clear that $NP^{min} = 4$ is better than all the other variants, especially in terms of average results obtained. Regarding the Wilcoxon signed-rank test, the proposed algorithm with $NP^{min} = 4$, was significantly better than $NP^{min} = 10$ and $NP^{min} = 30$, but was not significantly better than $NP^{min} = 20$ and $NP^{min} = 40$. Based on the above analysis, the proposed MEA-LS-R2S with $NP^{min} = 4$ is slightly better than all other variants.

6.3.2 Comparisons of MEA-LS-R2S and state-of-the-art algorithms

As, from the above-mentioned analyzes, it was found that MEA-LS-R2S with linear population size reduction, $CS = 25$, $NP^{init} = 100$ and $NP^{min} = 4$ was the best variant, its performance for solving the benchmark problems described in Chapter 4, Section 4.2, was compared with several state-of-the-art algorithms.

6.3.2.1 Comparisons of MEA-LS-R2S and state-of-the-art algorithms for CEC2006

To judge the performance of the the proposed (MEA-LS-R2S) algorithm, comparisons were carried out using the state-of-the-art algorithms to solve CEC2006, namely, adaptive hybrid DE algorithm (AH-DEa) [211], self-adaptive multi-operator genetic algorithm (SAMO-GA) [24], self-adaptive algorithm with multi-operator strategy (SAMO-DE) [24], improved version of SAMO-DE (ISAMODE-CMA) [47], adaptive penalty formulation with GA (APF-GA) [130], EP based on ensemble of constraint-handling techniques ECHT-EP2 [146], DE based on ensemble of constraint-handling techniques ECHT-DE [146], rank based multi-operator DE algorithm (rank-iMDDE) [212], ϵDEg with Gradient-Based Mutation ϵDE [141], artificial immune system based approach for COPs (AIS-ZYH) [213], modified JADE with R2S (MJADE-R2S) described in Chapter 5, and modified landscape-based multi-operator DE (MLS-MODE) described in Chapter 4.

Detailed results obtained from MEA-LS-R2S, based on 200000 FEs and the state-of-the-art algorithms, are presented in Appendix D, (Tables D.6, D.7, D.8, D.9, D.10, D.11), which shows the known optimal solutions for each problem and the mean and standard deviation (Std.) results obtained from 25 runs.

It must be mentioned, that whereas MEA-LS-R2S used 200000 FEs, ISAMODE-CMA, SAMO-DE, ECHT-EP2, ECHT-DE2, SAMO-GA, rank-iMDDE and AH-DEa used 240000, and APF-GA, MDE and ϵDEg used 500000. Also, as all the algorithms solved 22 of the 24 test problems, the analysis is based on those 22. The proposed MEA-LS-R2S algorithm was able to obtain optimal solutions for all of these 22 problems with 100% feasibility and success rates.

Table 6.12 presents a comparison of these algorithms. MEA-LS-R2S was better than AH-DEa, SAMO-GA, SAMO-DE, AIS-ZHY, ECHT-EP2, APF-GA, ECHT-DE, ISAMODE-CMA, ϵDEg, rank-iMDDE, MJADE-R2S and MLS-MODE for 7, 11, 8, 6, 6, 10, 6, 2, 1, 2, 8 and 7 test problems, respectively, but inferior to SAMO-GA and APF-GA for 2 and 1, respectively.

Considering the Wilcoxon signed-rank test, MEA-LS-R2S was significantly better than AH-DEa, SAMO-GA, SAMO-DE, AIS-ZHY, ECHT-EP2, ECHT-DE, APF-GA, MJADE-R2S and MLS-MODE. Although there was no significant difference between it and ISAMODE-CMA, DEg and rank-iMDDE, there was a bias towards it in terms of its number of better functions. One advantage of MEA-LS-R2S was its capability to reach optimal solutions faster than ISAMODE-CMA, ϵDEg and rank-iMDDE. In summary, the average numbers of FEs consumed by MEA-LS-R2S, MLS-MODE, ISAMODE-CMA, DEg and rank-iMDDE were 42332, 53377, 76420, 79659 and 65,906, respectively, which means that MEA-LS-R2S took 20.69%, 44.61%, 46.86% and 35.77% fewer FEs than MLS-MODE, ISAMODE-CMA, ϵDEg and rank-iMDDE, respectively.

For further analysis, a Friedman test is carried out to rank all algorithms, with their rank is presented in Table 6.13. The highest ranking is shown in boldface.

Table 6.12: Summary of comparisons of the proposed MEA-LS-R2S and the state-of-the-art algorithms, where 'Dec.' statistical decision based on Wilcoxon signed-rank test results

Algorithms	**Better**	**Equal**	Worse	Dec.
MEA-LS-R2S vs. AH-DEa	7	15	0	+
MEA-LS-R2S vs. SAMO-GA	11	9	2	+
MEA-LS-R2S vs. SAMO-DE	8	14	0	+
MEA-LS-R2S vs. AIS-ZHY	6	16	0	+
MEA-LS-R2S vs. ECHT-EP2	6	16	0	+
MEA-LS-R2S vs. ECHT-DE	6	16	0	+
MEA-LS-R2S vs. APF-GA	10	11	1	+
MEA-LS-R2S vs. ISAMODE-CMA	2	20	0	$\approx$
MEA-LS-R2S vs. ϵDEg	1	21	0	$\approx$
MEA-LS-R2S vs. rank-iMDDE	2	20	0	$\approx$
MEA-LS-R2S vs. MJADE-R2S	8	14	0	+
MEA-LS-R2S vs. MLS-MODE	7	15	0	+

Table 6.13: Average ranking of MEA-LS-R2S, EHCT-DE, AIS-ZHY,ISMOADE-CMA, SAMO-DE, ECHT-EP2, ϵDEg, AH-DEa, SAMO-GA, APF-GA, rank-iMDDE, MJADE-R2S and MLS-MODE by the Friedman test for the 22 functions in terms of mean value

Algorithms	**Mean rank**
MEA-LS-R2S	**5.43**
EHCT-DE	7.43
AIS-ZYH	6.82
ISAMODE-CMA	5.98
SAMODE	7.66
ECHT-EP2	6.80
ϵDE	5.50
AH-DEa	7.61
SAMO-GA	8.95
APF-GA	8.25
Rank-iMDDE	5.89
MJADE-R2S	6.91
MLS-MODE	7.61

As seen, MEA-LS-R2S obtained the best ranking, while ϵDE came the second.

6.3.2.2 Comparisons of MEA-LS-R2S and state-of-the-art algorithms for CEC2010

The proposed MEA-LS-R2S was also compared with the state-of-the-art algorithms for solving CEC2010, namely, constrained DE with an archive and gradient-based mutation (ϵDEag) [214], which won the CEC2010 COP competition, self-adaptive multi-operator DE (SAMODE), DE combined with DE-DBmax (DE-DBmax) [215], co-evolutionary comprehensive learning particle swarm optimizer (Co-CLPSO) [216], adaptive ranking mutation operator-based DE (ECHT-ARMOR-DE) [179], elitist artificial bee colony (eABC) [217], multi-operator GA (SAMO-GA) [39], constraint-consensus mutation based DE (DEbavCC) [218], modified JADE with R2S described in Chapter 5, and modified landscape-based multi-operator DE (MLS-MODE) described in Chapter 4.

Detailed results obtained from them based on $20000 \times D$ FEs, shown in Appendix D, Tables D.12, D.13, D.14, D.15, D.16, D.17.

Regarding the feasibility rate, MEA-LS-R2S, SAMO-DE, DE-DBmax and DE-bavDBmax were able to obtain a 100% feasibility rate for the $10D$ and $30D$ test problems while ϵDEag achieved a 100% feasibility ratio for the $10D$ and for only 35 of 36 for $30D$ test problems, with only a 12% rate for C12 with $30D$. Co-CLPSO was able to achieve 94.4% and 87.3% for the $10D$ and $30D$ test problems respectively, with the others were also unable to produce a 100% ratio.

In terms of the quality of solutions, a summary of the results is presented in Table 6.14. For $10D$, MEA-LS-R2S was better than DEbavDBmax, SAMODE, ϵDEag, DE-DBmax, Co-CLPSO, eABC, SAMO-GA, ECHT-ARMOR-DE, MJADE-R2S and MLS-MODE for 4, 1, 5, 6, 13, 16, 10, 7, 8 and 4 test problems respectively, but inferior to these algorithms for 0, 1, 1, 0, 0, 1, 1, 0, 0 and 0, respectively for the best results. In terms of the average results for $10D$, MEA-LS-R2S was better than DEbavDBmax, SAMODE, ϵDEag, DE-DBmax, Co-CLPSO, eABC, SAMO-GA, ECHT-ARMOR-DE, MJADE-R2S and MLS-MODE for 11, 10, 7, 15, 17, 17, 15, 9, 13 and 5 test problems, respectively, but inferior to these algorithms for 2, 5, 3, 1, 1, 1, 3, 4, 2 and 2, respectively.

For $30D$, in terms of the best results, MEA-LS-R2S was superior to DEbavDBmax, SAMODE, ϵDEag, DE-DBmax, eABC, Co-CLPSO, SAMO-GA, ECHT-ARMOR-DE, MJADE-R2S and MLS-MODE for 6, 12, 16, 7, 16, 15, 13, 11, 8 and 5 test problems, respectively, but inferior to these algorithms for 3, 4, 1, 3, 2, 2, 4, 3 3 and 1, respectively, while for the average results MEA-LS-R2S was better than DEbavDBmax, SAMODE, ϵDEag, DE-DBmax, eABC, Co-CLPSO, SAMO-GA, ECHT-ARMOR-DE, MJADE-R2S and MLS-MODE for 12, 13, 16, 13, 17, 16, 14, 14, 14 and 19, respectively, but inferior to these algorithms for 4, 5, 2, 3, 1, 2, 4, 3, 3 and 4, respectively.

Regarding the Wilcoxon signed-rank test, for the $10D$ test problems, MEA-LS-R2S was significantly better than DEbavDBmax, DE-DBmax, eABC, Co-CLPSO, SAMO-GA and MJADE-R2S in terms of best and average results obtained and ECHT-ARMOR-DE and MLS-MODE in terms of best results obtained, while there is no significance difference between MEA-LS-R2S and SAMODE and ϵDEag. For the $30D$ test problems, MEA-LS-R2S was significantly better than ϵDEag, eABC, Co-CLPSO, ECHT-ARMOR-DE and MJADE-R2S in terms of both the best and average results obtained, and DE-DBmax and SAMO-GA for average results obtained, while there is no significance difference in the remaining cases.

For further analysis, a Friedman test was used to rank all the algorithms according to their best and averages results, presented in Table 6.15 and Figure

Table 6.14: Summary of comparison of MEA-LS-R2S and state-of-the-art algorithms for CEC2010

	Algorithms	Criteria	Better	Equal	Worse	Dec.
10D	MEA-LS-R2S vs. DEbavDBmax	Best	4	14	0	+
		Average	11	5	2	+
	MEA-LS-R2S vs. SAMODE	Best	1	16	1	≈
		Average	10	3	5	≈
	MEA-LS-R2S vs. ϵDEag	Best	5	12	1	≈
		Average	7	8	3	≈
	MEA-LS-R2S vs. DE-DBmax	Best	6	12	0	+
		Average	15	2	1	+
	MEA-LS-R2S vs. eABC	Best	16	1	1	+
		Average	17	0	1	+
	MEA-LS-R2S vs. Co-CLPSO	Best	14	4	0	+
		Average	17	0	1	+
	MEA-LS-R2S vs. SAMO-GA	Best	10	7	1	+
		Average	15	0	3	+
	MEA-LS-R2S vs. ECHT-ARMOR-DE	Best	7	11	0	+
		Average	9	5	4	≈
	MEA-LS-R2S vs. MJADE-R2S	Best	8	10	0	+
		Average	13	3	2	+
	MEA-LS-R2S vs. MLS-MODE	Best	4	14	0	+
		Average	5	11	2	≈
30D	MEA-LS-R2S vs. DEbavDBmax	Best	6	9	3	≈
		Average	12	2	4	≈
	MEA-LS-R2S vs. SAMODE	Best	12	2	4	≈
		Average	13	0	5	≈
	MEA-LS-R2S vs. ϵDEag	Best	16	1	1	+
		Average	16	0	2	+
	MEA-LS-R2S vs. DE-DBmax	Best	7	8	3	≈
		Average	13	2	3	+
	MEA-LS-R2S vs. eABC	Best	16	0	2	+
		Average	17	0	1	+
	MEA-LS-R2S vs. Co-CLPSO	Best	15	1	2	+
		Average	16	0	2	+
	MEA-LS-R2S vs. SAMO-GA	Best	13	1	4	≈
		Average	14	0	4	+
	MEA-LS-R2S vs. ECHT-ARMOR-DE	Best	11	4	3	+
		Average	14	1	3	+
	MEA-LS-R2S vs. MJADE-R2S	Best	8	7	3	+
		Average	14	1	3	+
	MEA-LS-R2S vs. MLS-MODE	Best	5	12	1	≈
		Average	10	4	4	≈

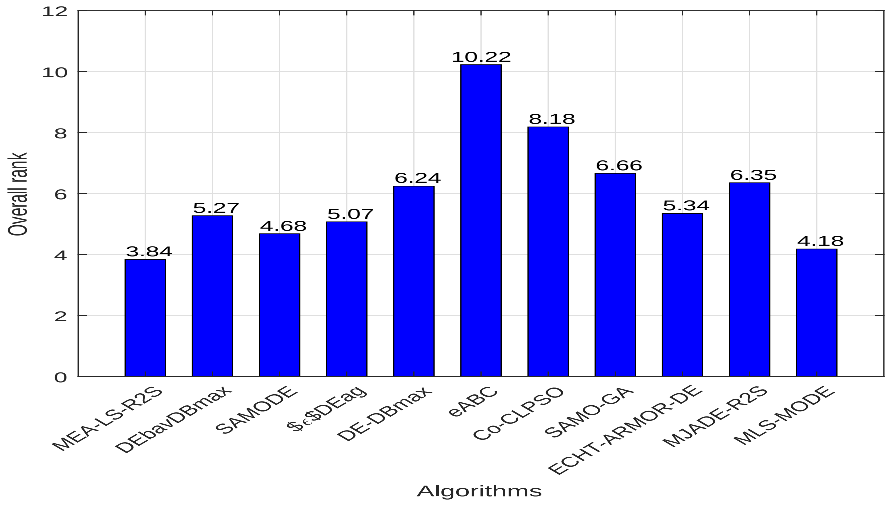
Table 6.15: Average rankings of MEA-LS-R2S, DEbavDBmax, SAMODE, ϵDEag, DE-DBmax, eABC, Co-CLPSO, SAMO-GA and ECHT-ARMOR-DE, MLS-MODE and MJADE-R2S obtained from Friedman test

	Algorithms	Average rank based on best results obtained	Average rank based on, average results obtained	Overall rank
10D	MEA-LS-R2S	**4.25**	3.42	**3.84**
	DEbavDBmax	5.11	5.42	5.27
	SAMODE	4.44	4.92	4.68
	ϵDEag	5.47	4.67	5.07
	DE-DBmax	5.69	6.78	6.24
	eABC	10.19	10.25	10.22
	Co-CLPSO	8.14	8.22	8.18
	SAMO-GA	6.92	6.39	6.66
	ECHT-ARMOR-DE	5.92	4.75	5.34
	MJADE-R2S	4.86	7.83	6.35
	MLS-MODE	5.00	**3.36**	4.18
30D	MEA-LS-R2S	4.00	**3.08**	**3.54**
	DEbavDBmax	4.03	3.86	3.95
	SAMODE	5.67	4.78	5.23
	ϵDEag	8.53	6.64	7.59
	DE-DBmax	4.22	4.64	4.43
	eABC	9.83	10.22	10.03
	Co-CLPSO	8.50	8.58	8.54
	SAMO-GA	7.03	6.11	6.57
	ECHT-ARMOR-DE	5.78	7.03	6.41
	MJADE-R2S	4.61	7.75	6.18
	MLS-MODE	**3.81**	3.70	3.76

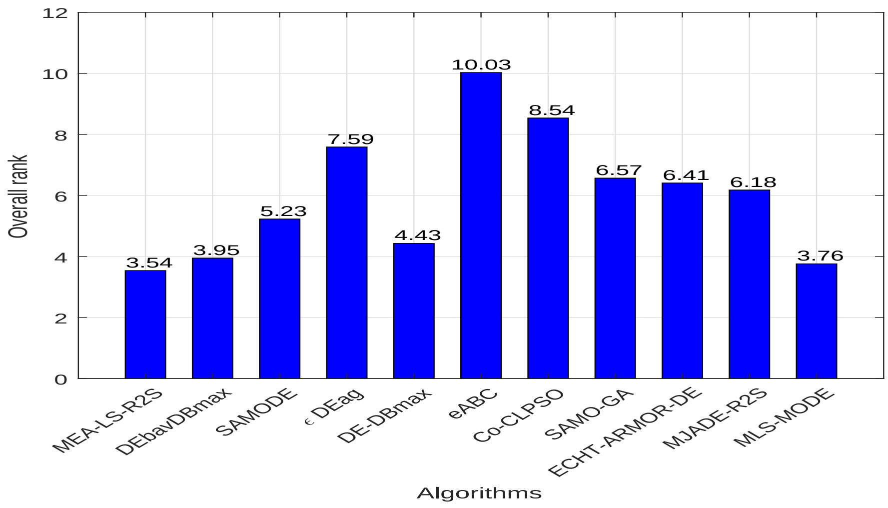
6.4 in which it can be seen that MEA-LS-R2S ranked first of the 9 algorithms for both $10D$ and $30D$.

6.3.2.3 Comparisons of MEA-LS-R2S and state-of-the-art algorithms for CEC2011

In this section the proposed MEA-LS-R2S algorithm is judged by solving the 10 real-world application problems taken from the CEC2011 [181] competition on real-world optimization problems. Its performance is compared with the state-of-the-art algorithms, namely, continuous DE ant-stigmergy algorithm (CDASA) for solving CEC2011 competition problems [219], adaptive DE algorithm (ADE) for solving CEC2011 competition problems [220], ensemble DE algorithm (EPSDE)



(a)



(b)

Figure 6.4: Overall rankings of MEA-LS-R2S, DEbavDBmax, SAMODE, ϵDEag, DE-DBmax, eABC, Co-CLPSO, SAMO-GA, ECHT-ARMOR-DE, MJADE-R2S and MLS-MODE obtained from Friedman test (a) based on $10D$; and (b) based on $30D$

for solving CEC2011 competition problems [221], DE with multi-population strategies (SAMODE) for solving CEC2011 competition problems [23], DE with adaptive crossover rate (DE-Acr) and a competitive DE with local search for solving CEC2011 competition problems [223], modified DE with local search (CDELS) for solving CEC2011 competition problems [222], hybrid EA-DE-Memetic Algorithm (EA-DE-MA) for solving CEC2011 competition problems [228], improved version of SAMO-DE (ISAMODE-CMA) [47], modified JADE with R2S described in Chapter 5, and modified landscape-based multi-operator DE (MLS-MODE) described in Chapter 4.

Detailed results obtained from them based on 150000 FEs shown in Appendix D, Tables D.18, D.19, D.20. Note that all the state-of-the-art algorithms used in this comparison, solve these test problems as unconstrained optimization problems, and there are no details about whether the optimal solution obtained is feasible or not.

Considering the quality of solutions, a summary of the results is presented in Table 6.16. In terms of the best results, MEA-LS-R2S was better than CDASA, ADE, EPSDE, SAMODE, DE-Acr, CDELS, EA-DE-MA, ISAMODE-CMA, MJADE-R2S and MLS-MODE for 10, 10, 7, 8, 8, 10, 10, 8, 7 and 6, respectively, but inferior to these algorithms for 0, 0, 2, 2, 2, 0, 0, 2, 3 and 3, respectively. In regards to the mean results, MEA-LS-R2S was superior to CDASA, ADE, EPSDE, SAMODE, DE-Acr, CDELS, EA-DE-MA, ISAMODE-CMA, MJADE-R2S and MLS-MODE for 10, 10, 8, 8, 7, 10, 10, 8, 7 and 7, respectively, but worse than these algorithms for 0, 0, 2, 2, 3, 0, 0, 2, 2 and 2, respectively.

Based on the Wilcoxon signed-rank test, it was clear that MEA-LS-R2S was significantly better than all algorithms, except SAMODE, ISAMODE-CMA and MLS-MODE for best results and DE-Acr and MJADE-R2S for both best and average results.

Also, a Friedman test was conducted to rank all the algorithms in terms of both best and average results, with the ranks presented in Table 6.17 and Figure

Table 6.16: Summary of comparison of MEA-LS-R2S and state-of-the-art algorithms CEC2011

Algorithms	Best results				Mean results			
	Better	Equal	Worse	Dec.	Better	Equal	Worse	Dec.
MEA-LS-R2S vs. CDASA	10	0	0	+	10	0	0	+
MEA-LS-R2S vs. ADE	10	0	0	+	10	0	0	+
MEA-LS-R2S vs. EPSDE	7	1	2	+	8	0	2	+
MEA-LS-R2S vs. SAMODE	8	0	2	≈	8	0	2	+
MEA-LS-R2S vs. DE-Acr	8	0	2	≈	7	0	3	+
MEA-LS-R2S vs. CDELS	10	0	0	+	10	0	0	+
MEA-LS-R2S vs. EA-DE-MA	10	0	0	+	10	0	0	+
MEA-LS-R2S vs. ISAMODE-CMA	8	0	2	≈	8	0	2	+
MEA-LS-R2S vs. MJADE-R2S	7	0	3	≈	7	1	2	≈
MEA-LS-R2S vs. MLS-MODE	6	1	3	≈	7	1	2	+

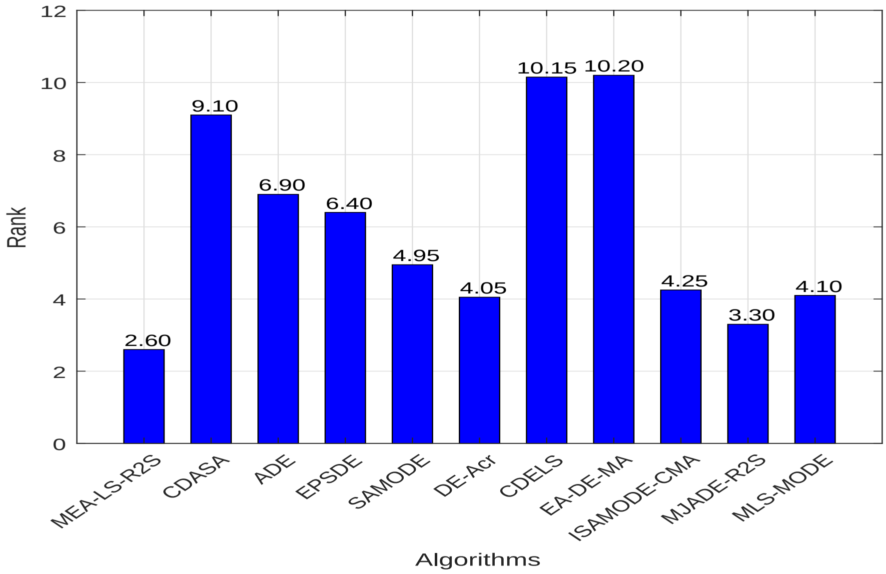
Table 6.17: Average rankings of MEA-LS-R2S, CDASS, ADE, EPSDE, SAMODE, DE-Acr, CDELS, EA-DE-MA, ISAMODE-CMA, MJADE-R2S and MLS-MODE obtained from Friedman test

Algorithms	Average rank based on best results obtained	Average rank based on average results obtained
MEA-LS-R2S	**2.60**	**2.35**
CDASA	9.10	9.40
ADE	6.90	7.50
EPSDE	6.40	6.50
SAMODE	4.95	5.50
DE-Acr	4.05	3.50
CDELS	10.15	9.25
EA-DE-MA	10.20	10.40
ISAMODE-CMA	4.25	3.90
MJADE-R2S	3.30	3.35
MLS-MODE	4.10	4.35

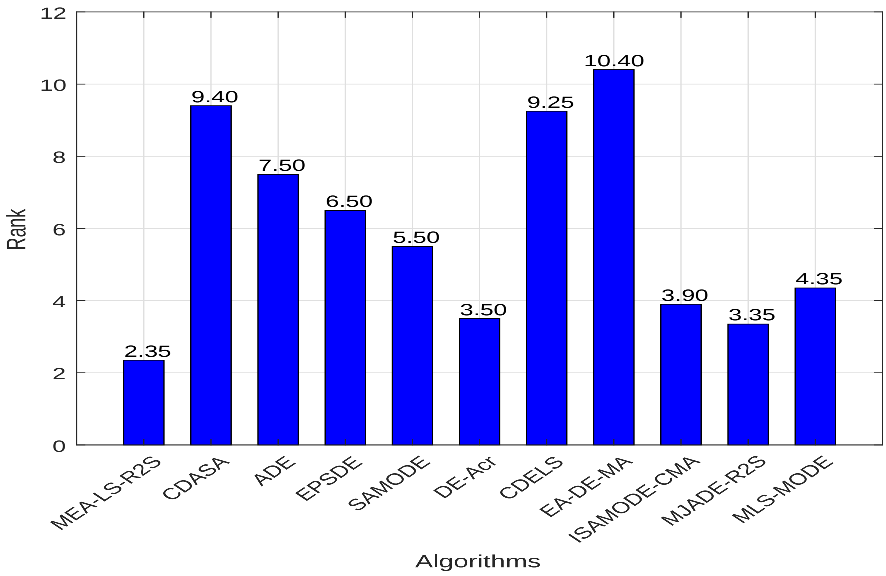
6.5. From which it is clear that the proposed MEA-LS-R2S was ranked first.

Furthermore, the performance profiles depicted in Figure 6.6 show that MEA-LS-R2S was the best for both best and average results.

Also, the proposed MEA-LS-R2S is able to obtain better solutions faster than MJADE-R2S and MLS-MODE. In summary, the average computational time that are taken by MEA-LS-R2S is 61.7, while MJADE-R2S and MLS-MODE take 73.29 and 87.97, respectively, which means that MEA-LS-R2S is able to save 15.70% and 29.77% of computational time in comparison to MJADE-R2S and MLS-MODE,

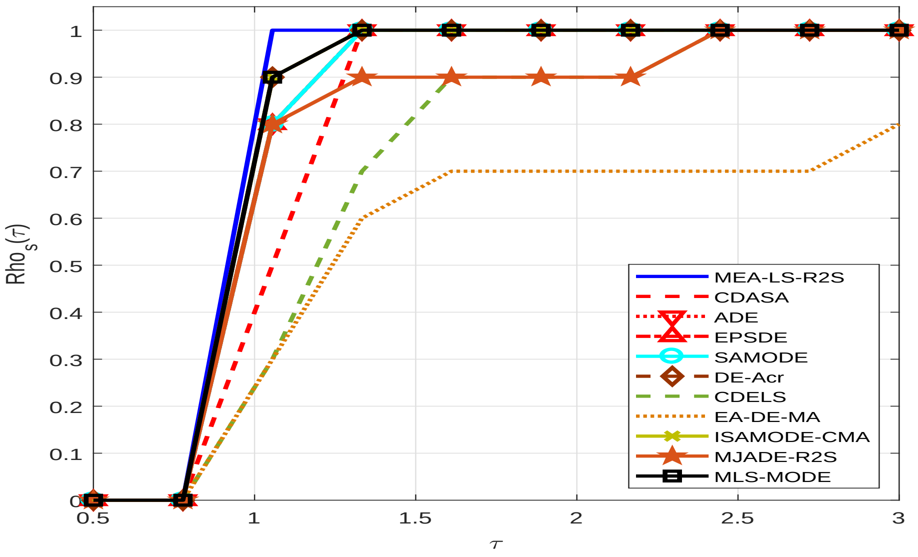


(a)

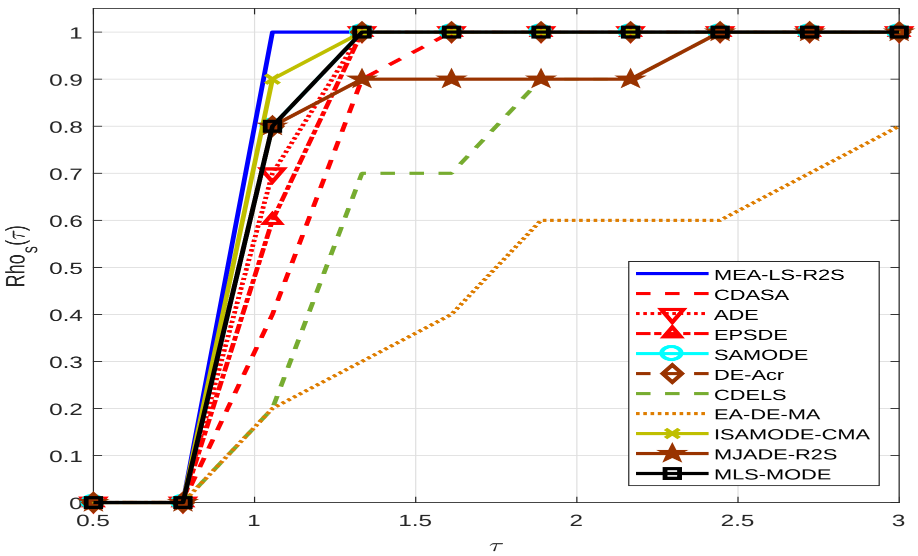


(b)

Figure 6.5: Rankings of MEA-LS-R2S, CDASS, ADE, EPSDE, SAMODE, DE-Acr, CDELS, EA-DE-MA, ISAMODE-CMA, MJADE-R2S and MLS-MODE obtained from Friedman test (a) based on best results; and (b) based on average results



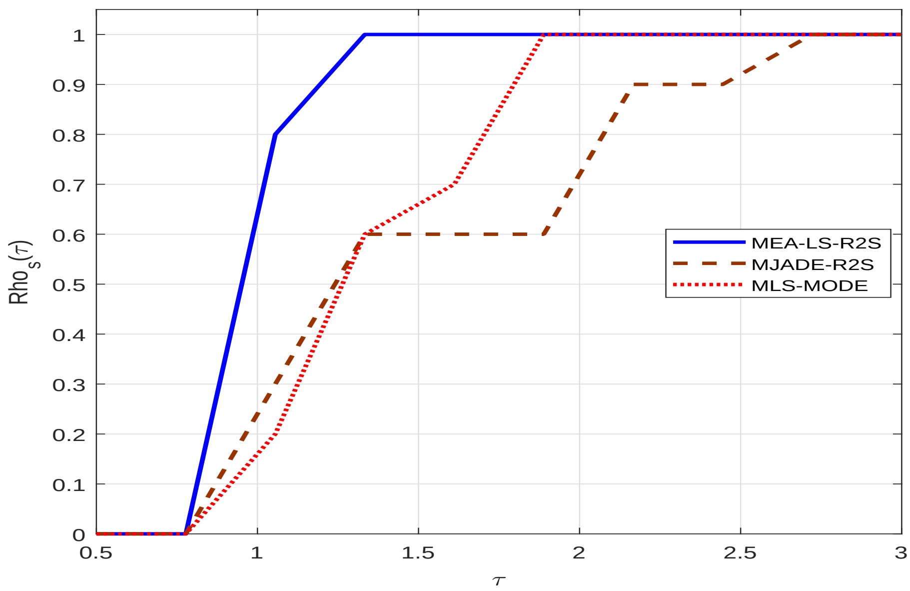
(a)



(b)

Figure 6.6: Comparison of performance profiles of MEA-LS-R2S, CDASA, ADE, EPSDE, SAMODE, DE-Acr, CDELS, EA-DE-MA, ISAMODE-CMA, MJADE-R2S and MLS-MODE (a) based on best results; and (b) based on average results

respectively. The performance profiles depicted in Figure 6.7 show that MEA-LS-R2S was the best in terms of computational time.



(a)

Figure 6.7: Comparison of performance profiles of MEA-LS-R2S, MJADE-R2S and MLS-MODE based on the average computational time

6.4 Chapter Summary

Motivated by the encouraging results obtained from the proposed landscape based algorithms in Chapters 3 and Chapter 4, and the proposed search space reduction mechanism in Chapter 5, in this chapter, a new multi-EA algorithms, which utilizes information from the problem's landscape and search space, namely MEA-LS-R2S, was proposed.

MEA-LS-R2S was tested by solving 22 test problems taken from CEC2006, 36 test problems taken from CEC2010, and 10 test problems taken from CEC2011. The results from MEA-LS-R2S were compared with those from state-of-the-art algorithms, and based on the quality of solutions obtained and non-parametric statistical testing, the results showed the superiority of the proposed MEA-LS-R2S.

Finally, based on the quality of solutions obtained, and the non-parametric statistical test results, MEA-LS-R2S was compared with the proposed methods in Chapter 4 and Chapter 5 (JADE-R2 and MLS-MODE, respectively), which demonstrated its superiority in all cases.

Chapter 7

Conclusions and Future Research Directions

The research conducted and discussed in this thesis is summarized in this chapter, and its conclusions and suggestions for possible future work presented.

7.1 Summary of Research Conducted

In this section, a summary of the research conducted in this thesis is presented.

7.1.1 summary of research conducted in chapters one and two

As discussed in Chapters 1 and 2, optimization problems, which can be classified as unconstrained or constrained, have different characteristics and mathematical properties that make the process of finding optimal solutions to them difficult. Of the existing solutions techniques developed, evolutionary algorithms (EAs) have demonstrated great success. However, researchers and practitioners acknowledged that, the performances of such methods are highly dependent on an algorithm's design, and the choice of its search operators and control parameters. Also, the most suitable EA and/or its search operators for solving a particular problem is often determined using a trail-and-error approach rather than any useful information about the problems landscape. Furthermore, one EA, search operator and/or set of parameters well suited for a certain set of problems may not work well for another, as is evident in the literature. Although fitness landscape analysis (FLA) has been widely used to judge a problem's complexity, it has not been fully explored in relation the designs of algorithms. Also, in most COPs, the optimal solution lies on the intersection of its active constraints. This basic information may help to concentrate the search process in certain regions instead of the entire search space and speed up the process of locating the optimal solution.

As discussed in Chapter 1, the main objective of this study was to use information from the problem landscape and search space to boost the performances of EAs. To achieve this main goal, four related aspects were introduced as follows: the first one was to propose a new algorithm that uses landscape information to select the most suitable DE, from a pool of them, during the evolutionary process for solving unconstrained optimization problems. The second was to propose an algorithm which utilizes the strength of more than one DE and choose the best one based on the objective function and constraints landscapes for solving COPs. The third was to propose a mechanism for reducing the search space by focusing the search on the area around the active constraints. The fourth was to propose an algorithm that uses multiple EAs and both a reduced search space mechanism and landscape information to solve COPs. These four goals have been achieved in Chapters 3 to 6.

7.1.2 summary of research conducted in chapter three

In Chapter 3, an algorithm that uses the function landscape to choose the best-performing differential evolution (DE) from a set was developed. In it, several different DE operators are used to evolve the entire population for a pre-defined number of generations (CS). Concurrently, the function landscape and success rate of each DE operator are calculated and stored in every generation, with the best-performing one based on their normalized measures used to evolve the entire population for the subsequent cycle, a process which continues until the stopping criteria are satisfied. The proposed algorithm was used to solve two specialized sets of unconstrained optimization problems and an analysis of its components undertaken. Its results were presented and compared with those of seven of its variants as well as several state-of-the-art algorithms.

7.1.3 summary of research conducted in chapter four

In Chapter 4, a landscape-based algorithm for solving COPs, in which information from the objective function and constraint landscapes is used to choose the best operator from a pool during the evolutionary process, was proposed. It starts with a random population generated by the Latin hyper-cube design mechanism and then several DE mutation strategies are used to evolve the entire population. After a certain number of generations, the best strategy based on the problems landscape is selected to evolve subsequent populations, a process that continues until the stopping condition is met. Its different components were analyzed and then the experimental results obtained from it and several state-of-the-art algorithms for solving different sets of COPs were compared.

7.1.4 Summary of research conducted in chapter five

In Chapter 5, a new reduced search space (R2S) mechanism, which automatically identifies the active constraints based on the current solutions and uses this information to determine the most promising boundary zone for searching, was proposed. Its different components were analyzed and it was integrated in six state-of-the-art algorithms for solving COPs, with the experimental results obtained from algorithms both with and without it compared.

7.1.5 Summary of research conducted in chapter six

In Chapter 6, a new multi-EA algorithm that uses information from the search space and problem's landscape was proposed. Its different components were analyzed and then the experimental results obtained from it and other state-of-the-art algorithm for solving different sets of COPs compared.

7.2 Summary of Conclusions

The conclusions drawn from this study can be summarized as follows.

7.2.1 Landscape-based algorithm for unconstrained problems

In chapter 3, 11 variants of DE mutation strategies were tested by solving 30 unconstrained optimization problems. The results obtained showed that there was no single one was able to attain the best results for all test problems. However, the DE/current-to-φbest/1 without archive was better than all others, with the DE/current-to-φbest/1 archive ranked second and DE/φbest/1 third.

Based on the five parametric analysis performed in this chapter, the best parameters were set as follows: the number of DE operators (m) initially of 5 and reduced from 5 to 1 every two cycles, NP^{init} $25 \times D$, NP^{min} of 7 and CS 150.

The results obtained from the proposed LSAOS-DE and seven of its variants, with the only difference among them their selection mechanisms, were compared. LSAOS-DE was shown to perform competitively against the others and be capable of saving 15.1% of computational time and 15.7% of the number of fitness evaluations (FEs). Also, compared with several state-of-the-art algorithms, it was either competitive or better.

7.2.2 Landscape-based algorithm for constrained problems

In Chapter 4, the results obtained by MLS-MODE for solving three specialized COPs were presented and analyzed.

Based on the parametric analysis, the best parameters were set as NP^{init} 150, NP^{min} 40, CS 125, with the self-adaptation of the control parameters proven to be better than using fixed values.

MLS-MODE showed its superiority over four of its variants, the only differences among which were their selection mechanisms, by obtaining savings in computational time and the number of FEs of up to 69% and 33%, respectively. Compared with several state-of-the-art algorithms, it was more competent and capable of reaching 100% feasibility ratios for all the test problems.

7.2.3 Reduced search space mechanism for constrained problems

In Chapter 5, a R2S mechanism for constrained problems, which was easily incorporated in six well-known algorithms, was proposed.

It was capable of determining the active constraints with a 96% success rate and helped the algorithms improve the quality of solutions, with savings in computational time and the number of FEs of up to 36.69% and 11.88%, respectively. Also, it could be easily integrated in other algorithms used to solve COPs.

7.2.4 Multi-EA framework for constrained problems

In Chapter 6, to take advantage of the benefits of both a problem's landscape and the R2S mechanism, a new multi-EA (MEA-LS-R2S) algorithm was proposed.

MEA-LS-R2S was tested and analyzed by solving three specialized COPs.

Firstly, based on a parametric analysis, it was concluded that it is better to use a NP^{init} of 100, NP^{min} of 4, CS of 25, m of 2 and mechanism for linearly reducing the population than a fixed population size.

Secondly, based on the results obtained from three variants of the proposed algorithm, it was found that Var3 (that used MODE and CMA-ES) was better than Var2 (that used MODE, MOGA and CMA-ES) and Var1 (that used MOGA and CMA-ES). Also, Var3 was able to save up to 44.5% of computational time and 21.3% of the number of FEs compared with those of the other two variants.

Finally, the proposed MEA-LS-R2S was either competitive with or better than several state-of-the-art algorithms and capable of reaching 100% feasibility ratios for all the test problems.

7.3 Link to Real-World Applications

In chapters 4, 5 and 6, the performance of the proposed algorithms were tested on 10 real-world applications problems, in which the proposed algorithms were either competitive with or better than many state-of-the-art algorithms. Also, in another work, not included in this thesis, an algorithm that uses the landscape information to choose the best-performing DE operators, among many, was proposed and used to solve a set of real-world applications problems (TP-MODE). The results obtained from TP-MODE was compared with those obtained from many of the state-of-the-art algorithms, showing that the performance of TP-MODE was better. The whole paper is presented in Appendix E.

7.4 Future Research Directions

The algorithms developed in this thesis could be extended by:

1. testing LSAOS-DE on more unconstrained optimization problems;
2. using more than one landscape measure to choose the most suitable EA and/or its search operators during the evolutionary process;
3. developing other landscape-based CI algorithms, such as GA and particle swarm optimization;
4. applying all the algorithms to solving large-scale optimization problems using decomposition techniques;
5. adopting all the algorithms to solve multi- and many-objective problems in which interest is increasing;
6. enhancing the performances of the algorithms by using local and/or multi-local search approaches;
7. incorporating the R2S mechanism in other EAs;
8. applying the proposed algorithms to solving dynamic optimization problems;
9. implementing surrogate models in all the proposed algorithms to reduce their numbers of FEs;
10. using a problems landscape to choose the best surrogate model from a set; and/or
11. proposing an algorithm that utilizes more than one mutation strategies, more than one crossover operators and more than one parameter adaptation techniques;

Bibliography

[1] D. Reich, Y. Shi, M. Epelman, A. Cohn, E. Barnes, K. Arthurs, and E. Klampfl, “Scheduling crash tests at ford motor company,” *Interfaces*, vol. 46, no. 5, pp. 409–423, 2016.

[2] E. Aktaş, Ö. Özaydın, B. Bozkaya, F. Ülengin, and Ş. Önsel, “Optimizing fire station locations for the istanbul metropolitan municipality,” *Interfaces*, vol. 43, no. 3, pp. 240–255, 2013.

[3] T. Kutz, M. Davis, R. Creek, N. Kenaston, C. Stenstrom, and M. Connor, “Optimizing chevrons refineries,” *Interfaces*, vol. 44, no. 1, pp. 39–54, 2014.

[4] T. Varelas, S. Archontaki, J. Dimotikalis, O. Turan, I. Lazakis, and O. Varelas, “Optimizing ship routing to maximize fleet revenue at danaos,” *Interfaces*, vol. 43, no. 1, pp. 37–47, 2013.

[5] Z. Hu, J. Putz, Y. Sutjandra, A. Chan, E. Mount, and K. Baker, “The energy authority optimizes water routing and hydroelectric generation on the columbia river,” *Interfaces*, vol. 45, no. 1, pp. 43–57, 2015.

[6] I. Farasyn, S. Humair, J. I. Kahn, J. J. Neale, O. Rosen, J. Ruark, W. Tarlton, W. Van de Velde, G. Wegryn, and S. P. Willems, “Inventory optimization at procter & gamble: Achieving real benefits through user adoption of inventory tools,” *Interfaces*, vol. 41, no. 1, pp. 66–78, 2011.

[7] B. Blank, E. Lunceford, J. Morik, S. He, M. Rana, P. Rajendran, G. Gnanapandithan, K. Lajoie, B. Kats, M. Revoor *et al.*, “BNY Mellon Optimization Reduces Intraday Credit Risk by $1.4 Trillion,” *Interfaces*, vol. 47, no. 1, pp. 38–51, 2017.

[8] K.Deb, *Optimization for engineering design: Algorithms and examples.* PHI Learning Pvt. Ltd., 2012.

[9] R. Sarker, J. Kamruzzaman, and C. Newton, "Evolutionary optimization (EvOpt): a brief review and analysis," *International Journal of Computational Intelligence and Applications*, vol. 3, no. 04, pp. 311–330, 2003.

[10] L. J. Fogel, A. J. Owens, and M. J. Walsh, "Artificial intelligence through simulated evolution," 1966.

[11] L. Rutkowski, *Computational intelligence: methods and techniques.* Springer, 2008.

[12] R. Storn and K. Price, "Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[13] X. Xia and A. Elaiw, "Optimal dynamic economic dispatch of generation: A review," *Electric Power Systems Research*, vol. 80, pp. 975 – 986, 2010.

[14] P. Attaviriyanupap, H. Kita, E. Tanaka, and J. Hasegawa, "A hybrid ep and sqp for dynamic economic dispatch with nonsmooth fuel cost function," *IEEE Power Engineering Review*, vol. 22, no. 4, pp. 77–77, 2002.

[15] D. E. Golberg, "Genetic algorithms in search, optimization, and machine learning," *Addion wesley*, vol. 1989, p. 102, 1989.

[16] F. Neri and V. Tirronen, "Recent advances in differential evolution: a survey and experimental analysis," *Artificial Intelligence Review*, vol. 33, no. 1-2, pp. 61–106, 2010.

[17] S. Das, S. S. Mullick, and P. Suganthan, "Recent advances in differential evolution–an updated survey," *Swarm and Evolutionary Computation*, vol. 27, pp. 1–30, 2016.

[18] S. M. Islam, S. Das, S. Ghosh, S. Roy, and P. N. Suganthan, "An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 2, pp. 482–500, 2012.

[19] A. P. Piotrowski, "Adaptive memetic differential evolution with global and local neighborhood-based mutation operators," *Information Sciences*, vol. 241, pp. 164–194, 2013.

[20] S. M. Elsayed, R. A. Sarker, and D. L. Essam, "A self-adaptive combined strategies algorithm for constrained optimization using differential evolution," *Applied Mathematics and Computation*, vol. 241, pp. 267–282, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0096300314006742

[21] M. Z. Ali, N. H. Awad, and P. N. Suganthan, "Multi-population differential evolution with balanced ensemble of mutation strategies for large-scale global optimization," *Applied Soft Computing*, vol. 33, pp. 304–327, 2015.

[22] G. Wu, R. Mallipeddi, P. Suganthan, R. Wang, and H. Chen, "Differential evolution with multi-population based ensemble of mutation strategies," *Information Sciences*, vol. 329, pp. 329–345, 2016.

[23] S. M. Elsayed, R. A. Sarker, and D. L. Essam, "Differential evolution with multiple strategies for solving CEC2011 real-world numerical optimization problems," in *Evolutionary Computation (CEC), 2011 IEEE Congress on*. IEEE, 2011, pp. 1041–1048.

[24] S. M. Elsayed, R. A. Sarker, and D. L. Essam, "Multi-operator based evolutionary algorithms for solving constrained optimization problems," *Computers & operations research*, vol. 38, no. 12, pp. 1877–1896, 2011.

[25] K. M. Malan and A. P. Engelbrecht, "A survey of techniques for characterising fitness landscapes and some possible ways forward," *Information Sciences*, vol. 241, pp. 148–163, 2013.

[26] F. Chicano, G. Luque, and E. Alba, "Autocorrelation measures for the quadratic assignment problem," *Applied Mathematics Letters*, vol. 25, no. 4, pp. 698–705, 2012.

[27] E. Pitzer and M. Affenzeller, "A comprehensive survey on fitness landscape analysis," in *Recent Advances in Intelligent Engineering Systems.* Springer, 2012, pp. 161–191.

[28] T. Jones and S. Forrest, "Fitness distance correlation as a measure of problem difficulty for genetic algorithms." in *ICGA*, vol. 95, 1995, pp. 184–192.

[29] A. M. Sutton, D. Whitley, M. Lunacek, and A. Howe, "PSO and multi-funnel landscapes: how cooperation might limit exploration," in *Proceedings of the 8th annual conference on Genetic and evolutionary computation.* ACM, 2006, pp. 75–82.

[30] Y. Sun, S. K. Halgamuge, M. Kirley, and M. Munoz, "On the selection of fitness landscape analysis metrics for continuous optimization problems," in *Information and Automation for Sustainability (ICIAfS), 2014 7th International Conference on.* IEEE, 2014, pp. 1–6.

[31] V. K. Vassilev, T. C. Fogarty, and J. F. Miller, "Information characteristics and the structure of landscapes," *Evolutionary Computation*, vol. 8, no. 1, pp. 31–60, 2000.

[32] P. Caamano, A. Prieto, J. A. Becerra, F. Bellas, and R. J. Duro, "Real-valued multimodal fitness landscape characterization for evolution," in *International Conference on Neural Information Processing*. Springer, 2010, pp. 567–574.

[33] B. Xin, J. Chen, and F. Pan, "Problem difficulty analysis for particle swarm optimization: deception and modality," in *Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation*. ACM, 2009, pp. 623–630.

[34] M. Lunacek and D. Whitley, "The dispersion metric and the CMA evolution strategy," in *Proceedings of the 8th annual conference on Genetic and evolutionary computation*. ACM, 2006, pp. 477–484.

[35] M. R. Bonyadi and Z. Michalewicz, "On the edge of feasibility: a case study of the particle swarm optimizer," in *Evolutionary Computation (CEC), 2014 IEEE Congress on*. IEEE, 2014, pp. 3059–3066.

[36] G. Leguizamón and C. A. C. Coello, "Boundary search for constrained numerical optimization problems with an algorithm inspired by the ant colony metaphor," *Evolutionary Computation, IEEE Transactions on*, vol. 13, no. 2, pp. 350–368, 2009.

[37] Z. Michalewicz, "Genetic algorithms [and] data structures [is] evolution programs," *Artificial intelligence*, 1992.

[38] A. S. Barkat Ullah, R. Sarker, and C. Lokan, "Handling equality constraints in evolutionary optimization," *European Journal of Operational Research*, vol. 221, no. 3, pp. 480–490, 2012.

[39] S. M. Elsayed, R. A. Sarker, and D. L. Essam, "Multi-operator based evolutionary algorithms for solving constrained optimization problems," *Computers & Operations Research*, vol. 38, pp. 1877–1896, 2011.

[40] M. A. Muñoz, Y. Sun, M. Kirley, and S. K. Halgamuge, "Algorithm selection for black-box continuous optimization problems: A survey on methods and challenges," *Information Sciences*, vol. 317, pp. 224–245, 2015.

[41] A. S. Barkat Ullah, R. Sarker, and D. Cornforth, "Search space reduction technique for constrained optimization with tiny feasible space," in *Proceedings of the 10th annual conference on Genetic and evolutionary computation.* ACM, 2008, pp. 881–888.

[42] A. Rowhanimanesh and S. Efati, "A novel approach to improve the performance of evolutionary methods for nonlinear constrained optimization," *Advances in Artificial Intelligence*, vol. 2012, p. 5, 2012.

[43] S. M. E. Elsayed, "Evolutionary approach for constrained optimization," Ph.D. dissertation, University of New South Wales (UNSW) at Australian Defence Force Academy, Canberra, Australia, 2012.

[44] A. Petrowski and S. Ben-Hamida, *Evolutionary Algorithms.* John Wiley & Sons, 2017.

[45] X. Yu and M. Gen, *Introduction to evolutionary algorithms.* Springer Science & Business Media, 2010.

[46] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation,*, vol. 15, pp. 4–31, 2011.

[47] S. M. Elsayed, R. A. Sarker, and D. L. Essam, "An improved self-adaptive differential evolution algorithm for optimization problems," *IEEE Transactions on Industrial Informatics,*, vol. 9, pp. 89–99, 2013.

[48] K. Price, R. M. Storn, and J. A. Lampinen, *Differential evolution: a practical approach to global optimization.* Springer Science & Business Media, 2006.

[49] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, p. 398, 2009.

[50] A. W. Iorio and X. Li, "Solving rotated multi-objective optimization problems using differential evolution," in *Australasian Joint Conference on Artificial Intelligence.* Springer, 2004, pp. 861–872.

[51] J. Zhang and A. Sanderson, "Jade: Adaptive differential evolution with optional external archive," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 945–958, Oct 2009.

[52] E. Mezura-Montes, J. Velázquez-Reyes, and C. A. Coello Coello, "A comparative study of differential evolution variants for global optimization," in *Proceedings of the 8th annual conference on Genetic and evolutionary computation.* ACM, 2006, pp. 485–492.

[53] G. Jeyakumar and C. Shanmugavelayutham, "Convergence analysis of differential evolution variants on unconstrained global optimization functions," *International Journal of Artificial Intelligence & Applications*, vol. 2, no. 2, 2011.

[54] J. Liu and J. Lampinen, "A fuzzy adaptive differential evolution algorithm," *Soft Computing*, vol. 9, no. 6, pp. 448–462, 2005.

[55] O. Kramer, "Evolutionary self-adaptation: a survey of operators and strategy parameters," *Evolutionary Intelligence*, vol. 3, no. 2, pp. 51–65, 2010.

[56] R. Storn and K. Price, "Differential evolution a simple and efficient adaptive scheme for global optimization over continuous spaces, international computer science institute, berkeley," *Berkeley, CA*, 1995.

[57] R. Gämperle, S. D. Müller, and P. Koumoutsakos, "A parameter study for differential evolution," *Advances in intelligent systems, fuzzy systems, evolutionary computation*, vol. 10, pp. 293–298, 2002.

[58] J. Ronkkonen, S. Kukkonen, and K. V. Price, "Real-parameter optimization with differential evolution," in *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, vol. 1. IEEE, 2005, pp. 506–513.

[59] A. E. Eiben, J. E. Smith *et al.*, *Introduction to evolutionary computing*. Springer, 2003, vol. 53.

[60] W. Qian *et al.*, "Adaptive differential evolution algorithm for multiobjective optimization problems," *Applied Mathematics and Computation*, vol. 201, no. 1, pp. 431–440, 2008.

[61] D. Zaharie, "Control of population diversity and adaptation in differential evolution algorithms," in *Proc. of MENDEL*, vol. 9, 2003, pp. 41–46.

[62] Y. Lu, J. Zhou, H. Qin, Y. Wang, and Y. Zhang, "Chaotic differential evolution methods for dynamic economic dispatch with valve-point effects," *Engineering Applications of Artificial Intelligence*, vol. 24, pp. 378–387, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0952197610002009

[63] G. Chen and X. Ding, "An improved differential evolution method based on the dynamic search strategy to solve dynamic economic dispatch problem

with valve-point effects," *Abstract and Applied Analysis*, vol. 2014, p. 15, 2014. [Online]. Available: http://dx.doi.org/10.1155/2014/175417

[64] J. Zhang and A. C. Sanderson, "Jade: adaptive differential evolution with optional external archive," *IEEE Transactions on evolutionary computation*, vol. 13, no. 5, pp. 945–958, 2009.

[65] V. L. Huang, A. K. Qin, and P. N. Suganthan, "Self-adaptive differential evolution algorithm for constrained real-parameter optimization," in *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*. IEEE, 2006, pp. 17–24.

[66] Q. Fan and X. Yan, "Self-adaptive differential evolution algorithm with zoning evolution of control parameters and adaptive mutation strategies," *IEEE transactions on cybernetics*, vol. 46, no. 1, pp. 219–232, 2016.

[67] R. Tanabe and A. S. Fukunaga, "Improving the search performance of shade using linear population size reduction," in *Evolutionary Computation (CEC), 2014 IEEE Congress on*. IEEE, 2014, pp. 1658–1665.

[68] Y. Wang, Z. Cai, and Q. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 55–66, 2011.

[69] J. Brest, V. Zumer, and M. Maucec, "Self-adaptive differential evolution algorithm in constrained real-parameter optimization," in *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*. IEEE, pp. 215–222.

[70] H. A. Abbass, "The self-adaptive pareto differential evolution algorithm," in *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on*, vol. 1. IEEE, 2002, pp. 831–836.

[71] M. G. Omran, A. Salman, and A. P. Engelbrecht, "Self-adaptive differential evolution," in *International Conference on Computational and Information Science*. Springer, 2005, pp. 192–199.

[72] S. Elsayed and R. Sarker, "Evolving the parameters of differential evolution using evolutionary algorithms," in *Proceedings of the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems, Volume 1*. Springer, 2015, pp. 523–534.

[73] J. Teo, "Exploring dynamic self-adaptive populations in differential evolution," *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, vol. 10, no. 8, pp. 673–686, 2006.

[74] V. Tirronen and F. Neri, "Differential evolution with fitness diversity self-adaptation," *Nature-inspired algorithms for optimisation*, pp. 199–234, 2009.

[75] R. Mallipeddi, P. N. Suganthan, Q. K. Pan, and M. F. Tasgetiren, "Differential evolution algorithm with ensemble of parameters and mutation strategies," *Applied Soft Computing*, vol. 11, pp. 1679–1696, 2011.

[76] M. Z. Ali, N. H. Awad, P. N. Suganthan, and R. G. Reynolds, "An adaptive multipopulation differential evolution with dynamic population reduction," *IEEE transactions on cybernetics*, 2016.

[77] J. Rönkkönen *et al.*, *ContinuousMultimodal Global Optimization with Differential Evolution-Based Methods*. Lappeenranta University of Technology, 2009.

[78] S. Ronald, "Robust encodings in genetic algorithms: A survey of encoding issues," in *Evolutionary Computation, 1997., IEEE International Conference on*. IEEE, 1997, pp. 43–48.

[79] Z. Michalewicz, C. Z. Janikow, and J. B. Krawczyk, “A modified genetic algorithm for optimal control problems,” *Computers & Mathematics with Applications*, vol. 23, no. 12, pp. 83–94, 1992.

[80] T. Nomura, “An analysis on linear crossover for real number chromosomes in an infinite population size,” in *Evolutionary Computation, 1997., IEEE International Conference on.* IEEE, 1997, pp. 111–114.

[81] N. J. Radcliffe, “Equivalence class analysis of genetic algorithms,” *Complex systems*, vol. 5, no. 2, pp. 183–205, 1991.

[82] J. Eshelman Larry and J. Schaffer David, “Real-coded genetic algorithms and interval-schemata,” *Foundations of Genetic Algorithms*, vol. 2.

[83] R. B. Agrawal, K. Deb, and R. Agrawal, “Simulated binary crossover for continuous search space,” *Complex systems*, vol. 9, no. 2, pp. 115–148, 1995.

[84] I. Ono and S. Kobayashi, “A real coded genetic algorithm for function optimization using unimodal normal distributed crossover.” in *Proceedings of the seventh international conference on genetic algorithms.* Morgan Kaufmann, 1997, pp. 246–253.

[85] K. Deb, A. Anand, and D. Joshi, “A computationally efficient evolutionary algorithm for real-parameter optimization,” *Evolutionary computation*, vol. 10, no. 4, pp. 371–395, 2002.

[86] S. Tsutsui, M. Yamamura, and T. Higuchi, “Multi-parent recombination with simplex crossover in real coded genetic algorithms,” in *Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation-Volume 1.* Morgan Kaufmann Publishers Inc., 1999, pp. 657–664.

[87] E. Elfeky, R. Sarker, and D. Essam, "A simple ranking and selection for constrained evolutionary optimization," *Simulated Evolution and Learning*, pp. 537–544, 2006.

[88] S. Elsayed, R. Sarker, and D. Essam, "A comparative study of different variants of genetic algorithms for constrained optimization," in *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2010, vol. 6457, pp. 177–186. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-17298-4_18

[89] H. K. Singh, T. Ray, and W. Smith, "Performance of infeasibility empowered memetic algorithm for CEC 2010 constrained optimization problems," in *Evolutionary Computation (CEC), 2010 IEEE Congress on*. IEEE, 2010, pp. 1–8.

[90] N. M. I. M. Hamza, "Hybridizing constraint consensus methods with evolutionary algorithms for constrained optimization," Ph.D. dissertation, University of New South Wales, 2012.

[91] Z. Michalewicz and S. J. Hartley, "Genetic algorithms+ data structures= evolution programs," *Mathematical Intelligencer*, vol. 18, no. 3, p. 71, 1996.

[92] S. M. Elsayed, R. A. Sarker, and D. L. Essam, "On an evolutionary approach for constrained optimization problem solving," *Applied Soft Computing*, vol. 12, no. 10, pp. 3208–3227, 2012.

[93] D. V. Arnold, *Noisy optimization with evolution strategies*. Springer Science & Business Media, 2002, vol. 8.

[94] T. Asselmeyer, W. Ebeling, and H. Rosé, "Evolutionary strategies of optimization," *Physical Review E*, vol. 56, no. 1, p. 1171, 1997.

[95] T. Bäck and M. Schütz, "Evolution strategies for mixed-integer optimization of optical multilayer systems." in *Evolutionary Programming*, 1995, pp. 33–51.

[96] E. Mezura-Montes and C. A. C. Coello, "A simple multimembered evolution strategy to solve constrained optimization problems," *IEEE Transactions on Evolutionary computation*, vol. 9, no. 1, pp. 1–17, 2005.

[97] H.-G. Beyer and D. V. Arnold, "Theory of evolution strategies - A tutorial," in *Theoretical aspects of evolutionary computing.* Springer, 2001, pp. 109–133.

[98] H.-G. Beyer, *The theory of evolution strategies.* Springer Science & Business Media, 2013.

[99] N. Hansen, "Benchmarking a BI-population CMA-ES on the BBOB-2009 function testbed," in *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers.* ACM, 2009, pp. 2389–2396.

[100] A. Auger and N. Hansen, "A restart CMA evolution strategy with increasing population size," in *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, vol. 2. IEEE, 2005, pp. 1769–1776.

[101] G. A. Jastrebski and D. V. Arnold, "Improving evolution strategies through active covariance matrix adaptation," in *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on.* IEEE, 2006, pp. 2814–2821.

[102] E. K. Burke, M. Hyde, G. Kendall, G. Ochoa, E. Özcan, and J. R. Woodward, "A classification of hyper-heuristic approaches," in *Handbook of metaheuristics.* Springer, 2010, pp. 449–468.

[103] J. A. Vrugt, B. A. Robinson, and J. M. Hyman, "Self-adaptive multimethod search for global optimization in real-parameter spaces," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 243–259, 2009.

[104] S. M. Elsayed, R. A. Sarker, D. L. Essam, and N. M. Hamza, "Testing united multi-operator evolutionary algorithms on the CEC2014 real-parameter numerical optimization," in *Evolutionary Computation (CEC), 2014 IEEE Congress on*. IEEE, 2014, pp. 1650–1657.

[105] F. V. Nepomuceno and A. P. Engelbrecht, "A self-adaptive heterogeneous PSO for real-parameter optimization," in *Evolutionary Computation (CEC), 2013 IEEE Congress on*. IEEE, 2013, pp. 361–368.

[106] L.-A. Gordián-Rivera and E. Mezura-Montes, "A combination of specialized differential evolution variants for constrained optimization," in *Advances in Artificial Intelligence–IBERAMIA 2012*. Springer, 2012, pp. 261–270.

[107] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine learning*, vol. 47, no. 2-3, pp. 235–256, 2002.

[108] K. Li, A. Fialho, S. Kwong, and Q. Zhang, "Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 1, pp. 114–130, 2014.

[109] F. V. Nepomuceno and A. P. Engelbrecht, "A self-adaptive heterogeneous PSO inspired by ants," in *International Conference on Swarm Intelligence*. Springer, 2012, pp. 188–195.

[110] J. Maturana, A. Fialho, F. Saubion, M. Schoenauer, F. Lardeux, and M. Sebag, "Adaptive operator selection and management in evolutionary algorithms," in *Autonomous Search*. Springer, 2011, pp. 161–189.

[111] D. E. Goldberg, "Probability matching, the magnitude of reinforcement, and classifier system bidding," *Machine Learning*, vol. 5, no. 4, pp. 407–425, 1990.

[112] D. Thierens, "An adaptive pursuit strategy for allocating operator probabilities," in *Proceedings of the 7th annual conference on Genetic and evolutionary computation*. ACM, 2005, pp. 1539–1546.

[113] Á. Fialho, L. Da Costa, M. Schoenauer, and M. Sebag, "Analyzing bandit-based adaptive operator selection mechanisms," *Annals of Mathematics and Artificial Intelligence*, vol. 60, no. 1, pp. 25–64, 2010.

[114] Q. Fan and X. Yan, "Self-adaptive differential evolution algorithm with discrete mutation control parameters," *Expert Systems with Applications*, vol. 42, no. 3, pp. 1551–1572, 2015.

[115] N. R. Sabar, M. Ayob, G. Kendall, and R. Qu, "A dynamic multiarmed bandit-gene expression programming hyper-heuristic for combinatorial optimization problems," *IEEE Transactions on Cybernetics*, vol. 45, no. 2, pp. 217–228, 2015.

[116] R. A. Gonçalves, C. P. Almeida, and A. Pozo, "Upper confidence bound (UCB) algorithms for adaptive operator selection in MOEA/D," in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2015, pp. 411–425.

[117] Á. Fialho, M. Schoenauer, and M. Sebag, "Toward comparison-based adaptive operator selection," in *Proceedings of the 12th annual conference on Genetic and evolutionary computation.* ACM, 2010, pp. 767–774.

[118] Á. Fialho, "Adaptive operator selection for optimization," Ph.D. dissertation, Université Paris Sud-Paris XI, 2010.

[119] O. Mersmann, B. Bischl, H. Trautmann, M. Preuss, C. Weihs, and G. Rudolph, "Exploratory landscape analysis," in *Proceedings of the 13th annual conference on Genetic and evolutionary computation.* ACM, 2011, pp. 829–836.

[120] S. Poursoltan and F. Neumann, "Ruggedness quantifying for constrained continuous fitness landscapes," in *Evolutionary Constrained Optimization.* Springer, 2015, pp. 29–50.

[121] N. D. Jana, J. Sil, and S. Das, "Selection of appropriate metaheuristic algorithms for protein structure prediction in ab off-lattice model: a perspective from fitness landscape analysis," *Information Sciences*, vol. 391, pp. 28–64, 2017.

[122] M. Tomassini, L. Vanneschi, P. Collard, and M. Clergue, "A study of fitness distance correlation as a difficulty measure in genetic programming," *Evolutionary Computation*, vol. 13, no. 2, pp. 213–239, 2005.

[123] Y. Borenstein and R. Poli, "Information landscapes," in *Proceedings of the 7th annual conference on Genetic and evolutionary computation.* ACM, 2005, pp. 1515–1522.

[124] Y. Borenstein and R. Poli, "Decomposition of fitness functions in random heuristic search," in *Foundations of Genetic Algorithms.* Springer, 2007, pp. 123–137.

[125] R. Morgan and M. Gallagher, "Analysing and characterising optimization problems using length scale," *Soft Computing*, vol. 21, no. 7, pp. 1735–1752, 2017.

[126] K. C. Steer, A. Wirth, and S. K. Halgamuge, "Information theoretic classification of problems for metaheuristics," in *Simulated Evolution and Learning*. Springer, 2008, pp. 319–328.

[127] K. M. Malan and A. P. Engelbrecht, "Quantifying ruggedness of continuous landscapes using entropy," in *Proceedings of the Eleventh conference on Congress on Evolutionary Computation*. IEEE Press, 2009, pp. 1440–1447.

[128] E. Mezura-Montes and C. A. C. Coello, "Constraint-handling in nature-inspired numerical optimization: past, present and future," *Swarm and Evolutionary Computation*, vol. 1, no. 4, pp. 173–194, 2011.

[129] R. Datta and K. Deb, *Evolutionary constrained optimization*. Springer, 2015.

[130] B. Tessema and G. G. Yen, "An adaptive penalty formulation for constrained evolutionary optimization," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 39, no. 3, pp. 565–578, 2009.

[131] C. A. C. Coello, "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*, vol. 191, no. 11, pp. 1245–1287, 2002.

[132] K. Deb, "An efficient constraint handling method for genetic algorithms," *Computer Methods in Applied Mechanics and Engineering*, vol. 186, pp.

311–338, 2000. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045782599003898

[133] A. H. Aguirre, S. B. Rionda, C. A. Coello Coello, G. L. Lizárraga, and E. M. Montes, "Handling constraints using multiobjective optimization concepts," *International Journal for Numerical Methods in Engineering*, vol. 59, no. 15, pp. 1989–2017, 2004.

[134] C. A. Coello Coello, "Constraint-handling using an evolutionary multiobjective optimization technique," *Civil Engineering Systems*, vol. 17, no. 4, pp. 319–346, 2000.

[135] T. Ray, K. Liew, and P. Saini, "An intelligent information sharing strategy within a swarm for unconstrained and constrained optimization problems," *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, vol. 6, no. 1, pp. 38–44, 2002.

[136] S. Venkatraman and G. G. Yen, "A generic framework for constrained optimization using genetic algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 4, pp. 424–435, 2005.

[137] Z. Cai and Y. Wang, "A multiobjective optimization-based evolutionary algorithm for constrained optimization," *IEEE Transactions on evolutionary computation*, vol. 10, no. 6, pp. 658–675, 2006.

[138] K. Deb, S. Lele, and R. Datta, "A hybrid evolutionary multi-objective and SQP based procedure for constrained optimization," in *International Symposium on Intelligence Computation and Applications.* Springer, 2007, pp. 36–45.

[139] Y. Wang, Z. Cai, Y. Zhou, and W. Zeng, "An adaptive tradeoff model for constrained evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 80–92, 2008.

[140] T. Takahama and S. Sakai, "Constrained optimization by the ε constrained differential evolution with an archive and gradient-based mutation," in *Evolutionary Computation (CEC), 2010 IEEE Congress on.* IEEE, 2010, pp. 1–9.

[141] T. Takahama and S. Sakai, "Solving difficult constrained optimization problems by the ε constrained differential evolution with gradient-based mutation," in *Constraint-Handling in Evolutionary Optimization.* Springer, 2009, pp. 51–72.

[142] L. Wang and L.-p. Li, "An effective differential evolution with level comparison for constrained engineering design," *Structural and Multidisciplinary Optimization*, vol. 41, no. 6, pp. 947–963, 2010.

[143] W. Yi, X. Li, L. Gao, Y. Zhou, and J. Huang, "ε constrained differential evolution with pre-estimated comparison using gradient-based approximation for constrained optimization problems," *Expert Systems With Applications*, vol. 44, pp. 37–49, 2016.

[144] T. P. Runarsson and X. Yao, "Stochastic ranking for constrained evolutionary optimization," *IEEE Transactions on evolutionary computation*, vol. 4, no. 3, pp. 284–294, 2000.

[145] M. Schoenauer and Z. Michalewicz, "Boundary operators for constrained parameter optimization problems." in *ICGA*, 1997, pp. 322–329.

[146] R. Mallipeddi and P. N. Suganthan, “Ensemble of constraint handling techniques,” *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 4, pp. 561–579, 2010.

[147] S. M. Elsayed, R. A. Sarker, and D. L. Essam, “A self-adaptive combined strategies algorithm for constrained optimization using differential evolution,” *Applied Mathematics and Computation*, vol. 241, pp. 267–282, 2014.

[148] R. Tanabe and A. Fukunaga, “Evaluating the performance of SHADE on CEC 2013 benchmark problems,” in *Evolutionary Computation (CEC), 2013 IEEE Congress on*. IEEE, 2013, pp. 1952–1959.

[149] J. Brest, M. S. Maučec, and B. Bošković, “iL-SHADE: Improved L-SHADE algorithm for single objective real-parameter optimization,” in *Evolutionary Computation (CEC), 2016 IEEE Congress on*. IEEE, 2016, pp. 1188–1195.

[150] S.-M. Guo, J. S.-H. Tsai, C.-C. Yang, and P.-H. Hsu, “A self-optimization approach for L-SHADE incorporated with eigenvector-based crossover and successful-parent-selecting framework on CEC 2015 benchmark set,” in *Evolutionary Computation (CEC), 2015 IEEE Congress on*. IEEE, 2015, pp. 1003–1010.

[151] N. Awad, M. Z. Ali, and R. G. Reynolds, “A differential evolution algorithm with success-based parameter adaptation for CEC2015 learning-based optimization,” in *Evolutionary Computation (CEC), 2015 IEEE Congress on*. IEEE, 2015, pp. 1098–1105.

[152] N. H. Awad, M. Z. Ali, P. N. Suganthan, and R. G. Reynolds, “An ensemble sinusoidal parameter adaptation incorporated with L-SHADE for solving CEC2014 benchmark problems,” in *Evolutionary Computation (CEC), 2016 IEEE Congress on*. IEEE, 2016, pp. 2958–2965.

[153] R. Poláková, J. Tvrdík, and P. Bujok, "Evaluating the performance of L-Shade with competing strategies on CEC2014 single parameter-operator test suite," in *Evolutionary Computation (CEC), 2016 IEEE Congress on.* IEEE, 2016, pp. 1181–1187.

[154] A. Zamuda and J. Brest, "Population reduction differential evolution with multiple mutation strategies in real world industry challenges," in *Swarm and Evolutionary Computation.* Springer, 2012, pp. 154–161.

[155] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, 2009.

[156] R. Mallipeddi, P. N. Suganthan, Q.-K. Pan, and M. F. Tasgetiren, "Differential evolution algorithm with ensemble of parameters and mutation strategies," *Applied Soft Computing*, vol. 11, no. 2, pp. 1679–1696, 2011.

[157] J. A. Vrugt and B. A. Robinson, "Improved evolutionary optimization from genetically adaptive multimethod search," *Proceedings of the National Academy of Sciences*, vol. 104, no. 3, pp. 708–711, 2007.

[158] F. Peng, K. Tang, G. Chen, and X. Yao, "Population-based algorithm portfolios for numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 5, pp. 782–800, 2010.

[159] O. Olorunda and A. P. Engelbrecht, "An analysis of heterogeneous cooperative algorithms," in *Evolutionary Computation, 2009. CEC'09. IEEE Congress on.* IEEE, 2009, pp. 1562–1569.

[160] J. Grobler, A. P. Engelbrecht, G. Kendall, and V. Yadavalli, "Heuristic space diversity control for improved meta-hyper-heuristic performance," *Information Sciences*, vol. 300, pp. 49–62, 2015.

[161] G. Li, Q. Lin, L. Cui, Z. Du, Z. Liang, J. Chen, N. Lu, and Z. Ming, “A novel hybrid differential evolution algorithm with modified CoDE and JADE,” *Applied Soft Computing*, vol. 47, pp. 577–599, 2016.

[162] S.-Z. Zhao, P. N. Suganthan, and S. Das, “Self-adaptive differential evolution with multi-trajectory search for large-scale optimization,” *Soft Computing*, vol. 15, no. 11, pp. 2175–2185, 2011.

[163] S. Elsayed, N. Hamza, and R. Sarker, “Testing united multi-operator evolutionary algorithms-II on single objective optimization problems,” in *Evolutionary Computation (CEC), 2016 IEEE Congress on.* IEEE, 2016, pp. 2966–2973.

[164] P. Merz and B. Freisleben, “Fitness landscape analysis and memetic algorithms for the quadratic assignment problem,” *IEEE transactions on evolutionary computation*, vol. 4, no. 4, pp. 337–352, 2000.

[165] B. Bischl, O. Mersmann, H. Trautmann, and M. Preuß, “Algorithm selection based on exploratory landscape analysis and cost-sensitive learning,” in *Proceedings of the 14th annual conference on Genetic and evolutionary computation.* ACM, 2012, pp. 313–320.

[166] N. Hansen, S. Finck, R. Ros, and A. Auger, “Real-parameter black-box optimization benchmarking 2009: Noiseless functions definitions,” Ph.D. dissertation, INRIA, 2009.

[167] N. Hansen, A. Auger, S. Finck, and R. Ros, “Real-parameter black-box optimization benchmarking 2010: Experimental setup,” Ph.D. dissertation, INRIA, 2010.

[168] R. W. Garden and A. P. Engelbrecht, "Analysis and classification of optimisation benchmark functions and benchmark suites," in *IEEE Congress on Evolutionary Computation (CEC), 2014*. IEEE, 2014, pp. 1641–1649.

[169] K. M. Malan and A. P. Engelbrecht, "Particle swarm optimisation failure prediction based on fitness landscape characteristics," in *Swarm Intelligence (SIS), 2014 IEEE Symposium on*. IEEE, 2014, pp. 1–9.

[170] M. A. Muñoz, M. Kirley, and S. K. Halgamuge, "A meta-learning prediction model of algorithm performance for continuous optimization problems," in *Parallel Problem Solving from Nature-PPSN XII*. Springer, 2012, pp. 226–235.

[171] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y.-P. Chen, A. Auger, and S. Tiwari, "Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization," *KanGAL report*, vol. 2005005, 2005.

[172] M. A. Muñoz, M. Kirley, and S. K. Halgamuge, "Exploratory landscape analysis of continuous space optimization problems using information content," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 1, pp. 74–87, 2015.

[173] Y. Sun, M. Kirley, and S. K. Halgamuge, "Quantifying variable interactions in continuous optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 2, pp. 249–264, 2017.

[174] F. Caraffini, F. Neri, and L. Picinali, "An analysis on separability for memetic computing automatic design," *Information Sciences*, vol. 265, pp. 1–22, 2014.

[175] N. D. Jana, J. Sil, and S. Das, "Continuous fitness landscape analysis using a chaos-based random walk algorithm," *Soft Computing*, pp. 1–28, 2016.

[176] T. Takahama and S. Sakai, "Constrained optimization by the ε constrained differential evolution with gradient-based mutation and feasible elites," in *IEEE congress on evolutionary computation.* IEEE, 2006, pp. 1–8.

[177] Y. Wang and Z. Cai, "Combining multiobjective optimization with differential evolution to solve constrained optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 1, pp. 117–134, 2012.

[178] N. Dong and Y. Wang, "An unbiased bi-objective optimization model and algorithm for constrained optimization," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 28, no. 08, p. 1459008, 2014.

[179] W. Gong, Z. Cai, and D. Liang, "Adaptive ranking mutation operator based differential evolution for constrained optimization," *IEEE transactions on cybernetics*, vol. 45, no. 4, pp. 716–727, 2015.

[180] S. M. Elsayed, R. A. Sarker, and D. L. Essam, "GA with a new multi-parent crossover for solving IEEE-CEC2011 competition problems," in *Evolutionary Computation (CEC), 2011 IEEE Congress on.* IEEE, 2011, pp. 1034–1040.

[181] S. Das and P. N. Suganthan, "Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems," *Jadavpur University, Nanyang Technological University, Kolkata*, 2010.

[182] S. M. Elsayed, R. A. Sarker, and D. L. Essam, "A new genetic algorithm for solving optimization problems," *Engineering Applications of Artificial Intelligence*, vol. 27, pp. 57–69, 2014.

[183] R. Mallipeddi and P. N. Suganthan, "Differential evolution with ensemble of constraint handling techniques for solving CEC 2010 benchmark problems," in *Evolutionary Computation (CEC), 2010 IEEE Congress on*. IEEE, 2010, pp. 1–8.

[184] E. K. da Silva, H. J. Barbosa, and A. C. Lemonge, "An adaptive constraint handling technique for differential evolution with dynamic use of variants in engineering optimization," *Optimization and Engineering*, vol. 12, no. 1-2, pp. 31–54, 2011.

[185] S. M. Elsayed, R. Sarker, and D. L. Essam, "An improved self-adaptive differential evolution algorithm for optimization problems," *Industrial Informatics, IEEE Transactions on*, vol. 9, no. 1, pp. 89–99, 2013.

[186] R. Mallipeddi and P. N. Suganthan, "Problem definitions and evaluation criteria for the CEC 2010 competition on constrained real-parameter optimization," *Nanyang Technological University, Singapore*, 2010.

[187] Y. Wang, B.-C. Wang, H.-X. Li, and G. G. Yen, "Incorporating objective function information into the feasibility rule for constrained evolutionary optimization," 2015.

[188] S. Elsayed, R. Sarker, C. C. Coello, and T. Ray, "Adaptation of operators and continuous control parameters in differential evolution for constrained optimization," *Soft Computing*, pp. 1–22, 2017.

[189] S. M. Elsayed, R. A. Sarker, and D. L. Essam, "Adaptive configuration of evolutionary algorithms for constrained optimization," *Applied Mathematics and Computation*, vol. 222, pp. 680–711, 2013.

[190] F. Zaman, S. M. Elsayed, T. Ray, and R. A. Sarker, "Configuring two-algorithm-based evolutionary approach for solving dynamic economic dispatch problems," *Engineering Applications of Artificial Intelligence*, vol. 53, pp. 105–125, 2016.

[191] M.-H. Tayarani-N and A. Prügel-Bennett, "On the landscape of combinatorial optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 420–434, 2014.

[192] M.-H. Tayarani-N and A. Prügel-Bennett, "Quadratic assignment problem: a landscape analysis," *Evolutionary Intelligence*, vol. 8, no. 4, pp. 165–184, 2015.

[193] P. A. Consoli, L. L. Minku, and X. Yao, "Dynamic selection of evolutionary algorithm operators based on online learning and fitness landscape metrics," in *Simulated Evolution and Learning.* Springer, 2014, pp. 359–370.

[194] F. Neumann and S. Poursoltan, "Feature-based algorithm selection for constrained continuous optimisation," in *Evolutionary Computation (CEC), 2016 IEEE Congress on.* IEEE, 2016, pp. 1461–1468.

[195] M. Schoenauer and Z. Michalewicz, "Evolutionary computation at the edge of feasibility," in *International Conference on Parallel Problem Solving from Nature.* Springer, 1996, pp. 245–254.

[196] Z. Y. Wu and A. R. Simpson, "A self-adaptive boundary search genetic algorithm and its application to water distribution systems," *Journal of Hydraulic Research*, vol. 40, no. 2, pp. 191–203, 2002.

[197] G. Wu, W. Pedrycz, P. Suganthan, and R. Mallipeddi, "A variable reduction strategy for evolutionary algorithms handling equality constraints," *Applied Soft Computing*, vol. 37, pp. 774–786, 2015.

[198] J. Liang, B. Qu, and P. Suganthan, "Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization," *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore*, 2013.

[199] J. Liang, B. Qu, P. Suganthan, and Q. Chen, "Problem definitions and evaluation criteria for the CEC 2015 competition on Learning-based Real-Parameter Single Objective Optimization," *Computational Intelligence Laboratory*, 2014.

[200] K. Q. Ye, "Orthogonal column latin hypercubes and their application in computer experiments," *Journal of the American Statistical Association*, vol. 93, no. 444, pp. 1430–1439, 1998.

[201] Z. Zhao, J. Yang, Z. Hu, and H. Che, "A differential evolution algorithm with self-adaptive strategy and control parameters based on symmetric latin hypercube design for unconstrained optimization problems," *European Journal of Operational Research*, vol. 250, no. 1, pp. 30–45, 2016.

[202] K. Malan and A. Engelbrecht, "Characterising the searchability of continuous optimisation problems for PSO," *Swarm Intelligence*, vol. 8, no. 4, pp. 275–302, 2014.

[203] R. Woolson, "Wilcoxon signed-rank test," *Wiley Encyclopedia of Clinical Trials*, 2008.

[204] J. d. Kroon and P. Laan, "A generalization of friedman's rank statistic," *Statistica Neerlandica*, vol. 37, no. 1, pp. 1–14, 1983.

[205] H. J. Barbosa, H. S. Bernardino, and A. M. Barreto, "Using performance profiles for the analysis and design of benchmark experiments," in *Advances in Metaheuristics*. Springer, 2013, pp. 21–36.

[206] E. D. Dolan and J. J. Moré, "Benchmarking optimization software with performance profiles," *Mathematical programming*, vol. 91, no. 2, pp. 201–213, 2002.

[207] J. Liang, T. P. Runarsson, E. Mezura-Montes, M. Clerc, P. Suganthan, C. C. Coello, and K. Deb, "Problem definitions and evaluation criteria for the CEC 2006 special session on constrained real-parameter optimization," *Journal of Applied Mechanics*, vol. 41, p. 8, 2006.

[208] S. Elsayed, R. Sarker, and C. C. Coello, "Enhanced multi-operator differential evolution for constrained optimization," in *Evolutionary Computation (CEC), 2016 IEEE Congress on*. IEEE, 2016, pp. 4191–4198.

[209] E. Mezura-Montes and C. A. C. Coello, "Adding a diversity mechanism to a simple evolution strategy to solve constrained optimization problems," in *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on*, vol. 1. IEEE, 2003, pp. 6–13.

[210] C. Si, J. An, T. Lan, T. Ußmüller, L. Wang, and Q. Wu, "On the equality constraints tolerance of constrained optimization problems," *Theoretical Computer Science*, vol. 551, pp. 55–65, 2014.

[211] M. Asafuddoula, T. Ray, and R. Sarker, "An adaptive hybrid differential evolution algorithm for single objective optimization," *Applied Mathematics and Computation*, vol. 231, pp. 601–618, 2014.

[212] W. Gong, Z. Cai, and D. Liang, "Engineering optimization by means of an improved constrained differential evolution," *Computer Methods in Applied Mechanics and Engineering*, vol. 268, pp. 884–904, 2014.

[213] W. Zhang, G. G. Yen, and Z. He, "Constrained optimization via artificial immune system," *IEEE transactions on cybernetics*, vol. 44, no. 2, pp. 185–198, 2014.

[214] T. Takahama and S. Sakai, "Constrained optimization by the ε constrained differential evolution with an archive and gradient-based mutation," in *IEEE congress on evolutionary computation*. IEEE, 2010, pp. 1–9.

[215] N. M. Hamza, R. A. Sarker, and D. L. Essam, "Differential evolution with a mix of constraint consenus methods for solving a real-world optimization problem," in *Evolutionary Computation (CEC), 2012 IEEE Congress on*. IEEE, 2012, pp. 1–7.

[216] J. J. Liang, S. Zhigang, and L. Zhihui, "Coevolutionary comprehensive learning particle swarm optimizer," in *Evolutionary Computation (CEC), 2010 IEEE Congress on*. IEEE, 2010, pp. 1–8.

[217] E. Mezura-Montes and R. E. Velez-Koeppel, "Elitist artificial bee colony for constrained real-parameter optimization," in *Evolutionary Computation (CEC), 2010 IEEE Congress on*. IEEE, 2010, pp. 1–8.

[218] N. M. Hamza, D. L. Essam, and R. A. Sarker, "Constraint consensus mutation-based differential evolution for constrained optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 3, pp. 447–459, 2016.

[219] P. Korošec and J. Šilc, "The continuous differential ant-stigmergy algorithm applied to real-world optimization problems," in *Evolutionary Computation (CEC), 2011 IEEE Congress on*. IEEE, 2011, pp. 1327–1334.

[220] M. Asafuddoula, T. Ray, and R. Sarker, "An adaptive differential evolution algorithm and its performance on real world optimization problems," in *2011 IEEE Congress of Evolutionary Computation (CEC)*. IEEE, 2011, pp. 1057–1062.

[221] R. Mallipeddi and P. N. Suganthan, "Ensemble differential evolution algorithm for CEC2011 problems," in *Evolutionary Computation (CEC), 2011 IEEE Congress on*. IEEE, 2011, pp. 1557–1564.

[222] A. Mandal, A. K. Das, P. Mukherjee, S. Das, and P. N. Suganthan, "Modified differential evolution with local search algorithm for real world optimization," in *Evolutionary Computation (CEC), 2011 IEEE Congress on*. IEEE, 2011, pp. 1565–1572.

[223] G. Reynoso-Meza, J. Sanchis, X. Blasco, and J. M. Herrero, "Hybrid DE algorithm with adaptive crossover operator for solving real-world numerical optimization problems," in *Evolutionary Computation (CEC), 2011 IEEE Congress on*. IEEE, 2011, pp. 1551–1556.

[224] K. Deb, "An efficient constraint handling method for genetic algorithms," *Computer methods in applied mechanics and engineering*, vol. 186, no. 2, pp. 311–338, 2000.

[225] Y. Wang, Z. Cai, and Q. Zhang, "Enhancing the search ability of differential evolution through orthogonal crossover," *Information Sciences*, vol. 185, no. 1, pp. 153–177, 2012.

[226] S. M. Elsayed, R. A. Sarker, and D. L. Essam, "United multi-operator evolutionary algorithms," in *Evolutionary Computation (CEC), 2014 IEEE Congress on*. IEEE, 2014, pp. 1006–1013.

[227] N. Hansen, S. D. Müller, and P. Koumoutsakos, "Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES)," *Evolutionary computation*, vol. 11, no. 1, pp. 1–18, 2003.

[228] H. K. Singh and T. Ray, "Performance of a hybrid EA-DE-memetic algorithm on CEC 2011 real world optimization problems," in *Evolutionary Computation (CEC), 2011 IEEE Congress on*. IEEE, 2011, pp. 1322–1326.

Appendices

In this section, detailed results from all the proposed algorithms are presented in four appendices, each containing different tables.

1. Appendix A: computational results from proposed landscape-based algorithm for solving unconstrained optimization problems.
2. Appendix B: computational results from proposed landscape-based algorithm for solving constrained optimization problems.
3. Appendix C: computational results from proposed reduced search space mechanism for solving constrained optimization problems.
4. Appendix D: computational results from proposed multi-EAs framework for solving constrained optimization problems.
5. Appendix E: published paper entitled "Two-phase Differential Evolution Framework for Solving Optimization Problems".

Appendix A

Table A.1: Error values obtained by LSAOS-DE for 10D, 30D, and 50D

Prob.	10D			30D			50D		
	Best	Mean	Std.	Best	Mean	Std.	Best	Mean	Std.
F01	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	6.560E+01	2.062E+03	2.499E+03
F02	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	7.021E-08	2.053E-07
F03	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
F04	0.000E+00	1.654E+01	1.739E+01	0.000E+00	7.186E-05	3.060E-04	0.000E+00	1.924E+01	3.934E+01
F05	1.123E+00	1.376E+01	7.896E+00	2.000E+01	2.025E+01	8.289E-02	2.000E+01	2.000E+01	2.302E-03
F06	0.000E+00	0.000E+00	0.000E+00	0.000E+00	4.491E-02	1.648E-01	9.308E-02	1.198E+00	7.815E-01
F07	0.000E+00	1.014E-03	3.311E-03	0.000E+00	0.000E+00	0.000E+00	0.000E+00	3.093E-03	4.645E-03
F08	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
F09	0.000E+00	8.418E-01	1.294E+00	1.285E+00	4.686E+00	1.618E+00	9.950E+00	1.632E+01	2.962E+00
F10	0.000E+00	2.449E-03	1.749E-02	2.070E-08	9.175E-03	1.590E-02	1.222E-01	1.730E+00	1.486E+00
F11	1.458E+00	5.147E+01	4.879E+01	1.028E+03	1.603E+03	4.422E+02	2.654E+03	3.992E+03	7.883E+02
F12	5.808E-03	1.493E-01	9.303E-02	4.091E-02	3.020E-01	1.229E-01	3.558E-02	6.974E-02	2.275E-02
F13	1.550E-02	5.062E-02	1.337E-02	8.075E-02	1.215E-01	1.754E-02	1.423E-01	1.942E-01	2.041E-02
F14	2.559E-02	1.055E-01	3.073E-02	1.677E-01	2.180E-01	2.384E-02	2.478E-01	2.915E-01	1.881E-02
F15	2.537E-01	4.033E-01	6.565E-02	1.954E+00	2.489E+00	2.390E-01	3.173E+00	4.674E+00	8.364E-01
F16	2.857E-01	1.265E+00	3.423E-01	8.141E+00	9.244E+00	5.547E-01	1.673E+01	1.815E+01	9.651E-01
F17	0.000E+00	1.044E+00	1.010E+00	4.414E+01	1.358E+02	7.155E+01	4.355E+02	1.044E+03	3.560E+02
F18	2.015E-03	1.298E-01	1.431E-01	2.206E+00	6.345E+00	2.562E+00	6.496E+01	8.604E+01	1.057E+01
F19	2.147E-02	7.844E-02	5.921E-02	1.888E+00	3.251E+00	5.955E-01	7.720E+00	1.059E+01	8.509E-01
F20	6.715E-03	1.751E-01	1.673E-01	1.439E+00	3.856E+00	1.294E+00	5.075E+00	1.126E+01	3.977E+00
F21	1.227E-05	2.879E-01	2.435E-01	1.606E+00	3.223E+01	4.474E+01	1.338E+02	4.149E+02	1.200E+02
F22	4.203E-02	1.220E-01	4.574E-02	2.320E+01	3.361E+01	1.711E+01	2.231E+01	1.497E+02	8.477E+01

F23	3.295E+02	3.295E+02	2.870E-13	2.000E+02	3.062E+02	3.129E+01	2.000E+02	3.408E+02	2.031E+01
F24	1.000E+02	1.080E+02	2.603E+00	2.000E+02	2.137E+02	1.160E+01	2.000E+02	2.522E+02	3.134E+01
F25	1.000E+02	1.099E+02	6.571E+00	2.000E+02	2.025E+02	6.392E-01	2.000E+02	2.037E+02	2.625E+00
F26	1.000E+02	1.000E+02	1.662E-02	1.001E+02	1.001E+02	1.662E-02	1.001E+02	1.002E+02	1.735E-02
F27	8.109E-01	1.524E+00	3.012E-01	3.000E+02	3.000E+02	2.671E-02	3.005E+02	3.243E+02	2.351E+01
F28	3.568E+02	3.655E+02	1.625E+01	2.000E+02	8.182E+02	9.196E+01	2.000E+02	1.107E+03	1.337E+02
F29	2.215E+02	2.218E+02	2.015E-01	1.607E+02	6.854E+02	1.272E+02	7.034E+02	7.914E+02	4.866E+01
F30	4.543E+02	4.622E+02	1.634E+00	3.678E+02	4.872E+02	1.122E+02	8.024E+03	8.569E+03	4.186E+02
F31	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	3.112E+01	8.611E+02	1.285E+03
F32	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
F33	2.449E+00	1.778E+01	5.713E+00	2.000E+01	2.021E+01	7.421E-02	2.000E+01	2.000E+01	6.025E-04
F34	0.000E+00	1.117E+00	1.470E+00	2.009E+00	4.700E+00	1.408E+00	1.293E+01	1.882E+01	3.427E+00
F35	9.449E-01	3.268E+01	5.525E+01	9.220E+02	1.431E+03	2.458E+02	2.108E+03	3.432E+03	6.236E+02
F36	0.000E+00	1.228E+00	1.715E+00	2.856E+01	1.331E+02	6.555E+01	6.114E+02	1.241E+03	2.844E+02
F37	1.315E-02	1.142E-01	9.670E-02	4.900E+00	6.431E+00	5.770E-01	3.901E+01	4.099E+01	2.778E+00
F38	6.272E-06	2.346E-01	2.499E-01	3.138E+00	2.898E+01	1.394E+01	1.365E+01	1.419E+02	8.285E+01
F39	1.001E+02	1.002E+02	2.480E-02	1.024E+02	1.027E+02	1.047E-01	1.038E+02	1.040E+02	1.617E-01
F40	2.165E+02	2.165E+02	1.123E-13	1.604E+02	4.516E+02	1.231E+02	7.041E+02	8.846E+02	9.646E+01
F41	7.256E-01	5.472E+01	1.147E+02	4.000E+02	4.008E+02	5.965E+00	4.009E+02	4.425E+02	2.152E+01
F42	1.003E+02	1.008E+02	2.132E-01	1.029E+02	1.033E+02	1.995E-01	1.046E+02	1.052E+02	2.803E-01
F43	3.042E-02	3.048E-02	5.268E-05	2.558E-02	2.638E-02	6.544E-04	7.383E-02	8.357E-02	4.327E-03
F44	2.934E+03	2.934E+03	1.434E-07	3.112E+04	3.163E+04	6.911E+02	4.952E+04	5.988E+04	5.390E+03
F45	1.000E+02	1.000E+02	0.000E+00	1.000E+02	1.000E+02	1.235E-13	1.000E+02	1.000E+02	1.649E-13

Table A.2: Error values achieved by the proposed LSAOS-DE, LSHADE, SHADE, JADE, CoDE and SaDE for $10D$

problem	LSHADE	SHADE	JADE	CoDE	SaDE	LSAOSDE
	Mean(Std.)	**Mean(Std.)**	**Mean(Std.)**	**Mean(Std.)**	**Mean(Std.)**	**Mean(Std.)**
F01	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)=	0.00E+00(0.00E+00)
F02	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)
F03	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)
F04	3.21E+01(9.16E+00)	3.08E+01(1.11E+01)	3.21E+01(9.44E+00)	8.95E+00(1.46E+01)	1.95E+01(1.70E+01)	1.65E+01(1.74E+01)
F05	1.78E+01(5.64E+00)	1.82E+01(4.67E+00)	1.81E+01(3.70E+00)	1.88E+01(4.76E+00)	1.88E+01(4.78E+00)	1.38E+01(7.90E+00)
F06	0.00E+00(0.00E+00)	1.24E-06(8.84E-06)	0.00E+00(0.00E+00)	2.18E-10(1.55E-09)	2.19E-05(1.15E-04)	0.00E+00(0.00E+00)
F07	1.70E-03(5.27E-03)	4.02E-03(5.75E-03)	8.78E-03(7.77E-03)	3.70E-02(2.68E-02)	1.00E-03(3.06E-03)	1.01E-03(3.31E-03)
F08	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)
F09	2.54E+00(9.18E-01)	3.04E+00(7.93E-01)	3.26E+00(8.81E-01)	4.00E+00(2.18E+00)	7.27E+00(1.73E+00)	8.42E-01(1.29E+00)
F10	2.45E-03(1.22E-02)	8.57E-03(2.17E-02)	1.10E-02(2.71E-02)	3.43E-02(3.81E-02)	1.22E-03(8.75E-03)	2.45E-03(1.75E-02)
F11	2.86E+01(3.21E+01)	5.68E+01(4.61E+01)	8.63E+01(4.22E+01)	9.08E+01(9.05E+01)	3.84E+02(1.17E+02)	5.15E+01(4.88E+01)
F12	6.50E-02(1.70E-02)	1.34E-01(2.84E-02)	3.23E-01(5.99E-02)	4.70E-02(5.54E-02)	4.29E-01(9.11E-02)	1.49E-01(9.30E-02)
F13	5.14E-02(1.24E-02)	7.70E-02(1.74E-02)	7.56E-02(1.77E-02)	8.25E-02(3.12E-02)	1.38E-01(2.55E-02)	5.06E-02(1.34E-02)
F14	7.06E-02(2.28E-02)	9.28E-02(2.63E-02)	9.22E-02(3.29E-02)	1.01E-01(3.17E-02)	1.87E-01(4.81E-02)	1.06E-01(3.07E-02)
F15	3.81E-01(8.30E-02)	4.87E-01(9.05E-02)	5.50E-01(8.09E-02)	6.22E-01(1.69E-01)	1.23E+00(1.67E-01)	4.03E-01(6.57E-02)
F16	1.26E+00(2.51E-01)	1.64E+00(2.23E-01)	1.55E+00(3.67E-01)	1.20E+00(6.33E-01)	2.25E+00(2.39E-01)	1.26E+00(3.42E-01)
F17	7.92E-01(7.82E-01)	3.49E+00(1.68E+01)	1.39E+01(4.12E+01)	2.76E+00(4.42E+00)	1.11E+01(6.42E+00)	1.04E+00(1.01E+00)
F18	2.35E-01(1.80E-01)	2.00E-01(1.81E-01)	1.98E-01(2.22E-01)	3.67E-01(6.13E-01)	3.73E-01(4.76E-01)	1.30E-01(1.43E-01)
F19	1.11E-01(1.17E-01)	2.07E-01(1.56E-01)	2.35E-01(2.61E-01)	8.59E-02(7.55E-02)	2.20E-01(7.54E-02)	7.84E-02(5.92E-02)
F20	1.47E-01(1.24E-01)	2.89E-01(1.45E-01)	2.40E-01(1.67E-01)	2.42E-02(3.21E-02)	2.49E-01(1.00E-01)	1.75E-01(1.67E-01)
F21	3.36E-01(2.60E-01)	3.91E-01(2.85E-01)	4.06E-01(2.89E-01)	2.35E-01(2.48E-01)	3.37E-01(2.33E-01)	2.88E-01(2.43E-01)
F22	9.18E-02(4.03E-02)	2.75E-01(1.45E-01)	1.67E-01(5.46E-02)	6.70E-02(6.53E-02)	3.36E-01(1.46E-01)	1.22E-01(4.57E-02)
F23	3.29E+02(2.87E-13)	3.29E+02(2.87E-13)	3.29E+02(2.87E-13)	3.29E+02(2.87E-13)	3.29E+02(2.87E-13)	3.29E+02(2.87E-13)
F24	1.08E+02(1.29E+00)	1.09E+02(2.22E+00)	1.08E+02(2.53E+00)	1.12E+02(3.22E+00)	1.09E+02(4.16E+00)	1.08E+02(2.60E+00)
F25	1.47E+02(4.40E+01)	1.53E+02(4.44E+01)	1.37E+02(4.06E+01)	1.41E+02(4.12E+01)	1.20E+02(9.21E+00)	1.10E+02(6.57E+00)
F26	1.00E+02(1.64E-02)	1.00E+02(1.32E-02)	1.00E+02(1.82E-02)	1.00E+02(2.16E-02)	1.00E+02(2.49E-02)	1.00E+02(1.66E-02)
F27	5.60E+01(1.20E+02)	1.15E+02(1.57E+02)	1.60E+02(1.86E+02)	2.74E+01(8.85E+01)	1.02E+01(5.57E+01)	1.52E+00(3.01E-01)
F28	3.83E+02(3.60E+01)	3.98E+02(4.69E+01)	4.06E+02(5.09E+01)	3.60E+02(1.41E+01)	3.64E+02(2.40E+01)	3.65E+02(1.63E+01)
F29	2.22E+02(5.82E-01)	2.22E+02(5.80E-01)	2.20E+02(1.27E+01)	2.20E+02(1.26E+01)	2.12E+02(2.71E+01)	2.22E+02(2.02E-01)
F30	4.68E+02(1.73E+01)	4.68E+02(1.30E+01)	4.77E+02(2.42E+01)	4.64E+02(7.66E+00)	4.85E+02(2.41E+01)	4.62E+02(1.63E+00)

F31	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)
F32	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)
F33	1.66E+01(6.98E+00)	1.79E+01(4.71E+00)	1.93E+01(2.63E+00)	1.92E+01(3.92E+00)	1.95E+01(3.04E+00)	1.78E+01(5.71E+00)
F34	2.50E+00(7.26E-01)	3.05E+00(9.40E-01)	3.56E+00(8.90E-01)	4.31E+00(1.85E+00)	6.74E+00(1.39E+00)	1.12E+00(1.47E+00)
F35	1.55E+01(2.39E+01)	6.82E+01(5.75E+01)	7.98E+01(5.11E+01)	7.58E+01(8.45E+01)	3.51E+02(8.37E+01)	3.27E+01(5.53E+01)
F36	7.95E-01(7.39E-01)	3.80E+00(1.68E+01)	6.75E+00(2.38E+01)	6.11E-01(1.59E+00)	2.94E+00(4.07E+00)	1.23E+00(1.71E+00)
F37	1.21E-01(1.37E-01)	2.11E-01(9.70E-02)	3.50E-01(1.43E-01)	7.36E-02(5.80E-02)	3.51E-01(1.54E-01)	1.14E-01(9.67E-02)
F38	3.17E-01(2.43E-01)	3.10E-01(2.16E-01)	5.05E-01(6.38E-01)	1.26E-01(2.11E-01)	4.17E-01(2.08E-01)	2.35E-01(2.50E-01)
F39	1.00E+02(7.18E-14)	1.00E+02(2.09E-02)	1.00E+02(2.42E-02)	1.00E+02(4.91E-02)	1.00E+02(3.36E-02)	1.00E+02(2.48E-02)
F40	1.43E+02(3.51E+00)	2.18E+02(6.63E+00)	2.27E+02(3.29E+01)	2.17E+02(8.54E-01)	2.17E+02(1.10E+00)	2.17E+02(1.12E-13)
F41	7.96E+00(4.19E+01)	1.19E+02(1.47E+02)	1.43E+02(1.50E+02)	2.36E+02(1.24E+02)	6.71E+01(1.23E+02)	5.47E+01(1.15E+02)
F42	1.11E+02(4.95E-01)	1.01E+02(1.99E-01)	1.01E+02(2.07E-01)	1.02E+02(3.52E-01)	1.02E+02(3.39E-01)	1.01E+02(2.13E-01)
F43	9.41E-02(3.32E-03)	3.05E-02(8.21E-05)	3.05E-02(8.79E-05)	3.05E-02(5.78E-05)	3.06E-02(8.70E-05)	3.05E-02(5.27E-05)
F44	6.68E+03(2.72E+00)	4.20E+03(1.89E+03)	4.59E+03(2.07E+03)	3.30E+03(1.13E+03)	2.47E+03(1.11E+03)	2.93E+03(1.43E-07)
F45	1.00E+02(0.00E+00)	1.00E+02(0.00E+00)	1.00E+02(4.55E-14)	1.00E+02(0.00E+00)	1.00E+02(0.00E+00)	1.00E+02(0.00E+00)

Table A.3: Error values achieved by the proposed LSAOS-DE, EPSDE, MPEDE, UMOEAs and AMALGAM-SO for $10D$

problem	EPSDE	MPEDE	UMOEAs	AMALGAM-SO	LSAOSDE
	Mean(Std.)	**Mean(Std.)**	**Mean(Std.)**	**Mean(Std.)**	**Mean(Std.)**
F01	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	4.24E-05(1.41E-05)	0.00E+00(0.00E+00)
F02	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	7.75E-05(2.11E-05)	0.00E+00(0.00E+00)
F03	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	6.40E-05(4.10E-05)	0.00E+00(0.00E+00)
F04	0.00E+00(0.00E+00)	1.61E+01(1.71E+01)	1.42E+01(1.68E+01)	2.28E+01(1.58E+01)	1.65E+01(1.74E+01)
F05	2.01E+01(1.22E-02)	1.16E+01(9.88E+00)	2.00E+01(8.36E-02)	1.76E+01(6.53E+00)	1.38E+01(7.90E+00)
F06	3.01E+00(5.56E-01)	4.07E-04(1.92E-03)	1.60E-01(2.65E-01)	1.53E-02(8.55E-02)	0.00E+00(0.00E+00)
F07	1.49E-02(1.11E-02)	3.62E-03(6.35E-03)	1.45E-04(1.04E-03)	2.35E-04(6.28E-05)	1.01E-03(3.31E-03)
F08	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	6.00E-04(3.95E-03)	1.99E-03(3.79E-03)	0.00E+00(0.00E+00)
F09	3.86E+00(9.87E-01)	3.51E+00(1.22E+00)	3.73E+00(2.11E+00)	2.07E-01(4.04E-01)	8.42E-01(1.29E+00)
F10	4.90E-02(6.88E-02)	2.45E-03(1.22E-02)	1.95E+00(2.27E+00)	3.35E+00(4.29E+00)	2.45E-03(1.75E-02)
F11	3.40E+02(1.11E+02)	9.31E+01(8.68E+01)	3.80E+02(1.96E+02)	1.39E+01(1.30E+01)	5.15E+01(4.88E+01)
F12	3.14E-01(5.55E-02)	2.58E-01(8.37E-02)	3.79E-03(1.67E-02)	7.06E-02(4.88E-02)	1.49E-01(9.30E-02)
F13	1.21E-01(2.44E-02)	8.03E-02(2.66E-02)	1.85E-02(1.38E-02)	1.10E-02(4.83E-03)	5.06E-02(1.34E-02)
F14	1.32E-01(3.51E-02)	7.65E-02(2.38E-02)	1.69E-01(4.06E-02)	2.65E-01(5.97E-02)	1.06E-01(3.07E-02)
F15	7.11E-01(1.17E-01)	5.87E-01(1.65E-01)	8.50E-01(1.85E-01)	6.80E-01(1.14E-01)	4.03E-01(6.57E-02)
F16	2.46E+00(2.69E-01)	1.47E+00(3.55E-01)	2.77E+00(2.54E-01)	1.21E+00(4.63E-01)	1.26E+00(3.42E-01)
F17	4.93E+01(3.90E+01)	5.34E+00(5.38E+00)	3.12E+01(3.54E+01)	3.99E+01(4.08E+01)	1.04E+00(1.01E+00)
F18	1.21E+00(7.66E-01)	5.28E-01(4.36E-01)	1.04E+00(6.71E-01)	2.73E+00(4.02E+00)	1.30E-01(1.43E-01)
F19	1.42E+00(2.29E-01-)	1.40E-01(7.57E-02)	7.16E-01(4.69E-01)	5.94E-01(3.93E-01)	7.84E-02(5.92E-02)
F20	1.55E-01(7.26E-02)	1.73E-01(1.12E-01)	9.11E-01(3.90E-01)	1.43E+00(1.26E+00)	1.75E-01(1.67E-01)
F21	5.25E+00(2.00E+01)	4.93E-01(2.82E-01)	1.84E+00(3.04E+00)	9.50E+00(2.03E+01)	2.88E-01(2.43E-01)
F22	1.96E+01(2.23E+00)	2.76E-01(1.93E-01)	9.64E+00(6.42E+00)	4.03E+00(5.90E+00)	1.22E-01(4.57E-02)
F23	3.29E+02(2.87E-13)	3.29E+02(2.87E-13)	3.29E+02(2.87E-13)	3.29E+02(1.79E-04)	3.29E+02(2.87E-13)
F24	1.12E+02(2.10E+00)	1.09E+02(2.66E+00)	1.09E+02(3.33E+00)	1.06E+02(3.00E+00)	1.08E+02(2.60E+00)
F25	1.93E+02(2.28E+01)	1.17E+02(6.52E+00)	1.43E+02(3.14E+01)	1.39E+02(1.87E+01)	1.10E+02(6.57E+00)
F26	1.00E+02(2.86E-02)	1.00E+02(2.59E-02)	1.00E+02(1.84E-02)	1.00E+02(5.89E-03)	1.00E+02(1.66E-02)
F27	3.32E+02(1.41E+02)	1.71E+00(3.63E-01)	4.56E+01(1.09E+02)	2.19E+02(1.33E+02)	1.52E+00(3.01E-01)
F28	3.06E+02(1.23E-02)	3.67E+02(2.32E+01)	3.61E+02(4.06E+01)	3.33E+02(8.60E+01)	3.65E+02(1.63E+01)
F29	2.02E+02(3.04E-01)	2.22E+02(8.22E-02)	2.11E+02(2.79E+01)	2.23E+02(2.75E+00)	2.22E+02(2.02E-01)
F30	2.32E+02(6.15E+00)	4.64E+02(5.00E+00)	4.84E+02(4.69E+01)	4.91E+02(4.27E+01)	4.62E+02(1.63E+00)

F31	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	4.40E-05(1.21E-05)	0.00E+00(0.00E+00)
F32	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	8.29E-05(2.58E-05)	0.00E+00(0.00E+00)
F33	2.00E+01(1.33E-02)	1.59E+01(8.06E+00)	1.97E+01(1.93E+00)	1.89E+01(4.57E+00)	1.78E+01(5.71E+00)
F34	3.86E+00(1.11E+00)	3.43E+00(1.29E+00)	3.39E+00(1.74E+00)	3.36E-01(5.12E-01)	1.12E+00(1.47E+00)
F35	2.98E+02(1.17E+02)	6.06E+01(6.57E+01)	3.13E+02(1.97E+02)	2.27E+01(4.45E+01)	3.27E+01(5.53E+01)
F36	3.32E+01(2.53E+01)	2.42E+00(1.61E+00)	1.69E+01(2.48E+01)	5.00E+01(5.98E+01)	1.23E+00(1.71E+00)
F37	1.56E+00(1.77E-01)	2.00E-01(1.07E-01)	6.60E-01(5.01E-01)	3.46E-01(2.74E-01)	1.14E-01(9.67E-02)
F38	2.58E+00(2.98E+00)	5.66E-01(2.33E-01)	2.43E+00(4.14E+00)	1.71E+01(2.57E+01)	2.35E-01(2.50E-01)
F39	1.00E+02(3.63E-02)	1.00E+02(2.80E-02)	1.00E+02(4.99E-02)	1.00E+02(5.47E-02)	1.00E+02(2.48E-02)
F40	1.98E+02(1.16E+02)	2.17E+02(1.87E-02)	2.17E+02(3.82E+00)	2.22E+02(1.45E+01)	2.17E+02(1.12E-13)
F41	2.66E+02(1.05E+02)	2.58E+01(8.08E+01)	1.50E+02(1.49E+02)	2.13E+02(1.35E+02)	5.47E+01(1.15E+02)
F42	1.02E+02(9.66E-01)	1.01E+02(3.29E-01)	1.01E+02(3.69E-01)	1.01E+02(1.11E-01)	1.01E+02(2.13E-01)
F43	1.68E-03(9.42E-05)	3.05E-02(5.23E-05)	3.05E-02(7.32E-05)	3.09E-02(6.39E-04)	3.05E-02(5.27E-05)
F44	3.16E+02(3.35E-13)	3.67E+03(1.50E+03)	2.81E+03(1.16E+03)	3.60E+03(1.66E+03)	2.93E+03(1.43E-07)
F45	1.00E+02(0.00E+00)	1.00E+02(1.45E-13)	1.00E+02(0.00E+00)	1.00E+02(5.19E-04)	1.00E+02(0.00E+00)

Table A.4: Error values achieved by the proposed LSAOS-DE, LSHADE, SHADE, JADE, CoDE and SaDE for $30D$

problem	LSHADE	SHADE	JADE	CoDE	SaDE	LSAOSDE
	Mean(Std.)	Mean(Std.)	Mean(Std.)	Mean(Std.)	Mean(Std.)	Mean(Std.)
F01	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	2.84E+04(2.05E+04)	1.48E+04(3.03E+04)	0.00E+00(0.00E+00)
F02	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	4.30E-09(1.44E-08)	0.00E+00(0.00E+00)
F03	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)
F04	0.00E+00(0.00E+00)	1.09E-06(7.78E-06)	0.00E+00(0.00E+00)	1.27E+00(9.10E+00)	1.94E+01(2.94E+01)	7.19E-05(3.06E-04)
F05	2.01E+01(2.35E-02)	2.04E+01(3.47E-02)	2.04E+01(4.21E-02)	2.00E+01(6.19E-02)	2.06E+01(5.02E-02)	2.03E+01(8.29E-02)
F06	2.19E-02(1.09E-01)	0.00E+00(0.00E+00)	1.07E+01(5.38E+00)	2.39E+00(1.70E+00)	1.23E+00(3.36E+00)	4.49E-02(1.65E-01)
F07	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	5.80E-04(2.39E-03)	4.82E-04(3.44E-03)	0.00E+00(0.00E+00)
F08	0.00E+00(0.00E+00)	1.02E+01(1.25E+00)	1.21E-05(1.08E-05)	0.00E+00(0.00E+00)	1.27E+00(7.67E-01)	0.00E+00(0.00E+00)
F09	7.24E+00(1.82E+00)	5.22E+01(5.63E+00)	5.06E+01(5.77E+00)	3.69E+01(1.04E+01)	8.28E+01(7.46E+00)	4.69E+00(1.62E+00)
F10	4.08E-03(1.18E-02)	2.30E+02(3.34E+01)	3.60E+01(4.47E+00)	5.30E-01(5.14E-01)	1.93E+02(2.72E+01)	9.18E-03(1.59E-02)
F11	1.20E+03(1.91E+02)	3.00E+03(2.64E+02)	3.19E+03(2.34E+02)	1.85E+03(4.75E+02)	4.05E+03(2.61E+02)	1.60E+03(4.42E+02)
F12	1.60E-01(2.77E-02)	5.48E-01(6.02E-02)	6.29E-01(8.13E-02)	5.57E-02(3.79E-02)	9.19E-01(1.04E-01)	3.02E-01(1.23E-01)
F13	1.21E-01(1.63E-02)	1.70E-01(2.28E-02)	1.87E-01(2.67E-02)	2.38E-01(4.39E-02)	2.84E-01(3.12E-02)	1.22E-01(1.75E-02)
F14	2.29E-01(2.84E-02)	2.18E-01(1.96E-02)	2.33E-01(2.64E-02)	2.40E-01(4.15E-02)	2.46E-01(2.82E-02)	2.18E-01(2.38E-02)
F15	2.13E+00(1.99E-01)	6.65E+00(6.24E-01)	5.79E+00(5.45E-01)	2.89E+00(8.77E-01)	9.69E+00(7.66E-01)	2.49E+00(2.39E-01)
F16	8.40E+00(4.93E-01)	1.02E+01(2.91E-01)	1.06E+01(4.56E-01)	9.33E+00(7.70E-01)	1.14E+01(2.71E-01)	9.24E+00(5.55E-01)
F17	2.33E+02(1.25E+02)	2.96E+02(1.04E+02)	4.30E+02(2.39E+02)	1.28E+03(1.04E+03)	4.34E+02(2.19E+02)	1.36E+02(7.15E+01)
F18	7.86E+00(3.07E+00)	1.29E+01(6.26E+00)	1.41E+01(7.78E+00)	1.43E+01(6.91E+00)	5.01E+01(1.14E+01)	6.35E+00(2.56E+00)
F19	3.83E+00(6.27E-01)	4.25E+00(5.72E-01)	4.13E+00(7.12E-01)	2.60E+00(4.05E-01)	4.36E+00(4.26E-01)	3.25E+00(5.95E-01)
F20	2.74E+00(1.13E+00)	1.16E+01(2.04E+00)	1.51E+01(4.41E+00)	1.11E+01(4.67E+00)	2.58E+01(5.39E+00)	3.86E+00(1.29E+00)
F21	1.08E+02(7.81E+01)	2.38E+02(1.20E+02)	2.14E+02(1.34E+02)	2.43E+02(1.45E+02)	4.03E+02(1.09E+02)	3.22E+01(4.47E+01)
F22	2.42E+01(3.37E+00)	1.23E+02(6.12E+01)	1.86E+02(6.39E+01)	1.47E+02(9.58E+01)	2.13E+02(5.14E+01)	3.36E+01(1.71E+01)
F23	3.15E+02(3.59E-13)	3.15E+02(4.02E-13)	3.15E+02(4.02E-13)	3.15E+02(3.18E-13)	3.15E+02(7.53E-13)	3.06E+02(3.13E+01)
F24	2.24E+02(1.77E+00)	2.23E+02(1.02E+00)	2.23E+02(1.09E+00)	2.25E+02(2.42E+00)	2.24E+02(5.53E-01)	2.14E+02(1.16E+01)
F25	2.03E+02(5.09E-02)	2.03E+02(1.52E-01)	2.03E+02(2.93E-01)	2.04E+02(9.33E-01)	2.01E+02(1.98E+00)	2.02E+02(6.39E-01)
F26	1.00E+02(1.43E-02)	1.00E+02(2.10E-02)	1.00E+02(2.57E-02)	1.00E+02(5.65E-02)	1.00E+02(3.78E-02)	1.00E+02(1.66E-02)
F27	3.00E+02(1.37E-13)	3.02E+02(1.40E+01)	3.09E+02(2.84E+01)	3.83E+02(3.67E+01)	3.40E+02(4.67E+01)	3.00E+02(2.67E-02)
F28	8.46E+02(1.33E+01)	8.24E+02(1.42E+01)	8.06E+02(2.42E+01)	8.32E+02(2.72E+01)	8.48E+02(1.43E+02)	8.18E+02(9.20E+01)
F29	7.16E+02(2.80E+00)	7.18E+02(4.83E+00)	7.17E+02(3.18E+00)	7.68E+02(1.60E+02)	2.45E+02(1.07E+02)	6.85E+02(1.27E+02)
F30	1.99E+03(7.82E+02)	1.25E+03(4.69E+02)	1.08E+03(5.04E+02)	8.86E+02(3.74E+02)	6.66E+02(1.63E+02)	4.87E+02(1.12E+02)

F31	0.00E+00(0.00E+00)	1.12E+01(2.95E+01)	2.13E+00(4.82E+00)	1.49E+04(1.44E+04)	4.60E+03(5.28E+03)	0.00E+00(0.00E+00)
F32	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	6.80E+02(6.82E+02)	0.00E+00(0.00E+00)
F33	2.01E+01(2.55E-02)	2.01E+01(1.75E-02)	2.03E+01(2.95E-02)	2.01E+01(9.63E-02)	2.06E+01(3.96E-02)	2.02E+01(7.42E-02)
F34	7.39E+00(1.56E+00)	1.61E+01(2.70E+00)	2.44E+01(3.81E+00)	3.79E+01(1.15E+01)	8.83E+01(7.65E+00)	4.70E+00(1.41E+00)
F35	1.22E+03(1.72E+02)	1.47E+03(2.38E+02)	1.70E+03(2.08E+02)	1.86E+03(4.78E+02)	4.13E+03(2.92E+02)	1.43E+03(2.46E+02)
F36	2.17E+02(9.18E+01)	1.06E+03(4.46E+02)	8.74E+03(5.48E+04)	2.72E+03(3.31E+03)	4.03E+02(2.29E+02)	1.33E+02(6.55E+01)
F37	6.66E+00(6.78E-01)	7.36E+00(9.02E-01)	7.89E+00(9.43E-01)	2.70E+00(4.97E-01)	8.09E+00(5.20E-01)	6.43E+00(5.77E-01)
F38	3.93E+01(3.06E+01)	2.21E+02(1.34E+02)	8.95E+03(3.49E+04)	1.61E+02(1.24E+02)	2.01E+02(1.46E+02)	2.90E+01(1.39E+01)
F39	1.05E+02(1.25E+00)	1.03E+02(1.31E-01)	1.03E+02(2.15E-01)	1.03E+02(1.69E-01)	1.03E+02(1.58E-01)	1.03E+02(1.05E-01)
F40	6.19E+02(5.35E+01)	8.91E+02(2.36E+02)	1.43E+04(5.01E+04)	5.67E+02(2.25E+02)	5.01E+02(1.21E+02)	4.52E+02(1.23E+02)
F41	4.04E+02(4.78E+00)	4.35E+02(3.49E+01)	4.28E+02(4.25E+01)	3.90E+02(7.51E+01)	3.50E+02(5.89E+01)	4.01E+02(5.96E+00)
F42	1.08E+02(2.46E-01)	1.05E+02(3.62E-01)	1.05E+02(3.74E-01)	1.05E+02(6.27E-01)	1.05E+02(4.32E-01)	1.03E+02(1.99E-01)
F43	1.09E-02(2.22E-04)	2.63E-02(4.94E-04)	2.60E-02(2.92E-04)	2.61E-02(4.00E-04)	2.98E-02(9.27E-04)	2.64E-02(6.54E-04)
F44	4.14E+04(3.09E+03)	3.20E+04(9.41E+02)	3.21E+04(9.75E+02)	3.23E+04(1.05E+03)	3.15E+04(1.92E+02)	3.16E+04(6.91E+02)
F45	1.00E+02(1.44E-13)	1.00E+02(1.44E-13)	1.00E+02(1.44E-13)	1.00E+02(1.44E-13)	1.00E+02(2.84E-13)	1.00E+02(1.23E-13)

Table A.5: Error values achieved by the proposed LSAOS-DE, EPSDE, MPEDE, UMOEAs and AMALGAM-SO for $30D$

problem	EPSDE	MPEDE	UMOEAS	AMALGAM-SO	LSAOSDE
	Mean(Std.)	**Mean(Std.)**	**Mean(Std.)**	**Mean(Std.)**	**Mean(Std.)**
F01	6.10E+04(3.88E+05)	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	1.62E+01(1.16E+02)	0.00E+00(0.00E+00)
F02	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	4.74E-04(9.28E-05)	0.00E+00(0.00E+00)
F03	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	6.45E-04(9.55E-04)	0.00E+00(0.00E+00)
F04	7.86E-02(5.58E-01)	2.20E-04(1.35E-03)	2.57E+00(1.24E+01)	9.04E-04(1.45E-04)	7.19E-05(3.06E-04)
F05	2.04E+01(4.00E-02)	2.03E+01(6.60E-02)	2.02E+01(2.80E-01)	2.02E+01(1.13E-01)	2.03E+01(8.29E-02)
F06	1.87E+01(1.49E+00)	2.45E-01(4.24E-01)	1.86E+00(1.60E+00)	5.65E-03(2.40E-03)	4.49E-02(1.65E-01)
F07	6.76E-04(2.92E-03)	1.45E-04(1.04E-03)	0.00E+00(0.00E+00)	1.35E-0391.68E-04)	0.00E+00(0.00E+00)
F08	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	2.39E+00(1.38E+00)	2.48E-02(2.31E-02)	0.00E+00(0.00E+00)
F09	4.31E+01(7.14E+00)	2.76E+01(7.60E+00)	1.25E+01(4.47E+00)	2.97E+00(2.82E+00)	4.69E+00(1.62E+00)
F10	2.25E-01(1.95E-01)	1.40E-01(4.57E-02)	1.22E+01(2.08E+01)	1.35E+01(2.40E+01)	9.18E-03(1.59E-02)
F11	3.59E+03(4.00E+02)	1.82E+03(3.84E+02)	1.99E+03(8.82E+02)	2.67E+02(3.32E+02)	1.60E+03(4.42E+02)
F12	5.06E-01(8.57E-02)	2.17E-01(8.68E-02)	2.23E-03(2.13E-03)	1.31E-01(6.95E-02)	3.02E-01(1.23E-01)
F13	2.50E-01(3.90E-02)	1.59E-01(3.85E-02)	5.83E-02(2.55E-02)	7.64E-02(2.54E-02)	1.22E-01(1.75E-02)
F14	2.88E-01(4.89E-02)	2.19E-01(2.29E-02)	2.20E-01(4.58E-02)	3.06E-01(4.18E-02)	2.18E-01(2.38E-02)
F15	5.44E+00(7.31E-01)	2.68E+00(5.96E-01)	3.49E+00(1.15E+00)	2.68E+00(4.62E-01)	2.49E+00(2.39E-01)
F16	1.12E+01(3.76E-01)	9.23E+00(4.38E-01)	1.13E+01(5.45E-01)	9.43E+00(6.99E-01)	9.24E+00(5.55E-01)
F17	3.94E+04(3.99E+04)	9.93E+01(7.09E+01)	1.23E+03(3.20E+02)	1.07E+03(4.49E+02)	1.36E+02(7.15E+01)
F18	2.12E+02(2.75E+02)	1.04E+01(4.06E+00)	5.66E+01(2.16E+01)	7.87E+01(2.61E+01)	6.35E+00(2.56E+00)
F19	1.31E+01(1.04E+00)	3.06E+00(4.34E-01)	5.66E+00(7.45E-01)	3.91E+00(8.67E-01)	3.25E+00(5.95E-01)
F20	6.01E+01(1.10E+02)	6.61E+00(1.96E+00)	1.91E+01(9.97E+00)	5.18E+01(3.76E+01)	3.86E+00(1.29E+00)
F21	5.89E+03(6.24E+03)	7.25E+01(7.79E+01)	4.10E+02(1.98E+02)	5.72E+02(2.68E+02)	3.22E+01(4.47E+01)
F22	2.39E+02(1.00E+02)	8.87E+01(7.04E+01)	1.35E+02(7.19E+01)	1.25E+02(4.55E+01)	3.36E+01(1.71E+01)
F23	3.14E+02(8.89E-13)	3.15E+02(4.02E-13)	3.15E+02(3.44E-13)	3.15E+02(3.78E-04)	3.06E+02(3.13E+01)
F24	2.28E+02(5.46E+00)	2.24E+02(8.00E-01)	2.25E+02(4.17E+00)	2.00E+02(3.29E-02)	2.14E+02(1.16E+01)
F25	2.00E+02(1.43E-01)	2.03E+02(2.15E-01)	2.04E+02(1.05E+00)	2.02E+02(7.57E-01)	2.02E+02(6.39E-01)
F26	1.00E+02(3.66E-02)	1.00E+02(3.26E-02)	1.00E+02(7.37E-02)	1.00E+02(3.87E-02)	1.00E+02(1.66E-02)
F27	8.33E+02(1.21E+02)	3.63E+02(4.89E+01)	3.58E+02(4.15E+01)	3.00E+02(3.09E-02)	3.00E+02(2.67E-02)
F28	3.94E+02(1.26E+01)	8.19E+02(2.62E+01)	8.75E+02(3.97E+01)	7.96E+02(2.56E+01)	8.18E+02(9.20E+01)
F29	2.14E+02(1.21E+00)	6.73E+02(1.53E+02)	7.08E+02(2.10E+02)	7.11E+02(2.88E+02)	6.85E+02(1.27E+02)
F30	5.73E+02(1.27E+02)	5.20E+02(1.30E+02)	1.34E+03(5.54E+02)	1.53E+03(4.98E+02)	4.87E+02(1.12E+02)

F31	9.28E+02(4.84E+03)	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	2.72E-04(4.52E-05)	0.00E+00(0.00E+00)
F32	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	4.63E-04(7.47E-05)	0.00E+00(0.00E+00)
F33	2.03E+01(4.65E-02)	2.03E+01(5.03E-02)	2.02E+01(2.50E-01)	2.03E+01(2.30E-01)	2.02E+01(7.42E-02)
F34	4.51E+01(6.76E+00)	2.79E+01(6.68E+00)	1.19E+01(3.72E+00)	2.73E+00(2.03E+00)	4.70E+00(1.41E+00)
F35	3.67E+03(4.44E+02)	1.82E+03(3.88E+02)	2.03E+03(7.73E+02)	2.56E+02(2.42E+02)	1.43E+03(2.46E+02)
F36	5.07E+04(4.95E+04)	1.36E+02(7.63E+01)	9.96E+02(3.47E+02)	6.35E+02(3.19E+02)	1.33E+02(6.55E+01)
F37	1.23E+01(1.11E+00)	3.51E+00(6.16E-01)	6.77E+00(1.72E+00)	4.23E+00(1.53E+00)	6.43E+00(5.77E-01)
F38	5.99E+03(8.80E+03)	3.32E+01(2.04E+01)	3.82E+02(1.64E+02)	4.85E+02(3.00E+02)	2.90E+01(1.39E+01)
F39	1.03E+02(1.65E-01)	1.03E+02(1.67E-01)	1.03E+02(5.11E-01)	1.02E+02(1.38E-01)	1.03E+02(1.05E-01)
F40	9.58E+03(1.69E+04)	3.72E+02(1.08E+02)	9.26E+02(2.01E+02)	1.11E+03(2.86E+02)	4.52E+02(1.23E+02)
F41	9.51E+02(8.82E+01)	3.79E+02(4.85E+01)	4.32E+02(8.30E+01)	4.09E+02(3.18E+01)	4.01E+02(5.96E+00)
F42	1.97E+02(1.27E+01)	1.04E+02(3.89E-01)	1.03E+02(6.83E-01)	1.03E+02(5.14E-01)	1.03E+02(1.99E-01)
F43	6.28E-03(1.29E-04)	2.59E-02(2.53E-04)	2.69E-02(8.98E-04)	3.15E-02(3.24E-03)	2.64E-02(6.54E-04)
F44	6.91E+03(1.04E+04)	3.20E+04(8.98E+02)	3.19E+04(9.84E+02)	3.12E+04(4.28E+02)	3.16E+04(6.91E+02)
F45	1.00E+02(1.21E-13)	1.00E+02(1.44E-13)	1.00E+02(1.44E-13)	1.00E+02(9.54E-04)	1.00E+02(1.23E-13)

Table A.6: Error values achieved by the proposed LSAOS-DE, LSHADE, SHADE, JADE, CoDE and SaDE for $50D$

problem	LSHADE	SHADE	JADE	CoDE	SaDE	LSAOSDE
	Mean(Std.)	**Mean(Std.)**	**Mean(Std.)**	**Mean(Std.)**	**Mean(Std.)**	**Mean(Std.)**
F01	1.49E+03(1.96E+03)	9.49E+03(8.76E+03)	4.12E+03(4.27E+03)	2.18E+05(1.10E+05)	4.28E+05(1.83E+05)	1.23E+03(2.50E+03)
F02	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	1.36E+02(2.60E+02)	2.72E+03(1.56E+03)	0.00E+00(0.00E+00)
F03	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	1.85E+03(2.05E+03)	2.85E+01(5.16E+01)	5.91E-02(2.04E-01)	0.00E+00(0.00E+00)
F04	6.94E+01(4.29E+01)	5.49E+01(4.08E+01)	3.40E+01(3.99E+01)	3.16E+01(3.70E+01)	1.04E+02(2.79E+01)	3.93E+01(3.93E+01)
F05	2.03E+01(3.33E-02)	2.06E+01(3.28E-02)	2.06E+01(8.10E-02)	2.00E+01(8.54E-02)	2.08E+01(4.03E-02)	2.04E+01(2.30E-03)
F06	2.52E-01(5.39E-01)	7.52E-02(3.09E-01)	9.80E+00(1.30E+01)	8.36E+00(3.65E+00)	3.34E+00(1.46E+00)	1.20E+00(7.82E-01)
F07	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	1.30E-03(3.09E-03)	8.22E-03(3.18E-03)	3.09E-03(4.65E-03)
F08	0.00E+00(0.00E+00)	6.12E+01(4.58E+00)	1.36E+01(1.27E+00)	6.44E-01(7.13E-01)	5.62E+01(3.17E+00)	0.00E+00(0.00E+00)
F09	1.17E+01(2.23E+00)	1.70E+02(1.22E+01)	1.29E+02(9.03E+00)	7.94E+01(1.83E+01)	2.12E+02(9.83E+000-	1.70E+01(2.96E+00)
F10	4.78E-02(2.27E-02)	2.37E+03(2.13E+02)	6.11E+02(6.48E+01)	5.26E+00(2.63E+00)	1.93E+03(1.92E+02)	1.31E+00(1.49E+00)
F11	3.25E+03(2.51E+02)	7.95E+03(5.73E+02)	7.61E+03(3.80E+02)	4.30E+03(6.50E+02)	9.04E+03(3.40E+02)	4.02E+03(7.88E+02)
F12	2.10E-01(2.43E-02)	8.22E-01(8.23E-02)	8.52E-01(8.10E-02)	8.10E-02(4.70E-02)	1.22E+00(1.26E-01)	3.71E-01(2.27E-02)
F13	1.55E-01(1.92E-02)	2.44E-01(2.41E-02)	2.84E-01(3.07E-02)	3.28E-01(5.85E-02)	4.26E-01(3.65E-02)	1.97E-01(2.04E-02)
F14	3.02E-01(3.14E-02)	2.75E-01(2.84E-02)	2.94E-01(2.41E-02)	2.78E-01(3.10E-02)	3.08E-01(1.95E-02)	2.94E-01(1.88E-02)
F15	5.10E+00(4.91E-01)	1.89E+01(1.25E+00)	1.44E+01(8.81E-01)	6.48E+00(1.42E+00)	2.23E+01(1.49E+00)	6.08E+00(8.36E-01)
F16	1.68E+01(4.36E-01)	2.01E+01(8.64E-01)	2.00E+01(8.17E-01)	1.80E+01(1.04E+00)	2.06E+01(2.58E-01)	1.80E+01(9.65E-01)
F17	1.67E+03(4.53E+02)	1.89E+03(4.66E+02)	1.86E+03(4.61E+02)	1.47E+04(7.39E+03)	1.44E+03(3.94E+02)	1.04E+03(3.56E+02)
F18	1.06E+02(1.71E+01)	9.64E+01(2.22E+01)	1.15E+02(2.07E+01)	3.03E+02(3.59E+02)	9.94E+01(1.42E+01)	8.60E+01(1.06E+01)
F19	7.73E+00(1.74E+00)	1.14E+01(5.74E-01)	1.14E+01(7.35E-01)	6.75E+00(1.56E+00)	1.19E+01(3.83E+00)	1.06E+01(8.51E-01)
F20	1.49E+01(4.93E+00)	3.21E+01(1.63E+01)	2.26E+01(1.30E+01)	1.73E+02(1.54E+02)	8.34E+01(1.68E+01)	1.18E+01(3.98E+00)
F21	5.64E+02(1.62E+02)	7.03E+02(2.28E+02)	7.71E+02(2.35E+02)	9.74E+03(7.69E+03)	1.11E+03(3.69E+02)	4.15E+02(1.20E+02)
F22	9.83E+01(6.64E+01)	5.13E+02(1.25E+02)	7.74E+02(1.34E+02)	5.54E+02(1.97E+02)	7.37E+02(1.29E+02)	1.80E+02(8.48E+01)
F23	3.44E+02(3.84E-13)	3.44E+02(3.03E-13)	3.44E+02(2.78E-13)	3.44E+02(2.87E-13)	3.44E+02(2.21E-08)	3.41E+02(2.03E+01)
F24	2.75E+02(6.45E-01)	2.73E+02(1.39E+00)	2.73E+02(1.25E+00)	2.71E+02(2.34E+00)	2.71E+02(1.98E+00)	2.52E+02(3.13E+01)
F25	2.05E+02(2.99E-01)	2.05E+02(3.77E-01)	2.06E+02(2.18E+00)	2.09E+02(5.65E+00)	2.00E+02(8.68E-08)	2.04E+02(2.62E+00)
F26	1.00E+02(1.78E-02)	1.00E+02(3.42E-02)	1.00E+02(2.96E-02)	1.00E+02(8.38E-02)	1.26E+02(4.39E+01)	1.00E+02(1.74E-02)
F27	3.44E+02(2.78E+01)	3.25E+02(2.75E+01)	3.26E+02(3.06E+01)	5.39E+02(7.20E+01)	3.79E+02(4.57E+01)	3.24E+02(2.35E+01)
F28	1.12E+03(3.58E+01)	1.10E+03(2.34E+01)	1.09E+03(2.73E+01)	1.18E+03(5.53E+01)	1.52E+03(1.47E+02)	1.11E+03(1.34E+02)
F29	8.05E+02(3.46E+01)	8.01E+02(3.09E+01)	7.99E+02(4.10E+01)	9.65E+02(8.89E+01)	5.85E+02(9.10E+01)	7.91E+02(4.87E+01)
F30	8.78E+03(5.23E+02)	8.52E+03(3.47E+02)	8.64E+03(3.59E+02)	9.07E+03(4.84E+02)	1.11E+04(1.09E+03)	8.57E+03(4.19E+02)

F31	1.01E+03(2.09E+03)	8.83E+03(5.59E+03)	8.39E+03(5.61E+03)	1.93E+05(1.02E+05)	2.52E+05(1.26E+050-	1.74E+03(1.28E+03)
F32	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	1.09E-08(3.82E-08)	4.02E+03(2.14E+03)	0.00E+00(0.00E+00)
F33	2.03E+01(4.37E-02)	2.01E+01(1.68E-02)	2.04E+01(3.45E-02)	2.00E+01(7.59E-02)	2.08E+01(3.46E-02)	2.04E+01(6.02E-04)
F34	1.21E+01(2.71E+00)	3.43E+01(5.09E+00)	5.45E+01(9.01E+00)	7.96E+01(1.68E+01)	2.18E+02(1.30E+01)	1.79E+01(3.43E+00)
F35	2.89E+03(2.53E+02)	2.98E+03(3.21E+02)	3.48E+03(3.63E+02)	4.15E+03(7.44E+02)	8.69E+03(3.92E+02)	3.51E+03(6.24E+02)
F36	1.91E+03(3.93E+02)	3.36E+03(2.33E+03)	2.68E+03(8.19E+02)	2.33E+04(1.50E+04)	1.68E+03(5.04E+02)	1.07E+03(2.84E+02)
F37	4.03E+01(6.00E-01)	4.15E+01(1.49E+00)	4.29E+01(4.10E+00)	3.81E+01(8.16E+00)	4.00E+01(1.34E+01)	4.03E+01(2.78E+00)
F38	3.48E+02(1.54E+02)	1.30E+03(3.82E+02)	1.27E+03(3.94E+02)	6.97E+03(5.53E+03)	8.20E+02(4.05E+02)	1.75E+02(8.28E+01)
F39	1.02E+02(7.38E-02)	1.05E+02(2.91E-01)	1.05E+02(3.56E-01)	1.04E+02(2.99E-01)	1.06E+02(2.42E-01)	1.04E+02(1.62E-01)
F40	1.14E+03(2.42E+02)	1.80E+03(3.72E+02)	1.23E+04(5.24E+04)	1.38E+03(2.98E+02)	1.74E+03(4.32E+02)	9.02E+02(9.65E+01)
F41	4.22E+02(2.07E+01)	4.49E+02(2.45E+01)	4.81E+02(4.90E+01)	5.53E+02(6.18E+01)	5.35E+02(4.75E+01)	4.26E+02(2.15E+01)
F42	1.18E+02(1.71E+01)	1.55E+02(4.70E+01)	1.53E+02(4.70E+01)	1.33E+02(4.18E+01)	1.09E+02(4.81E-01)	1.07E+02(2.80E-01)
F43	2.53E-02(3.11E-04)	8.18E-02(3.39E-03)	7.76E-02(3.30E-03)	8.15E-02(4.15E-03)	1.16E-01(6.25E-03)	8.41E-02(4.33E-03)
F44	5.27E+04(1.00E+01)	6.26E+04(9.76E+03)	6.46E+04(9.13E+03)	6.46E+04(9.49E+03)	4.95E+04(9.96E+00)	6.16E+04(5.39E+03)
F45	1.00E+02(1.04E-13)	1.00E+02(6.88E-14)	1.00E+02(1.21E-13)	1.00E+02(1.44E-13)	1.00E+02(1.07E-07)	1.00E+02(1.65E-13)

Table A.7: Error values achieved by the proposed LSAOS-DE, EPSDE, MPEDE, UMOEAs and AMALGAM-SO for 50D

problem	EPSDE	MPEDE	UMOEAS	AMALGAM-SO	LSAOSDE
	Mean(Std.)	**Mean(Std.)**	**Mean(Std.)**	**Mean(Std.)**	**Mean(Std.)**
F01	2.19E+06(5.90E+06)	5.98E+03(6.15E+03)	0.00E+00(0.00E+00)	1.83E-02(1.11E-03)	1.23E+03(2.50E+03)
F02	8.01E-09(1.65E-08)	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	1.08E+04(1.89E+04)	0.00E+00(0.00E+00)
F03	2.42E-08(9.86E-08)	1.89E-04(3.42E-04)	0.00E+00(0.00E+00)	9.85E-04(6.09E-04)	0.00E+00(0.00E+00)
F04	2.82E+01(2.10E+01)	2.59E+01(3.35E+01)	9.05E+01(2.67E+01)	3.84E+01(3.13E+01)	3.93E+01(3.93E+01)
F05	2.06E+01(3.31E-02)	2.04E+01(7.67E-02)	2.02E+01(3.00E-01)	2.00E+01(1.56E-01)	2.04E+01(2.30E-03)
F06	4.67E+01(3.21E+00)	2.17E+00(1.36E+00)	4.12E+00(2.57E+00)	9.50E+00(4.79E-01)	1.20E+00(7.82E-01)
F07	5.55E-03(6.87E-03)	1.16E-03(2.97E-03)	0.00E+00(0.00E+00)	2.44E-03(2.10E-04)	3.09E-03(4.65E-03)
F08	7.80E-02(2.70E-01)	0.00E+00(0.00E+00)	5.94E+00(3.36E+00)	4.26E+01(2.68E-01)	0.00E+00(0.00E+00)
F09	1.46E+02(1.80E+01)	5.09E+01(9.95E+00)	2.28E+01(5.21E+00)	9.06E+01(4.02E+00)	1.70E+01(2.96E+00)
F10	5.89E+02(6.22E+02)	3.44E+00(1.62E+00)	2.07E+02(2.36E+02)	1.88E+03(2.37E+01)	1.31E+00(1.49E+00)
F11	9.03E+03(7.20E+02)	4.16E+03(6.81E+02)	4.05E+03(1.27E+03)	4.35E+03(4.23E+02)	4.02E+03(7.88E+02)
F12	8.62E-01(9.60E-02)	2.90E-01(7.97E-02)	1.27E-03(8.53E-04)	3.00E-02(7.73E-02)	3.71E-01(2.27E-02)
F13	3.69E-01(5.04E-02)	2.46E-01(3.34E-02)	9.58E-02(3.47E-02)	2.90E-01(3.41E-02)	1.97E-01(2.04E-02)
F14	3.46E-01(1.23E-01)	2.76E-01(2.88E-02)	2.85E-01(5.03E-02)	5.24E-01(3.34E-02)	2.94E-01(1.88E-02)
F15	1.80E+01(2.06E+00)	5.56E+00(1.02E+00)	6.13E+00(1.20E+00)	6.21E+00(4.81E-01)	6.08E+00(8.36E-01)
F16	2.07E+01(4.68E-01)	1.77E+01(6.80E-01)	2.02E+01(5.50E-01)	2.06E+01(1.28E+00)	1.80E+01(9.65E-01)
F17	2.05E+05(1.55E+05)	1.15E+03(3.60E+02)	2.50E+03(4.91E+02)	2.77E+03(4.43E+02)	1.04E+03(3.56E+02)
F18	4.02E+03(6.52E+03)	9.20E+01(1.15E+01)	1.31E+02(4.08E+01)	2.25E+02(5.57E+01)	8.60E+01(1.06E+01)
F19	2.47E+01(1.51E+00)	8.25E+00(1.24E+00)	1.09E+01(1.81E+00)	2.13E+01(2.57E+00)	1.06E+01(8.51E-01)
F20	6.36E+02(1.64E+03)	2.43E+01(7.47E+00)	1.28E+02(4.02E+01)	3.73E+02(1.25E+02)	1.18E+01(3.98E+00)
F21	7.43E+04(5.11E+04)	4.70E+02(1.40E+02)	1.50E+03(3.95E+02)	1.71E+03(3.76E+02)	4.15E+02(1.20E+02)
F22	7.98E+02(1.64E+02)	4.31E+02(1.81E+02)	3.67E+02(1.88E+02)	8.15E+02(1.18E+02)	1.80E+02(8.48E+01)
F23	3.37E+02(3.20E-12)	3.44E+02(4.73E-13)	3.44E+02(2.85E-13)	3.44E+02(7.75E-04)	3.41E+02(2.03E+01)
F24	2.72E+02(5.75E+00)	2.75E+02(1.08E+00)	2.73E+02(2.24E+00)	2.74E+02(2.03E+00)	2.52E+02(3.13E+01)
F25	2.01E+02(2.24E+00)	2.05E+02(1.66E+00)	2.08E+02(2.66E+00)	2.01E+02(2.26E+00)	2.04E+02(2.62E+00)
F26	1.00E+02(4.96E-02)	1.00E+02(3.19E-02)	1.00E+02(1.06E-01)	1.54E+02(8.49E-02)	1.00E+02(1.74E-02)
F27	1.57E+03(4.62E+01)	3.56E+02(4.41E+01)	4.55E+02(6.50E+01)	5.99E+02(2.95E+01)	3.24E+02(2.35E+01)
F28	3.87E+02(1.22E+01)	1.15E+03(3.81E+01)	1.29E+03(7.01E+01)	1.70E+03(4.74E+01)	1.11E+03(1.34E+02)
F29	2.25E+02(2.26E+00)	7.94E+02(5.77E+01)	8.13E+02(9.79E+01)	7.20E+02(2.57E+02)	7.91E+02(4.87E+01)
F30	1.08E+03(1.90E+02)	8.79E+03(4.33E+02)	8.99E+03(7.39E+02)	1.48E+04(7.71E+02)	8.57E+03(4.19E+02)

F31	3.40E+06(1.12E+07)	2.69E+03(3.11E+03)	0.00E+00(0.00E+00)	1.47E-03(1.68E-03)	1.74E+03(1.28E+03)
F32	6.91E-09(1.51E-08)	0.00E+00(0.00E+00)	0.00E+00(0.00E+00)	2.04E-03(8.07E-03)	0.00E+00(0.00E+00)
F33	2.06E+01(3.38E-02)	2.04E+01(5.75E-02)	2.01E+01(2.54E-01)	2.02E+01(5.95E-02)	2.04E+01(6.02E-04)
F34	1.51E+02(1.61E+01)	5.39E+01(1.28E+01)	2.37E+01(5.89E+00)	4.57E+00(3.29E+00)	1.79E+01(3.43E+00)
F35	8.80E+03(6.23E+02)	3.92E+03(5.67E+02)	4.08E+03(1.21E+03)	5.64E+02(4.79E+02)	3.51E+03(6.24E+02)
F36	1.80E+05(1.11E+05)	1.21E+03(3.99E+02)	2.59E+03(5.47E+02)	2.29E+03(4.78E+02)	1.07E+03(2.84E+02)
F37	2.52E+01(8.06E+00)	4.01E+01(7.45E-01)	4.41E+01(5.01E+00)	4.10E+01(3.97E+00)	4.03E+01(2.78E+00)
F38	9.73E+04(8.44E+04)	2.24E+02(1.52E+02)	1.46E+03(3.60E+02)	1.40E+03(3.56E+02)	1.75E+02(8.28E+01)
F39	1.04E+02(2.51E-01)	1.05E+02(3.15E-01)	1.04E+02(3.13E-01)	1.04E+02(2.56E-01)	1.04E+02(1.62E-01)
F40	4.50E+03(3.19E+03)	9.30E+02(1.37E+02)	1.89E+03(3.40E+02)	2.16E+03(3.82E+02)	9.02E+02(9.65E+01)
F41	1.64E+03(4.08E+01)	4.57E+02(2.50E+01)	4.68E+02(6.32E+01)	4.07E+02(5.25E+00)	4.26E+02(2.15E+01)
F42	2.00E+02(8.81E-05)	1.28E+02(4.02E+01)	1.37E+02(4.35E+01)	1.07E+02(1.13E+00)	1.07E+02(2.80E-01)
F43	1.12E-02(9.20E-05)	8.13E-02(3.84E-03)	8.92E-02(4.64E-03)	1.04E-01(8.24E-03)	8.41E-02(4.33E-03)
F44	1.35E+04(2.49E+04)	6.17E+04(8.84E+03)	6.02E+04(8.19E+03)	4.97E+04(1.33E+03)	6.16E+04(5.39E+03)
F45	1.00E+02(1.19E-01)	1.00E+02(6.96E-14)	1.00E+02(1.44E-13)	1.00E+02(8.29E-04)	1.00E+02(1.65E-13)

Table A.8: Mean error results and standard deviation achieved by the proposed algorithm and seven other heuristic- based selection methods for 10D

	AOSDE-LS	AOSDE-SR	AOSDE-FIR	AOSDE-PR	AOSDE-PBM	AOSDE-RD	AOSDE-PAOC	LSAOS-DE
	Mean(Std.)	Mean(Std.)	Mean(Std.)	Mean(Std.)	Mean(Std.)	Mean(Std.)	Mean(Std.)	Mean(Std.)
F01	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**
F02	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**
F03	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**
F04	**1.65E+01(1.74E+01)**	1.99E+01(1.73E+01)	2.01E+01(1.70E+01)	1.99E+01(1.73E+01)	**1.65E+01(1.74E+01)**	**1.65E+01(1.74E+01)**	1.99E+01(1.73E+01)	**1.65E+01(1.74E+01)**
F05	1.41E+01(7.33E+00)	1.67E+01(6.09E+00)	1.59E+01(6.76E+00)	1.65E+01(6.34E+00)	1.41E+01(7.88E+00)	1.41E+01(7.29E+00)	1.72E+01(6.05E+00)	**1.38E+01(7.90E+00)**
F06	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**
F07	8.21E-04(3.06E-03)	6.04E-03(8.73E-03)	5.22E-03(7.66E-03)	5.22E-03(9.31E-03)	3.87E-04(1.99E-03)	**1.45E-04(1.04E-03)**	5.60E-03(1.18E-02)	1.01E-03(3.31E-03)
F08	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**
F09	1.03E+00(1.73E+00)	3.47E+00(1.51E+00)	3.06E+00(1.29E+00)	3.79E+00(1.97E+00)	1.06E+00(1.62E+00)	1.13E+00(1.66E+00)	2.70E+00(1.26E+00)	**8.42E-01(1.29E+00)**
F10	4.90E-03(2.11E-02)	**0.00E+00(0.00E+00)**	1.22E-03(8.75E-03)	3.67E-03(1.48E-02)	1.22E-03(8.75E-03)	**0.00E+00(0.00E+00)**	2.45E-03(1.22E-02)	2.45E-03(1.75E-02)
F11	7.20E+01(5.26E+01)	7.39E+01(5.23E+01)	7.03E+01(6.20E+01)	7.70E+01(6.67E+01)	6.56E+01(4.94E+01)	8.17E+01(5.84E+01)	**4.77E+01(5.42E+01)**	5.15E+01(4.88E+01)
F12	1.93E-01(9.88E-02)	2.16E-01(9.80E-02)	1.67E-01(1.04E-01)	2.17E-01(9.02E-02)	2.03E-01(8.86E-02)	1.89E-01(9.71E-02)	**1.12E-01(6.83E-02)**	1.49E-01(9.30E-02)
F13	4.38E-02(1.76E-02)	**2.91E-02(1.21E-02)**	4.40E-02(1.64E-02)	3.04E-02(1.18E-02)	3.63E-02(1.73E-02)	3.96E-02(1.84E-02)	5.06E-02(1.68E-02)	5.06E-02(1.34E-02)
F14	9.61E-02(3.44E-02)	**8.98E-02(3.04E-02)**	1.06E-01(3.31E-02)	9.19E-02(3.29E-02)	9.94E-02(3.05E-02)	1.07E-01(3.63E-02)	9.26E-02(3.09E-02)	1.06E-01(3.07E-02)
F15	5.10E-01(1.99E-01)	6.17E-01(1.67E-01)	5.05E-01(1.75E-01)	5.93E-01(2.07E-01)	5.51E-01(2.04E-01)	5.49E-01(1.98E-01)	4.17E-01(7.25E-02)	**4.03E-01(6.57E-02)**
F16	1.28E+00(4.20E-01)	**1.17E+00(5.36E-01)**	1.36E+00(4.41E-01)	**1.17E+00(4.94E-01)**	1.32E+00(3.94E-01)	1.24E+00(3.91E-01)	1.30E+00(3.00E-01)	1.26E+00(3.42E-01)
F17	1.18E+00(1.63E+00)	1.47E+00(1.04E+00)	1.05E+00(1.63E+00)	1.09E+00(9.49E-01)	1.11E+00(1.09E+00)	1.05E+00(9.50E-01)	1.14E+00(1.23E+00)	**1.04E+00(1.01E+00)**
F18	**9.94E-02(1.15E-01)**	1.34E-01(1.68E-01)	2.34E-01(3.64E-01)	1.71E-01(2.41E-01)	1.37E-01(2.07E-01)	1.38E-01(1.88E-01)	1.74E-01(1.73E-01)	1.30E-01(1.43E-01)
F19	1.15E-01(7.12E-02)	1.33E-01(1.39E-01)	1.29E-01(1.53E-01)	1.18E-01(6.33E-02)	1.07E-01(7.57E-02)	9.23E-02(6.22E-02)	**6.85E-02(3.84E-02)**	7.84E-02(5.92E-02)
F20	1.84E-01(1.60E-01)	1.78E-01(1.47E-01)	2.05E-01(1.73E-01)	2.54E-01(1.80E-01)	2.30E-01(1.60E-01)	2.49E-01(1.89E-01)	**1.74E-01(1.47E-01)**	1.75E-01(1.67E-01)
F21	3.60E-01(2.72E-01)	3.18E-01(1.96E-01)	3.10E-01(2.78E-01)	2.88E-01(2.17E-01)	3.55E-01(2.90E-01)	2.84E-01(2.23E-01)	**2.64E-01(2.45E-01)**	2.88E-01(2.43E-01)
F22	1.48E-01(5.28E-02)	1.60E-01(6.16E-02)	1.44E-01(5.21E-02)	1.55E-01(5.27E-02)	1.49E-01(5.62E-02)	1.42E-01(4.37E-02)	1.27E-01(5.37E-02)	**1.22E-01(4.57E-02)**
F23	**3.29E+02(2.87E-13)**	**3.29E+02(2.87E-13)**	**3.29E+02(2.87E-13)**	**3.29E+02(2.87E-13)**	**3.29E+02(2.87E-13)**	**3.29E+02(2.87E-13)**	**3.23E+02(2.87E-13)**	**3.29E+02(2.87E-13)**
F24	1.09E+02(2.12E+00)	1.10E+02(2.89E+00)	1.09E+02(2.90E+00)	1.10E+02(2.96E+00)	1.09E+02(2.75E+00)	1.09E+02(3.34E+00)	**1.08E+02(3.06E+00)**	**1.08E+02(2.60E+00)**
F25	1.14E+02(6.15E+00)	1.18E+02(5.74E+00)	1.12E+02(6.54E+00)	1.18E+02(5.48E+00)	1.16E+02(1.21E+01)	1.17E+02(1.32E+01)	1.11E+02(5.94E+00)	**1.10E+02(6.57E+00)**
F26	**1.00E+02(1.81E-02)**	**1.00E+02(1.01E-02)**	**1.00E+02(2.03E-02)**	**1.00E+02(1.74E-02)**	**1.00E+02(1.70E-02)**	**1.00E+02(1.93E-02)**	**1.00E+02(1.64E-02)**	**1.00E+02(1.66E-02)**
F27	1.56E+00(2.95E-01)	7.27E+00(4.18E+01)	1.44E+00(3.41E-01)	7.26E+00(4.18E+01)	1.48E+00(3.74E-01)	**1.43E+00(3.73E-01)**	7.31E+00(4.18E+01)	1.52E+00(3.01E-01)
F28	3.65E+02(1.63E+01)	**3.62E+02(6.51E+00)**	3.64E+02(6.39E+00)	**3.62E+02(6.50E+00)**	3.66E+02(1.62E+01)	3.65E+02(1.45E+01)	**3.62E+02(6.50E+00)**	3.65E+02(1.63E+01)
F29	**2.22E+02(2.02E-01)**	**2.22E+02(3.09E-01)**	**2.22E+02(3.10E-01)**	**2.22E+02(2.89E-01)**	**2.22E+02(2.33E-01)**	**2.22E+02(2.04E-01)**	**2.22E+02(2.97E-01)**	**2.22E+02(2.02E-01)**
F30	**4.62E+02(3.48E+00)**	4.63E+02(2.82E+00)	4.63E+02(4.24E+00)	4.65E+02(6.18E+00)	4.64E+02(4.99E+00)	4.66E+02(1.01E+01)	**4.62E+02(1.64E+00)**	**4.62E+02(1.63E+00)**

F31	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**
F32	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**
F33	1.72E+01(5.58E+00)	1.77E+01(4.84E+00)	1.75E+01(5.22E+00)	**1.65E+01(5.84E+00)**	1.68E+01(6.42E+00)	**1.65E+01(6.16E+00)**	1.98E+01(1.09E+00)	1.78E+01(5.71E+00)
F34	2.04E+00(2.35E+00)	2.48E+00(9.18E-01)	2.70E+00(1.13E+00)	3.39E+00(9.63E-01)	2.01E+00(2.23E+00)	1.69E+00(2.11E+00)	9.68E+00(2.15E+00)	**1.12E+00(1.47E+00)**
F35	3.37E+01(2.68E+01)	**2.35E+01(2.60E+01)**	2.99E+01(3.18E+01)	6.83E+01(6.19E+01)	3.68E+01(4.12E+01)	4.01E+01(3.69E+01)	3.87E+02(1.06E+02)	3.27E+01(5.53E+01)
F36	1.07E+00(1.92E+00)	1.14E+00(1.16E+00)	1.14E+00(1.16E+00)	1.17E+00(9.88E-01)	**8.58E-01(1.00E+00)**	9.45E-01(8.14E-01)	2.73E+01(8.50E+00)	1.23E+00(1.71E+00)
F37	1.37E-01(1.44E-01)	1.45E-01(1.87E-01)	1.59E-01(1.89E-01)	1.73E-01(1.26E-01)	1.34E-01(9.43E-02)	1.46E-01(1.24E-01)	1.17E+00(2.76E-01)	**1.14E-01(9.67E-02)**
F38	4.40E-01(3.32E-01)	2.49E-01(2.75E-01)	2.49E-01(2.75E-01)	3.13E-01(2.91E-01)	3.68E-01(2.92E-01)	3.92E-01(2.77E-01)	2.80E+00(1.49E+00)	**2.35E-01(2.50E-01)**
F39	**1.00E+02(2.65E-02)**	**1.00E+02(3.06E-02)**	**1.00E+02(3.16E-02)**	**1.00E+02(2.92E-02)**	**1.00E+02(2.67E-02)**	**1.00E+02(3.14E-02)**	**1.00E+02(2.56E-02)**	**1.00E+02(2.48E-02)**
F40	**2.17E+02(9.59E-03)**	**2.17E+02(6.04E-03)**	**2.17E+02(6.04E-03)**	**2.17E+02(1.28E-02)**	**2.17E+02(9.13E-03)**	**2.17E+02(8.94E-03)**	**2.17E+02(2.59E-02)**	**2.17E+02(1.12E-13)**
F41	5.47E+01(1.15E+02)	6.04E+01(1.20E+02)	6.03E+01(1.20E+02)	6.02E+01(1.20E+02)	**5.46E+01(1.15E+02)**	5.48E+01(1.15E+02)	5.52E+01(1.14E+02)	5.47E+01(1.15E+02)
F42	**1.01E+02(2.58E-01)**	**1.01E+02(1.89E-01)**	**1.01E+02(2.06E-01)**	**1.01E+02(1.83E-01)**	**1.01E+02(2.24E-01)**	**1.01E+02(1.95E-01)**	**1.01E+02(2.26E-01)**	**1.01E+02(2.13E-01)**
F43	**3.05E-02(4.75E-05)**	**3.05E-02(5.01E-05)**	**3.05E-02(5.01E-05)**	**3.05E-02(4.85E-05)**	**3.05E-02(4.85E-05)**	**3.05E-02(4.51E-05)**	**3.05E-02(5.14E-05)**	**3.05E-02(5.27E-05)**
F44	3.08E+03(7.35E+02)	2.95E+03(6.64E+02)	2.95E+03(6.64E+02)	2.95E+03(6.64E+02)	3.08E+03(7.35E+02)	3.08E+03(7.35E+02)	2.93E+03(2.23E-06)	**2.93E+03(1.43E-07)**
F45	**1.00E+02(0.00E+00)**	**1.00E+02(0.00E+00)**	**1.00E+02(0.00E+00)**	**1.00E+02(7.51E-14)**	**1.00E+02(3.22E-14)**	**1.00E+02(3.22E-14)**	**1.00E+02(6.43E-14)**	**1.00E+02(0.00E+00)**

Table A.9: Mean error results and standard deviation achieved by the proposed algorithm and seven other heuristic- based selection methods for 30D

	AOSDE-LS	AOSDE-SR	AOSDE-FIR	AOSDE-PR	AOSDE-PBM	AOSDE-RD	AOSDE-PAOC	LSAOS-DE
	Mean(Std.)	Mean(Std.)	Mean(Std.)	Mean(Std.)	Mean(Std.)	Mean(Std.)	Mean(Std.)	Mean(Std.)
F01	1.33E+03(3.58E+03)	1.69E+02(2.67E+020	1.83E+03(7.93E+03)	1.54E+02(1.94E+02)	2.17E+03(4.97E+03)	2.55E+03(5.63E+03	1.37E-08(3.88E-08)	**0.00E+00(0.00E+00)**
F02	1.54E+03(2.99E+03)	0.00E+00(0.00E+00)	1.37E+03(2.89E+03)	**0.00E+00(0.00E+00)**	9.92E+02(2.69E+03)	2.08E+03(3.70E+03)	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**
F03	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	7.04E-09(3.26E-08)	**0.00E+00(0.00E+00)**	2.68E-06(1.72E-05)	1.01E-06(4.68E-06)	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**
F04	4.48E+00(1.79E+01)	1.35E+00(1.06E+00)	1.27E+01(2.92E+01)	1.45E+00(1.05E+00)	1.55E+01(2.94E+01)	2.06E+01(3.43E+01)	**5.74E-06(1.96E-05)**	7.19E-05(3.06E-04)
F05	**2.02E+01(8.41E-02)**	2.03E+01(7.14E-02)	2.03E+01(6.51E-02)	2.03E+01(9.00E-02)	2.03E+01(1.05E-01)	2.03E+01(8.07E-02)	2.03E+01(5.11E-02)	2.03E+01(8.29E-02)
F06	1.19E-01(3.50E-01)	2.47E-01(1.76E+00)	6.13E-01(2.00E+00)	2.79E-01(1.76E+00)	1.79E-01(3.45E-01)	3.80E-01(6.93E-01)	8.20E-02(2.65E-01)	**4.49E-02(1.65E-01)**
F07	1.80E-07(1.24E-06)	**0.00E+00(0.00E+00)**	2.59E-07(1.28E-06)	**0.00E+00(0.00E+00)**	9.13E-08(4.61E-07)	1.29E-07(4.34E-07)	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**
F08	2.61E-07(1.14E-06)	5.44E-10(2.73E-09)	4.07E-06(2.80E-05)	2.70E-09(1.76E-08)	2.96E-07(1.70E-06)	8.30E-06(4.53E-05)	6.05E-07(2.47E-06)	**0.00E+00(0.00E+00)**
F09	6.45E+00(4.67E+00)	2.02E+01(6.90E+00)	1.06E+01(5.42E+00)	1.98E+01(5.79E+00)	8.80E+00(6.46E+00)	9.83E+00(8.18E+00)	9.04E+00(4.51E+00)	**4.69E+00(1.62E+00)**
F10	9.39E-03(1.53E-02)	6.17E-03(1.60E-02)	8.32E-03(1.37E-02)	9.39E-03(1.53E-02)	6.38E-03(1.32E-02)	6.29E-03(1.24E-02)	**4.15E-03(1.04E-02)**	9.18E-03(1.59E-02)
F11	1.89E+03(4.10E+02)	1.96E+03(4.69E+02)	1.87E+03(4.69E+02)	2.03E+03(3.73E+02)	1.73E+03(4.78E+02)	1.85E+03(4.32E+02)	1.74E+03(3.75E+02)	**1.60E+03(4.42E+02)**
F12	2.97E-01(1.21E-01)	3.37E-01(1.12E-01)	3.24E-01(1.21E-01)	3.52E-01(1.01E-01)	3.32E-01(1.26E-01)	3.01E-01(1.28E-01)	**2.35E-01(7.94E-02)**	3.02E-01(1.23E-01)
F13	1.17E-01(2.69E-02)	**9.48E-02(1.97E-02)**	1.23E-01(2.01E-02)	1.27E-01(1.42E-02)	1.09E-01(2.87E-02)	1.17E-01(2.46E-02)	1.28E-01(1.30E-02)	1.22E-01(1.75E-02)
F14	2.22E-01(2.74E-02)	2.29E-01(2.67E-02)	2.41E-01(2.21E-02)	2.31E-01(2.93E-02)	2.29E-01(3.36E-02)	2.31E-01(3.67E-02)	2.31E-01(2.59E-02)	**2.18E-01(2.38E-02)**
F15	2.75E+00(4.15E-01)	3.07E+00(8.04E-01)	2.75E+00(5.84E-01)	3.00E+00(7.87E-01)	2.92E+00(5.31E-01)	2.93E+00(6.40E-01)	2.50E+00(2.62E-01)	**2.49E+00(2.39E-01)**
F16	9.36E+00(4.80E-01)	9.48E+00(6.95E-01)	9.29E+00(5.59E-01)	9.56E+00(7.57E-01)	9.33E+00(5.21E-01)	9.63E+00(6.91E-01)	**9.15E+00(4.90E-01)**	9.24E+00(5.55E-01)
F17	1.72E+02(1.18E+02)	**9.07E+01(6.73E+01)**	1.18E+02(7.33E+01)	8.78E+01(7.57E+01)	1.46E+02(1.05E+02)	1.36E+02(9.98E+01)	1.07E+02(6.16E+01)	1.36E+02(7.15E+01)
F18	1.02E+01(5.22E+00)	8.22E+00(3.23E+00)	7.70E+00(3.95E+00)	8.03E+00(2.77E+00)	8.04E+00(3.87E+00)	8.56E+00(4.61E+00)	**5.74E+00(1.98E+00)**	6.35E+00(2.56E+00)
F19	3.43E+00(6.05E-01)	**2.99E+00(4.54E-01)**	3.16E+00(5.05E-01)	3.05E+00(4.93E-01)	3.18E+00(5.91E-01)	3.33E+00(5.96E-01)	3.25E+00(5.64E-01)	3.25E+00(5.95E-01)
F20	4.12E+00(1.57E+00)	6.28E+00(2.04E+00)	5.17E+00(1.82E+00)	5.95E+00(1.84E+00)	4.62E+00(1.77E+00)	4.74E+00(1.89E+00)	4.07E+00(1.53E+00)	**3.86E+00(1.29E+00)**
F21	4.80E+01(5.22E+01)	5.57E+01(6.43E+01)	4.70E+01(5.30E+01)	6.27E+01(6.46E+01)	4.00E+01(5.06E+01)	3.92E+01(4.80E+01)	5.41E+01(5.47E+01)	**3.22E+01(4.47E+01)**
F22	5.12E+01(4.79E+01)	3.62E+01(3.13E+01)	3.83E+01(1.09E+01)	3.55E+01(2.97E+01)	4.48E+01(3.59E+01)	3.49E+01(2.82E+01)	**3.15E+01(1.77E+01)**	3.36E+01(1.71E+01)
F23	**3.06E+02(3.13E+01)**	3.15E+02(4.50E-13)	3.15E+02(9.22E-09)	3.15E+02(4.64E-08)	3.06E+02(3.13E+01)	**3.06E+02(3.13E+01)**	3.15E+02(4.02E-13)	**3.06E+02(3.13E+01)**
F24	**2.14E+02(1.16E+01)**	2.24E+02(9.59E-01)	2.24E+02(9.83E-01)	2.24E+02(9.45E-01)	**2.14E+02(1.15E+01)**	**2.14E+02(1.16E+01)**	2.24E+02(8.76E-01)	**2.14E+02(1.16E+01)**
F25	**2.02E+02(6.44E-01)**	2.03E+02(5.78E-02)	2.03E+02(1.14E-01)	2.03E+02(5.84E-02)	**2.02E+02(5.44E-01)**	2.03E+02(6.40E-01)	2.03E+02(8.63E-02)	**2.02E+02(6.39E-01)**
F26	**1.00E+02(2.16E-02)**	**1.00E+02(2.03E-02)**	**1.00E+02(2.81E-02)**	**1.00E+02(2.80E-02)**	**1.00E+02(2.58E-02)**	**1.00E+02(2.51E-02)**	**1.00E+02(1.42E-02)**	**1.00E+02(1.66E-02)**
F27	**3.00E+02(3.06E-02)**	**3.00E+02(2.40E-03)**	**3.00E+02(8.73E-01)**	**3.00E+02(1.48E-03)**	**3.00E+02(7.74E-01)**	**3.00E+02(1.13E+00)**	**3.00E+02(4.89E-04)**	**3.00E+02(2.67E-02)**
F28	8.09E+02(1.26E+02)	8.10E+02(2.39E+01)	8.34E+02(1.95E+01)	8.14E+02(2.59E+01)	**8.04E+02(1.26E+02)**	8.06E+02(1.26E+02)	8.26E+02(2.38E+01)	8.18E+02(9.20E+01)
F29	5.87E+02(2.15E+02)	**3.08E+02(2.31E+02)**	5.85E+02(2.28E+02)	2.98E+02(2.23E+02)	5.07E+02(2.50E+02)	4.30E+02(2.53E+02)	6.74E+02(1.46E+02)	6.85E+02(1.27E+02)
F30	5.01E+02(1.06E+02)	4.99E+02(1.21E+02)	5.05E+02(1.16E+02)	4.99E+02(1.18E+02)	4.97E+02(1.05E+02)	5.01E+02(1.15E+02)	5.00E+02(1.31E+02)	**4.87E+02(1.12E+02)**
F31	2.92E+03(5.88E+03)	2.42E+02(3.47E+02)	6.18E-09(2.91E-08)	2.33E+02(3.46E+02)	2.80E+03(6.40E+03)	1.67E+03(4.13E+03)	2.35E-07(2.28E-07)	**0.00E+00(0.00E+00)**
F32	1.92E+02(3.77E+02)	6.55E-02(3.05E-01)	**0.00E+00(0.00E+00)**	8.15E-02(3.13E-01)	1.59E+02(3.34E+02)	1.52E+02(4.02E+02)	4.27E-08(5.05E-08)	**0.00E+00(0.00E+00)**
F33	**2.02E+01(8.79E-02)**	**2.02E+01(8.40E-02)**	**2.02E+01(7.88E-02)**	2.03E+01(5.69E-02)	2.03E+01(6.34E-02)	2.03E+01(9.29E-02)	2.08E+01(1.42E-01)	**2.02E+01(7.42E-02)**
F34	6.62E+00(4.00E+00)	2.09E+01(6.29E+00)	7.82E+00(1.77E+00)	1.99E+01(6.05E+00)	1.04E+01(6.78E+00)	1.03E+01(7.61E+00)	7.61E+01(4.04E+01)	**4.70E+00(1.41E+00)**
F35	1.83E+03(4.95E+02)	1.97E+03(4.74E+02)	1.80E+03(5.95E+02)	2.07E+03(5.04E+02)	1.85E+03(5.24E+02)	1.82E+03(5.07E+02)	5.73E+03(6.85E+02)	**1.43E+03(2.46E+02)**
F36	1.47E+02(6.27E+01)	1.05E+02(7.36E+01)	1.08E+02(5.45E+01)	**1.01E+02(6.42E+01)**	1.41E+02(6.75E+01)	1.26E+02(6.59E+01)	6.88E+02(1.56E+02)	1.33E+02(6.55E+01)
F37	6.46E+00(7.69E-01)	**5.19E+00(8.52E-01)**	6.40E+00(5.16E-01)	5.65E+00(1.13E+00)	5.40E+00(1.13E+00)	5.87E+00(9.85E-01)	8.03E+00(3.30E-01)	6.43E+00(5.77E-01)
F38	3.23E+01(1.44E+01)	2.99E+01(1.48E+01)	**2.85E+01(1.45E+01)**	3.25E+01(1.84E+01)	3.12E+01(1.32E+01)	3.29E+01(1.82E+01)	3.69E+02(6.47E+01)	2.90E+01(1.39E+01)
F39	1.03E+02(1.85E-01)	1.03E+02(1.17E-01)	1.03E+02(1.03E-01)	1.03E+02(1.36E-01)	1.03E+02(1.75E-01)	1.03E+02(1.85E-01)	1.03E+02(8.90E-02)	**1.02E+02(1.05E-01)**
F40	4.02E+02(1.28E+02)	**3.50E+02(1.27E+02)**	3.74E+02(1.42E+02)	3.54E+02(1.24E+02)	3.92E+02(1.19E+02)	3.97E+02(1.32E+02)	6.98E+02(1.32E+02)	4.52E+02(1.23E+02)
F41	**3.95E+02(2.45E+01)**	3.96E+02(1.94E+01)	3.96E+02(1.95E+01)	3.96E+02(1.95E+01)	**3.95E+02(2.46E+01)**	**3.95E+02(2.47E+01)**	3.98E+02(1.39E+01)	4.01E+02(5.96E+00)
F42	**1.03E+02(2.80E-01)**	**1.03E+02(2.61E-01)**	**1.03E+02(1.87E-01)**	**1.03E+02(2.36E-01)**	**1.03E+02(3.77E-01)**	**1.03E+02(3.44E-01)**	1.04E+02(2.84E-01)	**1.03E+02(1.99E-01)**
F43	2.64E-02(9.19E-04)	**2.60E-02(3.94E-04)**	2.62E-02(4.99E-04)	2.64E-02(8.35E-04)	2.65E-02(9.98E-04)	2.66E-02(1.22E-03)	2.61E-02(4.63E-04)	2.64E-02(6.54E-04)
F44	**3.13E+04(1.63E+02)**	3.14E+04(4.64E+02)	3.15E+04(4.66E+02)	3.14E+04(4.72E+02)	**3.13E+04(1.26E+02)**	**3.13E+04(1.51E+02)**	3.14E+04(4.78E+02)	3.16E+04(6.91E+02)
F45	**1.00E+02(1.23E-13)**	**1.00E+02(1.39E-13)**	**1.00E+02(1.39E-13)**	**1.00E+02(2.23E-08)**	**1.00E+02(5.28E-05)**	**1.00E+02(8.21E-07)**	**1.00E+02(8.35E-12)**	**1.00E+02(1.23E-13)**

Table A.10: Mean error results and standard deviation achieved by the proposed algorithm and seven other heuristic- based selection methods for 50D

	AOSDE-LS	AOSDE-SR	AOSDE-FIR	AOSDE-PR	AOSDE-PBM	AOSDE-RD	AOSDE-PAOC	LSAOS-DE
	Mean(Std.)	Mean(Std.)	Mean(Std.)	Mean(Std.)	Mean(Std.)	Mean(Std.)	Mean(Std.)	Mean(Std.)
F01	1.00E+05(1.97E+05)	4.88E+03(3.72E+03)	3.31E+04(9.52E+04)	1.51E+05(7.23E+04)	1.30E+05(1.60E+05)	7.74E+04(1.23E+05)	4.88E+03(3.72E+03)	**1.23E+03(2.50E+03)**
F02	2.66E+02(9.38E+02)	7.89E-08(2.05E-07)	4.47E+02(1.18E+03)	3.24E+03(1.08E+03)	1.39E+03(1.67E+03)	2.24E+03(1.94E+03)	7.89E-08(2.05E-07)	**0.00E+00(0.00E+00)**
F03	5.06E-02(3.62E-01)	**0.00E+00(0.00E+00)**	9.28E-01(2.36E+00)	5.19E-02(4.63E-02)	8.74E-01(4.64E+00)	1.85E-03(5.33E-01)	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**
F04	5.12E+01(3.75E+01)	3.03E+01(3.28E+01)	3.97E+01(3.69E+01)	9.31E+01(1.30E+01)	9.04E+01(1.09E+01)	8.84E+01(7.32E+01)	**3.03E+01(3.28E+01)**	3.93E+01(3.93E+01)
F05	2.04E+01(1.30E-01)	2.04E+01(1.26E-01)	2.04E+01(1.56E-01)	2.04E+01(1.15E-01)	2.04E+01(1.18E-01)	2.04E+01(2.04E+01)	2.04E+01(1.17E-01)	2.04E+01(2.30E-03)
F06	1.18E+00(7.66E-01)	1.43E+00(8.83E-01)	1.91E+00(1.99E+00)	**2.52E-01(4.00E-01)**	1.57E+00(1.17E+00)	1.09E+00(1.88E+00)	1.43E+00(8.83E-01)	1.20E+00(7.82E-01)
F07	2.90E-03(4.56E-03)	2.27E-03(4.32E-03)	**6.22E-04(2.00E-03)**	2.42E-03(4.37E-03)	3.53E-03(4.62E-03)	3.27E-03(3.30E-03)	2.27E-03(4.32E-03)	3.09E-03(4.65E-03)
F08	2.07E-05(2.89E-05)	3.44E-05(2.80E-05)	2.77E-05(3.15E-05)	8.16E-06(5.16E-06)	1.27E-05(2.08E-05)	2.68E-06(9.75E-06)	6.78E-06(1.06E-05)	**0.00E+00(0.00E+00)**
F09	1.81E+01(3.78E+00)	1.87E+01(2.21E+00)	1.75E+01(3.84E+00)	3.65E+01(9.37E+00)	2.57E+01(1.06E+01)	1.99E+01(2.21E+01)	2.11E+01(5.24E+00)	**1.70E+01(2.96E+00)**
F10	1.29E+00(1.32E+00)	1.13E+00(1.13E+00)	1.46E+00(1.40E+00)	1.37E+00(1.27E+00)	1.26E+00(1.23E+00)	**5.55E-01(1.36E+00)**	6.48E-01(7.23E-01)	1.31E+00(1.49E+00)
F11	4.50E+03(8.20E+02)	4.06E+03(7.36E+02)	4.10E+03(7.28E+02)	4.73E+03(8.09E+02)	4.70E+03(7.99E+02)	4.46E+03(4.58E+03)	4.03E+03(6.50E+02)	**4.02E+03(7.88E+02)**
F12	3.44E-01(1.76E-01)	3.61E-01(1.62E-01)	3.39E-01(1.69E-01)	3.71E-01(1.71E-01)	3.30E-01(1.71E-01)	**3.17E-01(3.31E-01)**	3.45E-01(1.49E-01)	3.71E-01(2.27E-02)
F13	1.88E-01(2.95E-02)	1.98E-01(1.67E-02)	1.87E-01(1.73E-02)	1.97E-01(2.61E-02)	1.91E-01(3.02E-02)	**1.83E-01(1.80E-01)**	1.99E-01(1.75E-02)	1.97E-01(2.04E-02)
F14	2.94E-01(2.42E-02)	**2.90E-01(1.81E-02)**	2.99E-01(2.33E-02)	2.93E-01(2.23E-02)	2.97E-01(2.73E-02)	2.96E-01(2.96E-01)	**2.90E-01(1.85E-02)**	2.94E-01(1.88E-02)
F15	6.15E+00(6.62E-01)	6.13E+00(4.86E-01)	7.00E+00(8.72E-01)	5.61E+00(7.98E-01)	**5.59E+00(1.09E+00)**	5.91E+00(5.87E+00)	6.13E+00(4.86E-01)	6.08E+00(8.36E-01)
F16	1.83E+01(8.66E-01)	1.79E+01(7.43E-01)	1.82E+01(9.66E-01)	1.87E+01(8.72E-01)	1.84E+01(8.59E-01)	1.79E+01(1.81E+01)	**1.77E+01(6.86E-01)**	1.80E+01(9.65E-01)
F17	1.31E+03(4.48E+02)	8.83E+02(3.05E+02)	9.64E+02(4.17E+02)	**8.09E+02(3.08E+02)**	1.26E+03(4.62E+02)	1.18E+03(1.24E+03)	8.83E+02(3.05E+02)	1.04E+03(3.56E+02)
F18	9.08E+01(1.19E+01)	7.07E+01(1.42E+01)	6.96E+01(2.19E+01)	**3.38E+01(8.94E+00)**	8.45E+01(1.06E+01)	8.78E+01(8.66E+01)	7.07E+01(1.42E+01)	8.60E+01(1.06E+01)
F19	1.14E+01(2.89E+00)	1.07E+01(7.30E-01)	1.08E+01(5.91E-01)	1.02E+01(6.68E-01)	1.10E+01(3.01E+00)	1.09E+01(1.20E+01)	1.07E+01(7.30E-01)	**1.06E+01(8.51E-01)**
F20	1.50E+01(8.22E+00)	1.27E+01(3.75E+00)	2.11E+01(8.92E+00)	2.47E+01(7.74E+00)	2.06E+01(6.15E+00)	1.79E+01(1.89E+01)	1.27E+01(3.75E+00)	**1.18E+01(3.98E+00)**
F21	5.14E+02(1.77E+02)	**4.08E+02(1.00E+02)**	4.45E+02(1.29E+02)	4.23E+02(1.45E+02)	4.65E+02(1.58E+02)	4.89E+02(4.90E+02)	**4.08E+02(1.00E+02)**	4.15E+02(1.20E+02)
F22	1.83E+02(9.36E+01)	1.89E+02(8.16E+01)	1.60E+02(8.55E+01)	1.97E+02(1.05E+02)	1.72E+02(1.22E+02)	**1.53E+02(1.65E+02)**	1.96E+02(8.28E+01)	1.80E+02(8.48E+01)
F23	**3.41E+02(2.02E+01)**	3.44E+02(4.59E-13)	3.44E+02(9.25E-07)	3.44E+02(3.24E-06)	**3.41E+02(2.03E+01)**	3.44E+02(3.41E+02)	3.44E+02(4.59E-13)	**3.41E+02(2.03E+01)**
F24	2.55E+02(3.13E+01)	2.75E+02(7.74E-01)	2.75E+02(6.32E-01)	2.75E+02(7.39E-01)	2.61E+02(3.07E+01)	2.71E+02(2.52E+02)	2.75E+02(7.74E-01)	**2.52E+02(3.13E+01)**
F25	**2.04E+02(2.72E+00)**	2.05E+02(4.59E-01)	2.06E+02(5.14E-01)	2.05E+02(3.37E-01)	**2.04E+02(2.64E+00)**	2.05E+02(2.04E+02)	2.05E+02(4.59E-01)	**2.04E+02(2.62E+00)**
F26	**1.00E+02(2.13E-02)**	**1.00E+02(1.89E-02)**	**1.00E+02(2.04E-02)**	**1.00E+02(1.84E-02)**	**1.00E+02(2.60E-02)**	**1.00E+02(1.00E+02)**	**1.00E+02(1.89E-02)**	**1.00E+02(1.74E-02)**
F27	3.24E+02(2.49E+01)	3.26E+02(2.36E+01)	3.23E+02(2.36E+01)	**3.13E+02(2.22E+01)**	3.28E+02(3.18E+01)	**3.13E+02(3.27E+02)**	3.26E+02(2.36E+01)	3.24E+02(2.35E+01)
F28	1.12E+03(1.36E+02)	1.12E+03(3.02E+01)	1.14E+03(2.60E+01)	1.13E+03(3.79E+01)	1.12E+03(1.38E+02)	1.13E+03(1.11E+03)	1.12E+03(3.02E+01)	**1.11E+03(1.34E+02)**
F29	8.11E+02(6.78E+01)	7.82E+02(4.77E+01)	7.99E+02(5.63E+01)	**7.03E+02(9.55E+01)**	7.83E+02(5.99E+01)	7.81E+02(7.95E+02)	7.82E+02(4.77E+01)	7.91E+02(4.87E+01)
F30	8.75E+03(4.76E+02)	8.47E+03(3.86E+02)	8.53E+03(4.61E+02)	8.53E+03(4.26E+02)	8.68E+03(5.55E+02)	8.73E+03(8.72E+03)	**8.47E+03(3.86E+02)**	8.57E+03(4.19E+02)
F31	7.00E+04(1.52E+05)	3.19E+03(3.50E+03)	3.19E+03(3.50E+03)	1.09E+05(4.51E+04)	1.48E+05(2.01E+05)	2.42E+04(7.28E+04)	**4.79E+02(5.54E+02)**	1.74E+03(1.28E+03)
F32	3.48E+02(1.09E+03)	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	3.42E+03(1.43E+03)	1.81E+03(2.05E+03)	1.98E+03(1.90E+03)	1.16E+00(1.26E+00)	**0.00E+00(0.00E+00)**
F33	**2.04E+01(1.32E-01)**	**2.04E+01(1.48E-01)**	**2.04E+01(1.74E-01)**	**2.04E+01(1.47E-01)**	**2.04E+01(1.23E-01)**	**2.04E+01(2.04E+01)**	2.11E+01(7.63E-02)	**2.04E+01(6.02E-04)**
F34	2.10E+01(4.57E+00)	2.12E+01(2.97E+00)	2.12E+01(2.97E+00)	3.52E+01(1.01E+01)	3.03E+01(1.06E+01)	2.38E+01(2.68E+01)	1.53E+02(1.02E+02)	**1.79E+01(3.43E+00)**
F35	4.01E+03(7.01E+02)	3.79E+03(6.76E+02)	4.07E+03(8.48E+02)	4.58E+03(8.55E+02)	4.20E+03(8.91E+02)	4.06E+03(4.20E+03)	1.25E+04(7.66E+02)	**3.51E+03(6.24E+02)**
F36	1.49E+03(4.43E+02)	**1.07E+03(3.78E+02)**	**1.07E+03(3.78E+02)**	**1.07E+03(3.56E+02)**	1.24E+03(4.37E+02)	1.25E+03(1.29E+03)	1.10E+03(3.85E+02)	**1.07E+03(2.84E+02)**
F37	4.10E+01(2.36E+00)	4.00E+01(7.30E-01)	4.00E+01(7.30E-01)	**3.94E+01(7.30E-01)**	**3.94E+01(8.08E-01)**	4.04E+01(4.08E+01)	4.21E+01(2.10E+00)	4.03E+01(2.78E+00)
F38	2.13E+02(1.55E+02)	1.72E+02(1.15E+02)	1.72E+02(1.15E+02)	1.88E+02(1.45E+02)	2.12E+02(1.82E+02)	**1.63E+02(2.35E+02)**	8.70E+02(2.26E+02)	1.75E+02(8.28E+01)
F39	**1.04E+02(2.39E-01)**	**1.04E+02(1.23E-01)**	**1.04E+02(1.23E-01)**	**1.04E+02(1.87E-01)**	**1.04E+02(2.74E-01)**	**1.04E+02(1.04E+02)**	1.05E+02(1.38E-01)	**1.04E+02(1.62E-01)**
F40	1.00E+03(1.81E+02)	**8.79E+02(1.04E+02)**	**8.79E+02(1.04E+02)**	9.30E+02(1.40E+02)	9.22E+02(1.75E+02)	9.08E+02(9.57E+02)	1.30E+03(5.78E+02)	9.02E+02(9.65E+01)
F41	4.37E+02(2.34E+01)	4.39E+02(2.02E+01)	4.39E+02(2.02E+01)	**4.21E+02(1.88E+01)**	4.34E+02(2.42E+01)	4.39E+02(4.42E+02)	4.26E+02(2.02E+01)	4.26E+02(2.15E+01)
F42	**1.05E+02(2.59E-01)**	**1.05E+02(3.58E-01)**	**1.05E+02(3.58E-01)**	**1.05E+02(3.87E-01)**	**1.05E+02(4.51E-01)**	**1.05E+02(1.05E+02)**	1.07E+02(4.09E-01)	1.07E+02(2.80E-01)
F43	8.40E-02(5.20E-03)	8.46E-02(4.15E-03)	8.46E-02(4.15E-03)	8.58E-02(7.17E-03)	8.46E-02(7.29E-03)	8.41E-02(8.48E-02)	**8.18E-02(3.86E-03)**	8.41E-02(4.33E-03)
F44	5.98E+04(4.56E+03)	6.05E+04(5.09E+03)	6.00E+04(5.09E+03)	5.99E+04(4.95E+03)	5.99E+04(4.91E+03)	**5.90E+04(5.98E+04)**	5.96E+04(5.08E+03)	6.16E+04(5.39E+03)
F45	**1.00E+02(1.57E-13)**	**1.00E+02(1.25E-13)**	**1.00E+02(1.25E-13)**	**1.00E+02(2.83E-10)**	**1.00E+02(8.11E-02)**	**1.00E+02(1.32E-04)**	1.00E+02(9.80E-08)	**1.00E+02(1.65E-13)**

Table A.11: Detailed results for reduction mechanism

	Ver1		Ver2		Ver3		Ver4		Ver5	
	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
F01	2.11E-07	8.39E-07	1.07E-05	3.21E-05	7.95E-04	2.70E-03	3.11E-06	1.83E-05	**0.00E+00**	**0.00E+00**
F02	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F03	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F04	5.40E-06	3.69E-05	5.72E-06	1.76E-05	1.90E-05	1.02E-04	3.00E-05	2.12E-04	**0.00E+00**	**0.00E+00**
F05	2.02E+01	8.73E-02	2.02E+01	8.73E-02	2.02E+01	8.23E-02	**2.01E+01**	**2.91E-02**	2.02E+01	5.44E-02
F06	**1.96E-02**	**7.18E-02**	8.04E-02	2.12E-01	1.26E-01	3.04E-01	1.59E-01	3.95E-01	1.80E-01	1.28E+00
F07	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F08	**0.00E+00**	3.71E-08	6.43E-08	3.22E-07	1.36E-08	9.48E-08	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F09	1.49E+01	6.31E+00	1.46E+01	6.12E+00	1.50E+01	6.93E+00	1.23E+01	4.93E+00	**1.19E+01**	**7.15E+00**
F10	8.16E-04	4.08E-03	**4.08E-04**	**2.92E-03**	1.63E-03	5.65E-03	8.16E-04	4.08E-03	2.45E-03	6.77E-03
F11	1.57E+03	3.88E+02	1.64E+03	4.28E+02	1.65E+03	4.63E+02	**1.37E+03**	**3.24E+02**	1.61E+03	4.91E+02
F12	2.70E-01	1.13E-01	2.76E-01	1.12E-01	2.66E-01	9.74E-02	**1.75E-01**	**2.46E-02**	2.26E-01	4.44E-02
F13	1.07E-01	2.64E-02	1.09E-01	2.76E-02	1.08E-01	3.03E-02	1.17E-01	1.92E-02	**8.12E-02**	**2.58E-02**
F14	2.21E-01	2.75E-02	2.33E-01	2.59E-02	**2.16E-01**	**2.37E-02**	2.17E-01	2.12E-02	2.44E-01	2.83E-02
F15	2.91E+00	7.28E-01	2.85E+00	6.49E-01	2.79E+00	6.00E-01	**2.77E+00**	**5.74E-01**	2.94E+00	7.56E-01
F16	9.02E+00	5.15E-01	8.95E+00	5.92E-01	9.02E+00	6.36E-01	**8.72E+00**	**3.69E-01**	9.16E+00	8.39E-01
F17	1.39E+02	7.97E+01	1.80E+02	1.04E+02	2.13E+02	1.10E+02	**1.37E+02**	**7.41E+01**	1.73E+02	9.88E+01
F18	7.86E+00	3.30E+00	8.88E+00	4.19E+00	9.87E+00	4.35E+00	8.33E+00	3.74E+00	**6.94E+00**	**2.87E+00**
F19	3.05E+00	5.61E-01	3.32E+00	6.37E-01	3.23E+00	7.27E-01	3.34E+00	6.17E-01	**2.70E+00**	**6.18E-01**
F20	4.50E+00	1.28E+00	5.28E+00	1.74E+00	5.23E+00	1.79E+00	4.86E+00	1.84E+00	**3.64E+00**	**1.34E+00**
F21	**5.03E+01**	**5.97E+01**	6.39E+01	6.45E+01	9.24E+01	8.30E+01	6.79E+01	8.01E+01	6.96E+01	7.06E+01
F22	**2.77E+01**	**1.80E+01**	3.71E+01	2.89E+01	3.61E+01	3.19E+01	3.57E+01	3.32E+01	2.83E+01	1.92E+01
F23	**3.11E+02**	**2.26E+01**	**3.11E+02**	**2.26E+01**	3.12E+02	1.71E+01	**3.11E+02**	**2.26E+01**	3.15E+02	3.73E-13
F24	**2.18E+02**	**1.02E+01**	2.19E+02	9.51E+00	2.19E+02	9.81E+00	2.18E+02	1.05E+01	2.22E+02	6.46E+00
F25	**2.03E+02**	**4.06E-01**	2.03E+02	5.26E-01	2.03E+02	5.34E-01	2.03E+02	5.32E-01	2.03E+02	5.20E-02
F26	**1.00E+02**	**2.24E-02**	**1.00E+02**	**2.35E-02**	**1.00E+02**	**3.30E-02**	**1.00E+02**	**1.58E-02**	**1.00E+02**	**2.04E-02**
F27	**3.00E+02**	**1.08E-01**	3.02E+02	1.41E+01	**3.00E+02**	**4.86E+01**	**3.00E+02**	**6.33E-02**	**3.00E+02**	**1.59E-03**
F28	8.06E+02	1.26E+02	8.09E+02	1.26E+02	**8.04E+02**	**1.25E+02**	8.10E+02	1.27E+02	8.31E+02	1.98E+01
F29	**6.63E+02**	**1.64E+02**	7.06E+02	7.56E+01	6.87E+02	1.20E+02	7.06E+02	7.79E+01	7.18E+02	5.36E+00
F30	5.69E+02	1.74E+02	5.72E+02	1.81E+02	**5.18E+02**	**1.54E+02**	5.84E+02	2.11E+02	7.68E+02	2.61E+02

Table A.12: Detailed results for NP^{init}

	$NP^{init} = 5D$		$NP^{init} = 10D$		$NP^{init} = 15D$		$NP^{init} = 18D$		$NP^{init} = 20D$		$NP^{init} = 25D$		$NP^{init} = 30D$	
	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
F01	2.34E+03	3.12E+03	2.38E-01	9.20E-01	1.76E-05	4.83E-05	4.62E-07	1.68E-06	2.55E-07	1.07E-06	8.98E-09	2.63E-08	**4.71E-09**	**1.20E-08**
F02	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F03	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F04	9.20E-05	4.15E-04	**3.10E-08**	**1.22E-07**	1.91E-07	7.48E-07	1.28E-07	4.18E-07	2.72E-07	1.38E-06	7.19E-05	3.06E-04	6.04E-03	2.44E-02
F05	**2.01E+01**	**6.72E-02**	**2.01E+01**	**9.49E-02**	2.02E+01	9.55E-02	2.02E+01	8.66E-02	2.03E+01	8.92E-02	2.03E+01	8.29E-02	2.03E+01	8.78E-02
F06	3.19E+00	3.10E+00	1.41E+00	2.73E+00	9.52E-01	2.80E+00	4.88E-02	1.40E-01	6.76E-02	1.82E-01	4.49E-02	1.65E-01	**7.20E-03**	**4.85E-02**
F07	1.45E-03	4.24E-03	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F08	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.95E-09	1.39E-08	**0.00E+00**	**0.00E+00**	4.69E-09	3.35E-08	**0.00E+00**	**0.00E+00**	7.51E-09	1.69E-08
F09	1.43E+01	5.91E+00	1.12E+01	5.28E+00	9.06E+00	4.42E+00	7.29E+00	4.35E+00	8.29E+00	4.86E+00	4.69E+00	1.62E+00	**4.62E+00**	**1.54E+00**
F10	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.22E-03	4.95E-03	9.18E-03	1.59E-02	2.91E-02	3.67E-02
F11	**1.46E+03**	**4.17E+02**	1.49E+03	5.21E+02	1.48E+03	4.09E+02	1.49E+03	3.68E+02	1.59E+03	4.00E+02	1.60E+03	4.42E+02	1.68E+03	4.93E+02
F12	**1.04E-01**	**2.49E-02**	1.63E-01	7.32E-02	2.16E-01	9.70E-02	2.32E-01	1.16E-01	2.81E-01	1.18E-01	3.02E-01	1.23E-01	3.69E-01	1.13E-01
F13	1.51E-01	2.66E-02	1.27E-01	2.38E-02	1.24E-01	2.45E-02	1.21E-01	2.12E-02	**1.16E-01**	**2.45E-02**	1.22E-01	1.75E-02	1.26E-01	1.47E-02
F14	2.44E-01	3.41E-02	2.31E-01	2.54E-02	2.21E-01	2.50E-02	2.29E-01	2.36E-02	**2.16E-01**	**2.64E-02**	2.18E-01	2.38E-02	**2.16E-01**	**2.24E-02**
F15	2.58E+00	8.34E-01	2.70E+00	6.55E-01	2.74E+00	5.81E-01	2.83E+00	6.89E-01	2.65E+00	5.56E-01	**2.49E+00**	**2.39E-01**	2.66E+00	2.52E-01
F16	9.07E+00	8.16E-01	8.88E+00	5.78E-01	**8.71E+00**	**4.37E-01**	8.95E+00	6.33E-01	8.89E+00	5.22E-01	9.24E+00	5.55E-01	9.27E+00	5.38E-01
F17	4.65E+02	2.26E+02	2.25E+02	1.27E+02	1.58E+02	9.46E+01	1.38E+02	6.91E+01	**1.08E+02**	**7.09E+01**	1.36E+02	7.15E+01	1.21E+02	6.68E+01
F18	2.80E+01	1.24E+01	1.26E+01	6.67E+00	8.14E+00	3.54E+00	8.29E+00	2.89E+00	6.83E+00	2.97E+00	6.35E+00	2.56E+00	**6.15E+00**	**2.17E+00**
F19	3.38E+00	5.49E-01	**2.89E+00**	**5.43E-01**	2.98E+00	6.86E-01	3.09E+00	5.79E-01	3.32E+00	5.73E-01	3.25E+00	5.95E-01	3.20E+00	5.89E-01
F20	1.01E+01	4.51E+00	5.48E+00	1.91E+00	4.00E+00	1.25E+00	4.46E+00	1.25E+00	4.15E+00	1.41E+00	**3.86E+00**	**1.29E+00**	4.06E+00	1.25E+00
F21	1.56E+02	8.47E+01	8.79E+01	7.41E+01	6.71E+01	6.40E+01	3.81E+01	5.52E+01	4.31E+01	5.80E+01	**3.22E+01**	**4.47E+01**	3.60E+01	4.55E+01
F22	8.54E+01	5.91E+01	4.47E+01	3.65E+01	3.57E+01	2.94E+01	3.48E+01	2.81E+01	**2.98E+01**	**1.11E+01**	3.36E+01	1.71E+01	3.35E+01	1.06E+01
F23	3.15E+02	4.02E-13	3.15E+02	4.02E-13	3.15E+02	4.16E-13	3.11E+02	2.26E+01	3.15E+02	4.02E-13	**3.06E+02**	**3.13E+01**	3.13E+02	1.61E+01
F24	2.27E+02	4.66E+00	2.25E+02	3.00E+00	2.22E+02	7.23E+00	2.17E+02	1.08E+01	2.18E+02	1.04E+01	**2.14E+02**	**1.16E+01**	2.15E+02	1.14E+01
F25	**2.03E+02**	**5.43E-01**	**2.03E+02**	**3.11E-01**	**2.03E+02**	**8.64E-02**	**2.03E+02**	**4.46E-01**	**2.03E+02**	**6.81E-02**	**2.02E+02**	**6.39E-01**	**2.03E+02**	**3.69E-01**
F26	**1.00E+02**	**2.52E-02**	**1.00E+02**	**2.44E-02**	**1.00E+02**	**2.50E-02**	**1.00E+02**	**2.34E-02**	**1.00E+02**	**2.20E-02**	**1.00E+02**	**1.66E-02**	**1.00E+02**	**1.50E-02**
F27	3.22E+02	1.69E+01	3.08E+02	2.37E+01	3.03E+02	1.43E+01	**2.95E+02**	**4.16E+01**	3.00E+02	3.85E-02	3.00E+02	2.67E-02	3.00E+02	1.08E-03
F28	8.53E+02	3.05E+01	8.30E+02	2.24E+01	8.25E+02	2.14E+01	**8.07E+02**	**9.49E+01**	8.34E+02	2.40E+01	8.18E+02	9.20E+01	8.25E+02	3.45E+01
F29	6.82E+02	1.24E+02	6.97E+02	1.05E+02	6.81E+02	1.32E+02	**6.63E+02**	**1.60E+02**	6.95E+02	1.06E+02	6.85E+02	1.27E+02	7.06E+02	7.68E+01
F30	7.12E+02	2.27E+02	6.48E+02	2.78E+02	5.75E+02	2.01E+02	5.06E+02	1.39E+02	5.29E+02	1.56E+02	4.87E+02	1.12E+02	**4.85E+02**	**1.43E+02**

Table A.13: Detailed results for CS

	$CS = 25$		$CS = 50$		$CS = 75$		$CS = 100$		$CS = 150$		$CS = 200$	
	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
F01	2.11E-07	8.39E-07	2.98E-06	1.73E-05	5.76E-07	1.91E-06	2.87E-07	8.77E-07	4.62E-07	1.68E-06	**2.00E-07**	**4.15E-07**
F02	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F03	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F04	5.40E-06	3.69E-05	4.31E-08	2.64E-07	6.53E-04	4.66E-03	**2.97E-08**	**1.35E-07**	1.28E-07	4.18E-07	1.22E-07	3.84E-07
F05	**2.02E+01**	**8.73E-02**	**2.02E+01**	**9.46E-02**	**2.02E+01**	**8.20E-02**	**2.02E+01**	**8.48E-02**	**2.02E+01**	**8.66E-02**	**2.02E+01**	**8.33E-02**
F06	**1.96E-02**	**7.18E-02**	3.99E-01	1.62E+00	7.08E-02	2.65E-01	2.07E-01	1.34E+00	4.88E-02	1.40E-01	6.00E-02	1.79E-01
F07	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F08	**0.00E+00**	**3.71E-08**	2.10E-08	1.18E-07	1.53E-09	1.09E-08	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F09	1.49E+01	6.31E+00	1.50E+01	7.15E+00	1.48E+01	5.53E+00	1.16E+01	5.66E+00	**7.29E+00**	**4.35E+00**	1.03E+01	5.65E+00
F10	8.16E-04	4.08E-03	8.16E-04	4.08E-03	8.16E-04	4.08E-03	4.08E-04	2.92E-03	**0.00E+00**	**0.00E+00**	1.63E-03	5.65E-03
F11	1.57E+03	3.88E+02	1.63E+03	4.66E+02	1.60E+03	4.65E+02	1.51E+03	4.22E+02	1.49E+03	3.68E+02	**1.41E+03**	**3.83E+02**
F12	2.70E-01	1.13E-01	2.69E-01	1.03E-01	2.38E-01	1.09E-01	2.96E-01	1.13E-01	**2.32E-01**	**1.16E-01**	2.84E-01	1.16E-01
F13	1.07E-01	2.64E-02	1.09E-01	2.27E-02	**9.97E-02**	**2.61E-02**	1.08E-01	2.35E-02	1.21E-01	2.12E-02	1.18E-01	1.77E-02
F14	**2.21E-01**	**2.75E-02**	2.35E-01	2.45E-02	2.30E-01	2.55E-02	2.31E-01	2.23E-02	2.29E-01	2.36E-02	2.23E-01	2.43E-02
F15	2.91E+00	7.28E-01	2.87E+00	5.75E-01	2.92E+00	5.83E-01	2.88E+00	6.14E-01	2.83E+00	6.89E-01	**2.29E+00**	**2.61E-01**
F16	9.02E+00	5.15E-01	**8.86E+00**	**5.51E-01**	9.01E+00	6.13E-01	8.87E+00	4.13E-01	8.95E+00	6.33E-01	8.96E+00	4.52E-01
F17	1.39E+02	7.97E+01	**1.35E+02**	**6.84E+01**	1.55E+02	7.23E+01	1.39E+02	9.64E+01	1.38E+02	6.91E+01	1.49E+02	8.71E+01
F18	7.86E+00	3.30E+00	8.24E+00	3.86E+00	**6.53E+00**	**2.56E+00**	7.77E+00	3.05E+00	8.29E+00	2.89E+00	6.75E+00	3.20E+00
F19	**3.05E+00**	**5.61E-01**	3.25E+00	6.88E-01	3.23E+00	4.63E-01	3.25E+00	6.33E-01	3.09E+00	5.79E-01	3.26E+00	5.67E-01
F20	4.50E+00	1.28E+00	**4.35E+00**	**1.19E+00**	4.71E+00	1.35E+00	4.42E+00	1.47E+00	4.46E+00	1.25E+00	4.61E+00	1.60E+00
F21	5.03E+01	5.97E+01	4.59E+01	5.69E+01	4.96E+01	6.05E+01	5.98E+01	6.84E+01	**3.81E+01**	**5.52E+01**	4.16E+01	6.27E+01
F22	2.77E+01	1.80E+01	2.82E+01	1.89E+01	**2.62E+01**	**1.73E+01**	2.82E+01	1.81E+01	3.48E+01	2.81E+01	2.83E+01	1.73E+01
F23	**3.11E+02**	**2.26E+01**	**3.11E+02**	**2.26E+01**	**3.11E+02**	**2.26E+01**	**3.11E+02**	**2.26E+01**	**3.11E+02**	**2.26E+01**	**3.11E+02**	**2.26E+01**
F24	2.18E+02	1.02E+01	**2.16E+02**	**1.11E+01**	2.17E+02	1.10E+01	2.17E+02	1.07E+01	2.17E+02	1.08E+01	2.17E+02	1.08E+01
F25	**2.03E+02**	**4.06E-01**	**2.03E+02**	**5.21E-01**	**2.03E+02**	**5.22E-01**	**2.03E+02**	**5.65E-01**	**2.03E+02**	**4.46E-01**	**2.03E+02**	**4.42E-01**
F26	**1.00E+02**	**2.24E-02**	**1.00E+02**	**2.27E-02**	**1.00E+02**	**2.58E-02**	**1.00E+02**	**2.00E-02**	**1.00E+02**	**2.34E-02**	**1.00E+02**	**1.65E-02**
F27	3.00E+02	1.08E-01	3.00E+02	9.15E-01	3.00E+02	3.69E-02	**2.94E+02**	**4.15E+01**	2.95E+02	4.16E+01	2.95E+02	4.15E+01
F28	**8.06E+02**	**1.26E+02**	8.15E+02	9.51E+01	8.15E+02	9.62E+01	8.14E+02	9.58E+01	8.07E+02	9.49E+01	8.14E+02	9.59E+01
F29	6.63E+02	1.64E+02	6.79E+02	1.38E+02	7.07E+02	7.78E+01	6.74E+02	1.51E+02	6.63E+02	1.60E+02	**6.53E+02**	**1.79E+02**
F30	5.69E+02	1.74E+02	5.64E+02	1.77E+02	5.78E+02	2.09E+02	5.41E+02	1.65E+02	5.06E+02	1.39E+02	**4.96E+02**	**1.45E+02**

Table A.14: Detailed results for NP^{min}

	$NP^{min} = 7$		$NP^{min} = 10$		$NP^{min} = 20$		$NP^{min} = 30$		$NP^{min} = 40$		$NP^{min} = 50$	
	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
F01	**0.00E+00**	**0.00E+00**	3.20E-09	9.70E-09	1.49E-08	1.04E-07	9.71E-10	4.95E-09	5.90E-09	2.84E-08	2.40E-10	1.71E-09
F02	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F03	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F04	7.19E-05	3.06E-04	1.20E-04	8.26E-04	9.64E-06	6.33E-05	9.54E-07	6.02E-06	5.16E-06	2.90E-05	**9.41E-07**	**4.25E-06**
F05	**2.02E+01**	**7.03E-02**	2.03E+01	7.34E-02	2.03E+01	8.13E-02	2.03E+01	1.03E-01	2.03E+01	6.17E-02	2.03E+01	5.85E-02
F06	4.49E-02	1.65E-01	2.68E-02	8.76E-02	5.67E-02	1.79E-01	8.44E-02	3.26E-01	2.85E-01	1.73E+00	**7.92E-03**	**3.58E-02**
F07	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F08	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.73E-08	1.22E-07	8.14E-07	5.77E-06	2.11E-06	5.04E-06	4.96E-05	1.15E-04
F09	**4.69E+00**	**1.62E+00**	4.94E+00	1.89E+00	4.99E+00	1.91E+00	6.84E+00	2.45E+00	8.14E+00	3.01E+00	9.80E+00	3.63E+00
F10	**6.03E-03**	**1.27E-02**	9.59E-03	1.96E-02	1.54E-01	1.43E-01	9.37E-01	9.45E-01	2.87E+00	2.06E+00	6.62E+00	3.48E+00
F11	1.63E+03	4.17E+02	**1.54E+03**	**3.70E+02**	1.82E+03	3.99E+02	1.95E+03	4.39E+02	2.02E+03	4.05E+02	2.24E+03	3.48E+02
F12	**3.40E-01**	**1.26E-01**	3.43E-01	1.12E-01	3.47E-01	1.23E-01	3.91E-01	1.09E-01	4.02E-01	1.17E-01	4.60E-01	1.02E-01
F13	**1.22E-01**	**1.75E-02**	1.27E-01	1.68E-02	1.28E-01	1.51E-02	1.25E-01	1.45E-02	1.29E-01	1.52E-02	1.33E-01	1.76E-02
F14	**2.18E-01**	**2.38E-02**	2.25E-01	2.52E-02	**2.18E-01**	**2.68E-02**	2.26E-01	2.03E-02	2.26E-01	1.80E-02	2.23E-01	2.47E-02
F15	**2.49E+00**	**2.39E-01**	2.53E+00	2.65E-01	2.78E+00	2.87E-01	3.05E+00	2.74E-01	3.30E+00	3.09E-01	3.59E+00	2.95E-01
F16	**9.24E+00**	**5.05E-01**	9.32E+00	5.25E-01	9.26E+00	3.67E-01	9.44E+00	4.03E-01	9.44E+00	4.00E-01	9.60E+00	4.07E-01
F17	1.36E+02	7.15E+01	**1.02E+02**	**5.71E+01**	1.41E+02	7.86E+01	1.11E+02	5.91E+01	1.19E+02	5.88E+01	1.26E+02	7.45E+01
F18	6.35E+00	2.56E+00	**5.56E+00**	**2.16E+00**	5.96E+00	1.59E+00	5.77E+00	2.49E+00	6.32E+00	2.69E+00	5.88E+00	2.47E+00
F19	**3.25E+00**	**5.95E-01**	3.48E+00	6.40E-01	3.26E+00	5.73E-01	3.31E+00	5.44E-01	3.51E+00	5.39E-01	3.35E+00	5.61E-01
F20	3.86E+00	1.29E+00	3.76E+00	1.50E+00	4.00E+00	1.21E+00	**3.75E+00**	**1.53E+00**	4.09E+00	1.36E+00	3.83E+00	1.18E+00
F21	**3.22E+01**	**4.47E+01**	5.07E+01	5.65E+01	4.46E+01	5.54E+01	3.53E+01	5.04E+01	5.55E+01	6.69E+01	5.05E+01	6.55E+01
F22	**3.36E+01**	**1.71E+01**	3.76E+01	2.87E+01	4.02E+01	1.91E+01	4.56E+01	2.66E+01	5.28E+01	2.40E+01	6.26E+01	3.28E+01
F23	**3.06E+02**	**3.13E+01**	**3.06E+02**	**3.13E+01**	3.10E+02	2.29E+01	**3.06E+02**	**3.13E+01**	3.08E+02	2.74E+01	3.08E+02	2.74E+01
F24	2.14E+02	1.16E+01	**2.13E+02**	**1.19E+01**	2.14E+02	1.16E+01	2.15E+02	1.13E+01	2.15E+02	1.13E+01	2.16E+02	1.11E+01
F25	**2.02E+02**	**6.39E-01**	**2.02E+02**	**6.29E-01**	2.02E+02	6.28E-01	2.03E+02	4.14E-01	**2.02E+02**	**5.38E-01**	**2.02E+02**	**6.24E-01**
F26	**1.00E+02**	**1.66E-02**	**1.00E+02**	**1.55E-02**	**1.00E+02**	**1.63E-02**	**1.00E+02**	**1.74E-02**	**1.00E+02**	**1.53E-02**	**1.00E+02**	**1.35E-02**
F27	**3.00E+02**	**2.67E-02**	**3.00E+02**	**5.85E-04**	**3.00E+02**	**2.42E-03**	**3.00E+02**	**3.25E-03**	**3.00E+02**	**2.09E-01**	3.02E+02	1.41E+01
F28	8.18E+02	9.20E+01	**8.04E+02**	**1.25E+02**	8.15E+02	9.09E+01	8.18E+02	9.47E+01	8.05E+02	1.26E+02	8.12E+02	9.01E+01
F29	6.85E+02	1.27E+02	6.86E+02	1.28E+02	6.94E+02	1.11E+02	6.89E+02	1.12E+02	**6.82E+02**	**1.27E+02**	7.17E+02	2.89E+00
F30	4.87E+02	1.12E+02	4.60E+02	8.17E+01	4.88E+02	1.38E+02	4.95E+02	1.32E+02	4.84E+02	1.17E+02	**4.73E+02**	**1.13E+02**

Appendix B

Table B.1: Function values obtained by different variants of the proposed algorithm for adapting F and Cr

problems	V1			V2			V3		
	Best	**Average**	**Std.**	**Best**	**Average**	**Std.**	**Best**	**Average**	**Std.**
g01	-1.50000E+01	-1.50000E+01	0.00000E+00	-1.50000E+01	-1.50000E+01	0.00000E+00	-1.50000E+01	-1.50000E+01	0.00000E+00
g02	-8.02972E-01	-8.01869E-01	6.09504E-04	-8.03616E-01	-8.03601E-01	1.47832E-05	-8.03619E-01	-8.02380E-01	3.44185E-03
g03	-5.13226E-01	-2.18730E-01	9.29892E-02	-8.61860E-01	-7.14512E-01	9.59075E-02	-1.00050E+00	-1.00050E+00	1.28598E-15
g04	-3.06655E+04	-3.06655E+04	3.22138E-05	-3.06655E+04	-3.06655E+04	3.63798E-12	-3.06655E+04	-3.06655E+04	3.71300E-12
g05	5.12503E+03	5.17432E+03	6.21819E+01	5.12650E+03	5.12688E+03	1.62354E+00	5.12650E+03	5.12650E+03	2.78475E-12
g06	-6.96181E+03	-6.96181E+03	5.34838E-10	-6.96181E+03	-6.96181E+03	0.00000E+00	-6.96181E+03	-6.96181E+03	0.00000E+00
g07	2.45634E+01	2.47502E+01	1.39015E-01	2.43112E+01	2.43209E+01	8.53217E-03	2.43062E+01	2.43062E+01	7.67474E-15
g08	-9.58250E-02	-9.58250E-02	1.41640E-17	-9.58250E-02	-9.58250E-02	1.41640E-17	-9.58250E-02	-9.58250E-02	1.41640E-17
g09	6.80900E+02	6.81047E+02	6.37421E-02	6.80630E+02	6.80630E+02	7.26636E-05	6.80630E+02	6.80630E+02	3.28186E-13
g10	7.12902E+03	7.18003E+03	3.11182E+01	7.05062E+03	7.05873E+03	6.57210E+00	7.04925E+03	7.04925E+03	3.50283E-12
g11	7.49900E-01	7.52264E-01	7.06428E-03	7.49900E-01	7.49900E-01	1.13312E-16	7.49900E-01	7.49900E-01	1.13312E-16
g12	-1.00000E+00	-1.00000E+00	0.00000E+00	-1.00000E+00	-1.00000E+00	0.00000E+00	-1.00000E+00	-1.00000E+00	0.00000E+00
g13	7.84537E-02	8.80558E-01	2.21669E-01	9.45200E-01	9.96542E-01	1.12006E-02	5.69331E-02	4.89065E-01	2.73842E-01
g14	-4.58500E+01	-4.30584E+01	1.78232E+00	-4.61202E+01	-4.42629E+01	1.12208E+00	-4.77649E+01	-4.77649E+01	2.52051E-14
g15	9.61715E+02	9.63626E+02	2.30394E+00	9.61715E+02	9.61715E+02	1.25106E-04	9.61715E+02	9.61715E+02	5.80156E-13
g16	-1.90507E+00	-1.90424E+00	7.72471E-04	-1.90516E+00	-1.90516E+00	5.20741E-16	-1.90516E+00	-1.90516E+00	4.53247E-16
g17	8.88292E+03	9.00496E+03	9.81875E+01	8.85927E+03	8.92641E+03	3.96003E+01	8.85353E+03	8.88700E+03	4.05872E+01
g18	-8.62543E-01	-8.55529E-01	5.28642E-03	-8.65629E-01	-7.58089E-01	9.63742E-02	-8.66025E-01	-8.50742E-01	5.28977E-02
g19	6.07134E+01	8.19026E+01	9.58561E+00	3.27738E+01	3.31460E+01	2.67972E-01	3.26556E+01	3.26556E+01	3.36523E-10
g21	3.00272E+02	5.99803E+02	2.31587E+02	1.93772E+02	2.69812E+02	5.00690E+01	1.93725E+02	2.06711E+02	3.76525E+01
g23	-2.59075E+02	6.07187E+01	1.89161E+02	-5.36082E+02	-1.81906E+02	2.09324E+02	-4.00055E+02	-4.00016E+02	1.32949E-01
g24	-5.50801E+00	-5.50801E+00	9.06493E-16	-5.50801E+00	-5.50801E+00	9.06493E-16	-5.50801E+00	-5.50801E+00	9.06493E-16

Table B.2: Function values obtained by proposed MLS-MODE with different CS values of CS= 25, 50 and 75 generations

problems	CS=**25**			CS=**50**			CS=**75**		
	Best	**Mean**	**Std.**	**Best**	**Mean**	**Std.**	**Best**	**Mean**	**Std.**
g01	-1.500E+01	-1.500E+01	0.000E+00	-1.500E+01	-1.500E+01	0.000E+00	-1.500E+01	-1.500E+01	0.000E+00
g02	-8.036E-01	-8.024E-01	3.442E-03	-8.036E-01	-8.032E-01	2.202E-03	-8.036E-01	-8.032E-01	2.202E-03
g03	-1.001E+00	-1.001E+00	1.286E-15	-1.001E+00	-1.001E+00	1.722E-10	-1.001E+00	-1.001E+00	8.296E-15
g04	-3.067E+04	-3.067E+04	3.713E-12	-3.067E+04	-3.067E+04	3.713E-12	-3.067E+04	-3.067E+04	3.713E-12
g05	5.126E+03	5.126E+03	2.785E-12	5.126E+03	5.126E+03	2.785E-12	5.126E+03	5.126E+03	2.785E-12
g06	-6.962E+03	-6.962E+03	0.000E+00	-6.962E+03	-6.962E+03	0.000E+00	-6.962E+03	-6.962E+03	0.000E+00
g07	2.431E+01	2.431E+01	7.675E-15	2.431E+01	2.431E+01	7.606E-15	2.431E+01	2.431E+01	1.264E-14
g08	-9.583E-02	-9.583E-02	1.416E-17	-9.583E-02	-9.583E-02	1.416E-17	-9.583E-02	-9.583E-02	1.416E-17
g09	6.806E+02	6.806E+02	3.282E-13	6.806E+02	6.806E+02	3.442E-13	6.806E+02	6.806E+02	3.403E-13
g10	7.049E+03	7.049E+03	3.503E-12	7.049E+03	7.049E+03	3.647E-12	7.049E+03	7.049E+03	3.107E-12
g11	7.499E-01	7.499E-01	1.133E-16	7.499E-01	7.499E-01	1.133E-16	7.499E-01	7.499E-01	1.133E-16
g12	-1.000E+00	-1.000E+00	0.000E+00	-1.000E+00	-1.000E+00	0.000E+00	-1.000E+00	-1.000E+00	0.000E+00
g13	5.693E-02	4.891E-01	2.738E-01	6.750E-02	5.544E-01	2.486E-01	5.436E-02	5.923E-01	2.762E-01
g14	-4.776E+01	-4.776E+01	2.521E-14	-4.776E+01	-4.776E+01	2.521E-14	-4.776E+01	-4.776E+01	2.635E-14
g15	9.617E+02	9.617E+02	5.802E-13	9.617E+02	9.617E+02	5.802E-13	9.617E+02	9.617E+02	5.802E-13
g16	-1.905E+00	-1.905E+00	4.532E-16	-1.905E+00	-1.905E+00	4.532E-16	-1.905E+00	-1.905E+00	4.532E-16
g17	8.854E+03	8.887E+03	4.059E+01	8.854E+03	8.893E+03	4.047E+01	8.854E+03	8.896E+03	4.132E+01
g18	-8.660E-01	-8.507E-01	5.290E-02	-8.660E-01	-8.355E-01	7.148E-02	-8.660E-01	-8.355E-01	7.148E-02
g19	3.266E+01	3.266E+01	3.365E-10	3.266E+01	3.266E+01	8.824E-10	3.266E+01	3.266E+01	5.710E-10
g21	1.937E+02	2.067E+02	3.765E+01	1.937E+02	2.042E+02	3.627E+01	1.937E+02	2.021E+02	2.782E+01
g23	-4.001E+02	-4.000E+02	1.329E-01	-4.001E+02	-3.998E+02	1.217E+00	-4.001E+02	-4.001E+02	2.426E-03
g24	-5.508E+00	-5.508E+00	9.065E-16	-5.508E+00	-5.508E+00	9.065E-16	-5.508E+00	-5.508E+00	9.065E-16

Table B.3: Function values obtained by proposed MLS-MODE with different CS values of CS= 100, 125 and 150 generations

problems	CS=**100**			CS=**125**			CS=**150**		
	Best	**Mean**	**Std.**	**Best**	**Mean**	**Std.**	**Best**	**Mean**	**Std.**
g01	-1.500E+01	-1.500E+01	0.000E+00	-1.500E+01	-1.500E+01	0.000E+00	-1.500E+01	-1.500E+01	0.000E+00
g02	-8.036E-01	-8.032E-01	2.202E-03	-8.036E-01	-8.036E-01	1.088E-06	-8.036E-01	-8.028E-01	2.780E-03
g03	-1.001E+00	-1.001E+00	1.041E-15	-1.001E+00	-1.001E+00	8.928E-16	-1.001E+00	-1.001E+00	5.368E-15
g04	-3.067E+04	-3.067E+04	3.713E-12	-3.067E+04	-3.067E+04	3.713E-12	-3.067E+04	-3.067E+04	3.713E-12
g05	5.126E+03	5.126E+03	2.828E-12	5.126E+03	5.126E+03	2.785E-12	5.126E+03	5.126E+03	2.785E-12
g06	-6.962E+03	-6.962E+03	0.000E+00	-6.962E+03	-6.962E+03	0.000E+00	-6.962E+03	-6.962E+03	0.000E+00
g07	2.431E+01	2.431E+01	8.010E-15	2.431E+01	2.431E+01	1.359E-14	2.431E+01	2.431E+01	2.846E-13
g08	-9.583E-02	-9.583E-02	1.499E-17	-9.583E-02	-9.583E-02	1.499E-17	-9.583E-02	-9.583E-02	1.499E-17
g09	6.806E+02	6.806E+02	3.363E-13	6.806E+02	6.806E+02	3.481E-13	6.806E+02	6.806E+02	3.442E-13
g10	7.049E+03	7.049E+03	2.754E-12	7.049E+03	7.049E+03	3.383E-12	7.049E+03	7.049E+03	2.435E-12
g11	7.499E-01	7.499E-01	1.133E-16	7.499E-01	7.499E-01	1.133E-16	7.499E-01	7.499E-01	1.133E-16
g12	-1.000E+00	-1.000E+00	0.000E+00	-1.000E+00	-1.000E+00	0.000E+00	-1.000E+00	-1.000E+00	0.000E+00
g13	6.280E-02	6.005E-01	2.538E-01	5.394E-02	5.445E-01	2.484E-01	5.433E-02	5.172E-01	2.042E-01
g14	-4.776E+01	-4.776E+01	2.578E-14	-4.776E+01	-4.776E+01	2.578E-14	-4.776E+01	-4.776E+01	2.491E-14
g15	9.617E+02	9.617E+02	5.802E-13	9.617E+02	9.617E+02	5.802E-13	9.617E+02	9.617E+02	5.802E-13
g16	-1.905E+00	-1.905E+00	4.532E-16	-1.905E+00	-1.905E+00	4.532E-16	-1.905E+00	-1.905E+00	4.532E-16
g17	8.854E+03	8.893E+03	4.170E+01	8.854E+03	8.895E+03	4.345E+01	8.854E+03	8.896E+03	4.447E+01
g18	-8.660E-01	-8.507E-01	5.290E-02	-8.660E-01	-8.507E-01	5.290E-02	-8.660E-01	-8.507E-01	5.290E-02
g19	3.266E+01	3.266E+01	1.037E-09	3.266E+01	3.266E+01	1.076E-09	3.266E+01	3.266E+01	2.142E-08
g21	1.937E+02	1.963E+02	1.281E+01	1.937E+02	2.011E+02	2.293E+01	1.937E+02	2.029E+02	2.557E+01
g23	-4.001E+02	-4.000E+02	3.473E-01	-4.001E+02	-4.000E+02	1.337E-01	-4.001E+02	-4.000E+02	9.186E-02
g24	-5.508E+00	-5.508E+00	9.065E-16	-5.508E+00	-5.508E+00	9.065E-16	-5.508E+00	-5.508E+00	9.065E-16

Table B.4: Function values obtained by proposed MLS-MODE with different NP^{init} values of NP^{init}= 50, 75 and 100 individuals

problems	NP^{init}=**50**			NP^{init}=**75**			NP^{init}=**100**		
	Best	**Mean**	**Std.**	**Best**	**Mean**	**Std.**	**Best**	**Mean**	**Std.**
g01	-1.500E+01	-1.500E+01	0.000E+00	-1.500E+01	-1.500E+01	0.000E+00	-1.500E+01	-1.500E+01	0.000E+00
g02	-8.036E-01	-7.957E-01	8.246E-03	-8.036E-01	-8.009E-01	8.003E-03	-8.036E-01	-8.032E-01	2.202E-03
g03	-1.001E+00	-9.812E-01	6.771E-02	-1.001E+00	-1.001E+00	5.027E-16	-1.001E+00	-1.001E+00	3.846E-16
g04	-3.067E+04	-3.067E+04	3.713E-12	-3.067E+04	-3.067E+04	3.713E-12	-3.067E+04	-3.067E+04	3.713E-12
g05	5.126E+03	5.126E+03	1.339E-12	5.126E+03	5.126E+03	2.785E-12	5.126E+03	5.126E+03	2.785E-12
g06	-6.962E+03	-6.962E+03	0.000E+00	-6.962E+03	-6.962E+03	0.000E+00	-6.962E+03	-6.962E+03	0.000E+00
g07	2.431E+01	2.431E+01	3.518E-10	2.431E+01	2.431E+01	3.182E-14	2.431E+01	2.431E+01	1.533E-14
g08	-9.583E-02	-9.583E-02	1.267E-17	-9.583E-02	-9.583E-02	1.416E-17	-9.583E-02	-9.583E-02	1.416E-17
g09	6.806E+02	6.806E+02	3.890E-13	6.806E+02	6.806E+02	3.603E-13	6.806E+02	6.806E+02	6.240E-12
g10	7.049E+03	7.049E+03	2.230E-05	7.049E+03	7.049E+03	3.581E-12	7.049E+03	7.049E+03	1.553E-12
g11	7.499E-01	7.499E-01	1.133E-16	7.499E-01	7.499E-01	1.133E-16	7.499E-01	7.499E-01	1.133E-16
g12	-1.000E+00	-1.000E+00	0.000E+00	-1.000E+00	-1.000E+00	0.000E+00	-1.000E+00	-1.000E+00	0.000E+00
g13	5.394E-02	2.804E-01	1.856E-01	5.394E-02	3.298E-01	2.751E-01	5.610E-02	4.502E-01	2.616E-01
g14	-4.776E+01	-4.776E+01	4.839E-13	-4.776E+01	-4.776E+01	2.302E-14	-4.776E+01	-4.776E+01	2.431E-14
g15	9.617E+02	9.617E+02	5.802E-13	9.617E+02	9.617E+02	5.802E-13	9.617E+02	9.617E+02	5.802E-13
g16	-1.905E+00	-1.905E+00	7.036E-16	-1.905E+00	-1.905E+00	4.532E-16	-1.905E+00	-1.905E+00	4.532E-16
g17	8.854E+03	8.909E+03	3.495E+01	8.854E+03	8.901E+03	3.881E+01	8.854E+03	8.893E+03	4.118E+01
g18	-8.660E-01	-7.896E-01	9.552E-02	-8.660E-01	-8.355E-01	7.148E-02	-8.660E-01	-8.278E-01	7.799E-02
g19	3.266E+01	3.266E+01	2.234E-07	3.266E+01	3.266E+01	1.257E-10	3.266E+01	3.266E+01	4.717E-10
g21	1.937E+02	2.147E+02	4.901E+01	1.937E+02	2.199E+02	5.347E+01	1.937E+02	2.046E+02	2.984E+01
g23	-4.001E+02	-4.001E+02	2.110E-04	-4.001E+02	-4.001E+02	1.033E-08	-4.001E+02	-4.000E+02	1.486E-01
g24	-5.508E+00	-5.508E+00	9.065E-16	-5.508E+00	-5.508E+00	9.065E-16	-5.508E+00	-5.508E+00	9.065E-16

Table B.5: Function values obtained by proposed MLS-MODE with different NP^{init} values of NP^{init}= 125, 150 and 200 individuals

problems	NP^{init}=125			NP^{init}=150			NP^{init}=200		
	Best	**Mean**	**Std.**	**Best**	**Mean**	**Std.**	**Best**	**Mean**	**Std.**
g01	-1.500E+01	-1.500E+01	0.000E+00	-1.500E+01	-1.500E+01	0.000E+00	-1.500E+01	-1.500E+01	0.000E+00
g02	-8.036E-01	-8.021E-01	3.731E-03	-8.036E-01	-8.032E-01	2.202E-03	-8.036E-01	-8.025E-01	3.174E-03
g03	-1.001E+00	-1.001E+00	5.631E-12	-1.001E+00	-1.001E+00	5.582E-14	-1.001E+00	-1.001E+00	2.486E-12
g04	-3.067E+04	-3.067E+04	3.713E-12	-3.067E+04	-3.067E+04	3.713E-12	-3.067E+04	-3.067E+04	3.713E-12
g05	5.126E+03	5.126E+03	2.785E-12	5.126E+03	5.126E+03	2.785E-12	5.126E+03	5.126E+03	2.828E-12
g06	-6.962E+03	-6.962E+03	0.000E+00	-6.962E+03	-6.962E+03	0.000E+00	-6.962E+03	-6.962E+03	0.000E+00
g07	2.431E+01	2.431E+01	1.281E-14	2.431E+01	2.431E+01	8.793E-15	2.431E+01	2.431E+01	5.078E-12
g08	-9.583E-02	-9.583E-02	1.416E-17	-9.583E-02	-9.583E-02	1.416E-17	-9.583E-02	-9.583E-02	1.577E-17
g09	6.806E+02	6.806E+02	3.363E-13	6.806E+02	6.806E+02	3.442E-13	6.806E+02	6.806E+02	3.363E-13
g10	7.049E+03	7.049E+03	2.449E-12	7.049E+03	7.049E+03	3.367E-09	7.049E+03	7.049E+03	1.620E-08
g11	7.499E-01	7.499E-01	1.133E-16	7.499E-01	7.499E-01	1.133E-16	7.499E-01	7.499E-01	1.133E-16
g12	-1.000E+00	-1.000E+00	0.000E+00	-1.000E+00	-1.000E+00	0.000E+00	-1.000E+00	-1.000E+00	0.000E+00
g13	5.394E-02	4.338E-01	1.805E-01	5.394E-02	3.589E-01	2.196E-01	5.394E-02	3.376E-01	2.502E-01
g14	-4.776E+01	-4.776E+01	2.491E-14	-4.776E+01	-4.776E+01	2.431E-14	-4.776E+01	-4.776E+01	2.521E-14
g15	9.617E+02	9.617E+02	5.802E-13	9.617E+02	9.617E+02	5.802E-13	9.617E+02	9.617E+02	5.802E-13
g16	-1.905E+00	-1.905E+00	4.532E-16	-1.905E+00	-1.905E+00	4.532E-16	-1.905E+00	-1.905E+00	4.532E-16
g17	8.854E+03	8.894E+03	4.199E+01	8.854E+03	8.898E+03	3.886E+01	8.854E+03	8.891E+03	3.831E+01
g18	-8.660E-01	-8.660E-01	1.855E-16	-8.660E-01	-8.660E-01	1.855E-16	-8.660E-01	-8.660E-01	5.193E-16
g19	3.266E+01	3.266E+01	1.446E-09	3.266E+01	3.266E+01	1.704E-14	3.266E+01	3.266E+01	1.931E-07
g21	1.937E+02	2.055E+02	3.644E+01	1.937E+02	1.990E+02	2.620E+01	1.937E+02	2.042E+02	3.627E+01
g23	-4.001E+02	-3.999E+02	6.658E-01	-4.001E+02	-4.001E+02	4.325E-06	-4.001E+02	-4.001E+02	6.101E-03
g24	-5.508E+00	-5.508E+00	9.065E-16	-5.508E+00	-5.508E+00	9.065E-16	-5.508E+00	-5.508E+00	9.065E-16

Table B.6: Function values obtained by proposed MLS-MODE with different NP^{min} values of NP^{min}= 4, 10 and 20 individuals

problems	NP^{min}=**4**			NP^{min}=**10**			NP^{min}=**20**		
	Best	**Mean**	**Std.**	**Best**	**Mean**	**Std.**	**Best**	**Mean**	**Std.**
g01	-1.500E+01	-1.500E+01	0.000E+00	-1.500E+01	-1.500E+01	0.000E+00	-1.500E+01	-1.500E+01	0.000E+00
g02	-8.036E-01	-8.027E-01	3.046E-03	-8.036E-01	-8.020E-01	3.759E-03	-8.036E-01	-8.036E-01	1.906E-06
g03	-1.001E+00	-1.001E+00	4.382E-08	-1.001E+00	-1.001E+00	1.390E-11	-1.001E+00	-1.001E+00	1.639E-06
g04	-3.067E+04	-3.067E+04	3.713E-12	-3.067E+04	-3.067E+04	3.713E-12	-3.067E+04	-3.067E+04	3.713E-12
g05	5.126E+03	5.126E+03	2.870E-12	5.126E+03	5.126E+03	2.785E-12	5.126E+03	5.126E+03	2.785E-12
g06	-6.962E+03	-6.962E+03	0.000E+00	-6.962E+03	-6.962E+03	0.000E+00	-6.962E+03	-6.962E+03	0.000E+00
g07	2.431E+01	2.431E+01	1.798E-14	2.431E+01	2.431E+01	3.455E-13	2.431E+01	2.431E+01	7.216E-15
g08	-9.583E-02	-9.583E-02	1.416E-17	-9.583E-02	-9.583E-02	1.329E-17	-9.583E-02	-9.583E-02	1.416E-17
g09	6.806E+02	6.806E+02	3.458E-13	6.806E+02	6.806E+02	3.403E-13	6.806E+02	6.806E+02	3.496E-13
g10	7.049E+03	7.049E+03	4.829E-11	7.049E+03	7.049E+03	3.823E-08	7.049E+03	7.049E+03	2.347E-08
g11	7.499E-01	7.499E-01	1.133E-16	7.499E-01	7.499E-01	1.133E-16	7.499E-01	7.499E-01	1.133E-16
g12	-1.000E+00	-1.000E+00	0.000E+00	-1.000E+00	-1.000E+00	0.000E+00	-1.000E+00	-1.000E+00	0.000E+00
g13	5.394E-02	4.119E-01	2.382E-01	5.394E-02	3.972E-01	2.376E-01	5.394E-02	4.143E-01	2.672E-01
g14	-4.776E+01	-4.776E+01	2.491E-14	-4.776E+01	-4.776E+01	2.284E-14	-4.776E+01	-4.776E+01	2.461E-14
g15	9.617E+02	9.617E+02	5.802E-13	9.617E+02	9.617E+02	5.802E-13	9.617E+02	9.617E+02	5.802E-13
g16	-1.905E+00	-1.905E+00	4.532E-16	-1.905E+00	-1.905E+00	4.532E-16	-1.905E+00	-1.905E+00	4.532E-16
g17	8.854E+03	8.906E+03	3.638E+01	8.854E+03	8.911E+03	3.635E+01	8.854E+03	8.895E+03	3.965E+01
g18	-8.660E-01	-8.584E-01	3.821E-02	-8.660E-01	-8.584E-01	3.821E-02	-8.660E-01	-8.660E-01	1.813E-16
g19	3.266E+01	3.266E+01	2.859E-09	3.266E+01	3.266E+01	3.793E-11	3.266E+01	3.266E+01	6.880E-14
g21	1.937E+02	1.990E+02	2.620E+01	1.937E+02	1.937E+02	1.081E-08	1.937E+02	2.042E+02	3.627E+01
g23	-4.001E+02	-4.001E+02	4.343E-07	-4.001E+02	-4.001E+02	2.022E-04	-4.001E+02	-4.001E+02	9.662E-08
g24	-5.508E+00	-5.508E+00	9.065E-16	-5.508E+00	-5.508E+00	9.065E-16	-5.508E+00	-5.508E+00	9.065E-16

Table B.7: Function values obtained by proposed MLS-MODE with different NP^{min} values of NP^{min}= 30, 40 and 50 individuals

problems	NP^{min}=**30**			NP^{min}=**40**			NP^{min}=**50**		
	Best	**Mean**	**Std.**	**Best**	**Mean**	**Std.**	**Best**	**Mean**	**Std.**
g01	-1.500E+01	-1.500E+01	0.000E+00	-1.500E+01	-1.500E+01	0.000E+00	-1.500E+01	-1.500E+01	0.000E+00
g02	-8.036E-01	-8.036E-01	3.384E-06	-8.036E-01	-8.032E-01	2.202E-03	-8.036E-01	-8.032E-01	2.202E-03
g03	-1.001E+00	-1.001E+00	1.232E-15	-1.001E+00	-1.001E+00	5.582E-14	-1.001E+00	-1.001E+00	4.742E-13
g04	-3.067E+04	-3.067E+04	3.713E-12	-3.067E+04	-3.067E+04	3.713E-12	-3.067E+04	-3.067E+04	3.713E-12
g05	5.126E+03	5.126E+03	2.785E-12	5.126E+03	5.126E+03	2.785E-12	5.126E+03	5.126E+03	2.785E-12
g06	-6.962E+03	-6.962E+03	0.000E+00	-6.962E+03	-6.962E+03	0.000E+00	-6.962E+03	-6.962E+03	0.000E+00
g07	2.431E+01	2.431E+01	8.852E-15	2.431E+01	2.431E+01	8.793E-15	2.431E+01	2.431E+01	7.640E-15
g08	-9.583E-02	-9.583E-02	1.416E-17	-9.583E-02	-9.583E-02	1.416E-17	-9.583E-02	-9.583E-02	1.577E-17
g09	6.806E+02	6.806E+02	3.403E-13	6.806E+02	6.806E+02	3.442E-13	6.806E+02	6.806E+02	3.363E-13
g10	7.049E+03	7.049E+03	6.240E-09	7.049E+03	7.049E+03	3.367E-09	7.049E+03	7.049E+03	2.091E-08
g11	7.499E-01	7.499E-01	1.133E-16	7.499E-01	7.499E-01	1.133E-16	7.499E-01	7.499E-01	1.133E-16
g12	-1.000E+00	-1.000E+00	0.000E+00	-1.000E+00	-1.000E+00	0.000E+00	-1.000E+00	-1.000E+00	0.000E+00
g13	5.394E-02	3.319E-01	1.697E-01	5.394E-02	3.589E-01	2.196E-01	5.394E-02	3.959E-01	2.396E-01
g14	-4.776E+01	-4.776E+01	2.607E-14	-4.776E+01	-4.776E+01	2.431E-14	-4.776E+01	-4.776E+01	2.521E-14
g15	9.617E+02	9.617E+02	5.802E-13	9.617E+02	9.617E+02	5.802E-13	9.617E+02	9.617E+02	5.802E-13
g16	-1.905E+00	-1.905E+00	4.532E-16	-1.905E+00	-1.905E+00	4.532E-16	-1.905E+00	-1.905E+00	4.532E-16
g17	8.854E+03	8.894E+03	3.917E+01	8.854E+03	8.898E+03	3.886E+01	8.854E+03	8.885E+03	3.934E+01
g18	-8.660E-01	-8.584E-01	3.821E-02	-8.660E-01	-8.584E-01	3.821E-02	-8.660E-01	-8.584E-01	3.821E-02
g19	3.266E+01	3.266E+01	1.951E-14	3.266E+01	3.266E+01	1.704E-14	3.266E+01	3.266E+01	9.621E-15
g21	1.937E+02	1.990E+02	2.620E+01	1.937E+02	1.990E+02	2.620E+01	1.937E+02	1.990E+02	2.620E+01
g23	-4.001E+02	-4.001E+02	2.498E-05	-4.001E+02	-4.001E+02	4.325E-06	-4.001E+02	-4.001E+02	2.059E-05
g24	-5.508E+00	-5.508E+00	9.065E-16	-5.508E+00	-5.508E+00	9.065E-16	-5.508E+00	-5.508E+00	9.065E-16

Table B.8: Function values obtained by MLS-MODE, Q-MODE, D-MODE, QD-MODE and MFDC-MODE for g01-g07

problem	Alg.	Best	Mean	Std.
g01	D-MODE	-1.500000E+01	-1.500000E+01	0.000000E+00
	Q-MODE	-1.500000E+01	-1.500000E+01	0.000000E+00
	QD-MODE	-1.500000E+01	-1.500000E+01	0.000000E+00
	MFDC-MODE	-1.500000E+01	-1.500000E+01	0.000000E+00
	MLS-MODE	-1.500000E+01	-1.500000E+01	0.000000E+00
g02	D-MODE	-8.036185E-01	-7.962040E-01	1.639750E-02
	Q-MODE	-8.036185E-01	-8.011601E-01	5.202655E-03
	QD-MODE	-8.036188E-01	-7.999280E-01	9.133567E-03
	MFDC-MODE	-8.036191E-01	-8.000662E-01	1.258981E-02
	MLS-MODE	-8.036191E-01	-8.036170E-01	4.381993E-06
g03	D-MODE	-1.000500E+00	-9.997030E-01	3.393859E-03
	Q-MODE	-1.000500E+00	-9.857461E-01	5.671599E-02
	QD-MODE	-1.000500E+00	-9.991963E-01	6.493081E-03
	MFDC-MODE	-1.000500E+00	-9.308200E-01	1.242538E-01
	MLS-MODE	-1.000500E+00	-1.000356E+00	7.186536E-04
g04	D-MODE	-3.066554E+04	-3.066554E+04	3.712997E-12
	Q-MODE	-3.066554E+04	-3.066554E+04	3.712997E-12
	QD-MODE	-3.066554E+04	-3.066554E+04	3.712997E-12
	MFDC-MODE	-3.066554E+04	-3.066554E+04	3.712997E-12
	MLS-MODE	-3.066554E+04	-3.066554E+04	3.712997E-12
g05	D-MODE	5.126497E+03	5.126497E+03	2.784747E-12
	Q-MODE	5.126497E+03	5.126497E+03	1.947112E-12
	QD-MODE	5.126497E+03	5.126497E+03	2.911804E-12
	MFDC-MODE	5.126497E+03	5.126497E+03	2.784747E-12
	MLS-MODE	5.126497E+03	5.126497E+03	2.784747E-12
g06	D-MODE	-6.961814E+03	-6.961814E+03	0.000000E+00
	Q-MODE	-6.961814E+03	-6.961814E+03	0.000000E+00
	QD-MODE	-6.961814E+03	-6.961814E+03	0.000000E+00
	MFDC-MODE	-6.961814E+03	-6.961814E+03	0.000000E+00
	MLS-MODE	-6.961814E+03	-6.961814E+03	0.000000E+00
g07	D-MODE	2.430621E+01	2.430621E+01	7.858915E-14
	Q-MODE	2.430621E+01	2.430621E+01	8.236628E-15
	QD-MODE	2.430621E+01	2.430621E+01	8.792518E-15
	MFDC-MODE	2.430621E+01	2.430621E+01	3.434283E-10
	MLS-MODE	2.430621E+01	2.430621E+01	5.279497E-15

Table B.9: Function values obtained by MLS-MODE, Q-MODE, D-MODE, QD-MODE and MFDC-MODE for g08-g14

problem	Alg.	Best	Mean	Std.
g08	D-MODE	-9.582504E-02	-9.582504E-02	1.444446E-17
	Q-MODE	-9.582504E-02	-9.582504E-02	1.234785E-17
	QD-MODE	-9.582504E-02	-9.582504E-02	1.328697E-17
	MFDC-MODE	-9.582504E-02	-9.582504E-02	1.498972E-17
	MLS-MODE	-9.582504E-02	-9.582504E-02	1.498972E-17
g09	D-MODE	6.806301E+02	6.806301E+02	3.602575E-13
	Q-MODE	6.806301E+02	6.806301E+02	3.617493E-13
	QD-MODE	6.806301E+02	6.806301E+02	3.587596E-13
	MFDC-MODE	6.806301E+02	6.806301E+02	3.549870E-13
	MLS-MODE	6.806301E+02	6.806301E+02	3.322626E-13
g10	D-MODE	7.049248E+03	7.049248E+03	4.020508E-12
	Q-MODE	7.049248E+03	7.049248E+03	4.084296E-12
	QD-MODE	7.049248E+03	7.049248E+03	3.294958E-12
	MFDC-MODE	7.049248E+03	7.049248E+03	4.703718E-09
	MLS-MODE	7.049248E+03	7.049248E+03	3.413131E-12
g11	D-MODE	7.499000E-01	7.499000E-01	1.133117E-16
	Q-MODE	7.499000E-01	7.499000E-01	1.133117E-16
	QD-MODE	7.499000E-01	7.499000E-01	1.133117E-16
	MFDC-MODE	7.499000E-01	7.499000E-01	1.133117E-16
	MLS-MODE	7.499000E-01	7.499000E-01	1.133117E-16
g12	D-MODE	-1.000000E+00	-1.000000E+00	0.000000E+00
	Q-MODE	-1.000000E+00	-1.000000E+00	0.000000E+00
	QD-MODE	-1.000000E+00	-1.000000E+00	0.000000E+00
	MFDC-MODE	-1.000000E+00	-1.000000E+00	0.000000E+00
	MLS-MODE	-1.000000E+00	-1.000000E+00	0.000000E+00
g13	D-MODE	5.394151E-02	2.455780E-01	1.762731E-01
	Q-MODE	5.394151E-02	3.438241E-01	2.258916E-01
	QD-MODE	5.394151E-02	3.493872E-01	2.611549E-01
	MFDC-MODE	5.427658E-02	6.496030E-01	2.773624E-01
	MLS-MODE	5.394151E-02	3.900748E-01	2.663948E-01
g14	D-MODE	-4.776489E+01	-4.776489E+01	2.051160E-14
	Q-MODE	-4.776489E+01	-4.776489E+01	2.284075E-14
	QD-MODE	-4.776489E+01	-4.776489E+01	2.325151E-14
	MFDC-MODE	-4.776489E+01	-4.776489E+01	2.431295E-14
	MLS-MODE	-4.776489E+01	-4.776489E+01	2.461392E-14

Table B.10: Function values obtained by MLS-MODE, Q-MODE, D-MODE, QD-MODE and MFDC-MODE for g15-g24

problem	Alg.	Best	Mean	Std.
g15	D-MODE	9.617150E+02	9.617150E+02	5.801557E-13
	Q-MODE	9.617150E+02	9.617150E+02	5.801557E-13
	QD-MODE	9.617150E+02	9.617150E+02	5.801557E-13
	MFDC-MODE	9.617150E+02	9.617150E+02	5.801557E-13
	MLS-MODE	9.617150E+02	9.617150E+02	5.801557E-13
g16	D-MODE	-1.905155E+00	-1.905155E+00	4.532467E-16
	Q-MODE	-1.905155E+00	-1.905155E+00	4.532467E-16
	QD-MODE	-1.905155E+00	-1.905155E+00	4.532467E-16
	MFDC-MODE	-1.905155E+00	-1.905155E+00	4.532467E-16
	MLS-MODE	-1.905155E+00	-1.905155E+00	4.532467E-16
g17	D-MODE	8.853534E+03	8.884710E+03	3.857447E+01
	Q-MODE	8.853534E+03	8.883499E+03	3.746824E+01
	QD-MODE	8.853534E+03	8.884073E+03	3.818070E+01
	MFDC-MODE	8.853534E+03	8.872295E+03	3.736106E+01
	MLS-MODE	8.853534E+03	8.869420E+03	3.215566E+01
g18	D-MODE	-8.660254E-01	-8.660254E-01	2.299976E-16
	Q-MODE	-8.660254E-01	-8.507419E-01	5.289768E-02
	QD-MODE	-8.660254E-01	-8.583836E-01	3.820879E-02
	MFDC-MODE	-8.660254E-01	-8.583836E-01	3.820879E-02
	MLS-MODE	-8.660254E-01	-8.660254E-01	8.869285E-15
g19	D-MODE	3.265559E+01	3.265559E+01	1.241793E-09
	Q-MODE	3.265559E+01	3.265559E+01	1.677505E-13
	QD-MODE	3.265559E+01	3.265559E+01	9.409662E-14
	MFDC-MODE	3.265559E+01	3.265559E+01	5.226907E-07
	MLS-MODE	3.265559E+01	3.265559E+01	1.555369E-14
g21	D-MODE	1.937245E+02	2.042028E+02	3.626626E+01
	Q-MODE	1.937245E+02	2.042028E+02	3.626626E+01
	QD-MODE	1.937245E+02	1.989636E+02	2.619567E+01
	MFDC-MODE	1.937245E+02	2.106096E+02	4.085840E+01
	MLS-MODE	1.937245E+02	2.019837E+02	2.968475E+01
g23	D-MODE	-4.000551E+02	-3.998953E+02	6.334253E-01
	Q-MODE	-4.000551E+02	-3.977503E+02	4.609776E+00
	QD-MODE	-4.000551E+02	-4.000551E+02	1.209669E-05
	MFDC-MODE	-4.000551E+02	-3.997105E+02	1.048797E+00
	MLS-MODE	-4.000551E+02	-4.000545E+02	3.007736E-03
g24	D-MODE	-5.508013E+00	-5.508013E+00	9.064933E-16
	Q-MODE	-5.508013E+00	-5.508013E+00	9.064933E-16
	QD-MODE	-5.508013E+00	-5.508013E+00	9.064933E-16
	MFDC-MODE	-5.508013E+00	-5.508013E+00	9.064933E-16
	MLS-MODE	-5.508013E+00	-5.508013E+00	9.064933E-16

Table B.11: Function values obtained by MLS-MODE, Q-MODE, D-MODE, QD-MODE and MFDC-MODE for r01-r06

problem	Alg.	Best	Mean	Std.
	D-MODE	1.544418E+04	1.544418E+04	2.251581E-06
	Q-MODE	1.544418E+04	1.544436E+04	8.793190E-01
r01	QD-MODE	1.544418E+04	1.544454E+04	1.217362E+00
	MFDC-MODE	1.544418E+04	1.544418E+04	1.893493E-07
	MLS-MODE	1.544418E+04	1.544418E+04	6.184698E-07
	D-MODE	1.806264E+04	1.812390E+04	4.266121E+01
	Q-MODE	1.807523E+04	1.812145E+04	3.789364E+01
r02	QD-MODE	1.807839E+04	1.811303E+04	3.735577E+01
	MFDC-MODE	1.802537E+04	1.808985E+04	2.508071E+01
	MLS-MODE	1.805091E+04	1.810492E+04	3.711234E+01
	D-MODE	3.276626E+04	3.288048E+04	5.457285E+01
	Q-MODE	3.275845E+04	3.284286E+04	5.231876E+01
r03	QD-MODE	3.279047E+04	3.286095E+04	4.968852E+01
	MFDC-MODE	3.284191E+04	3.289203E+04	3.018005E+01
	MLS-MODE	3.279775E+04	3.289093E+04	4.777296E+01
	D-MODE	1.256059E+05	1.272531E+05	9.608122E+02
	Q-MODE	1.253963E+05	1.267771E+05	9.331146E+02
r04	QD-MODE	1.252848E+05	1.267037E+05	6.302580E+02
	MFDC-MODE	1.250032E+05	1.258866E+05	5.946653E+02
	MLS-MODE	1.228168E+05	1.244875E+05	7.285228E+02
	D-MODE	1.818684E+06	1.866965E+06	2.302252E+04
	Q-MODE	1.812447E+06	1.844349E+06	1.947029E+04
r05	QD-MODE	1.850348E+06	1.870118E+06	1.005462E+04
	MFDC-MODE	1.822830E+06	1.843878E+06	1.177728E+04
	MLS-MODE	1.676914E+06	1.781277E+06	4.984852E+04
	D-MODE	5.012135E+04	5.133615E+04	5.742851E+02
	Q-MODE	4.841733E+04	4.999552E+04	9.386280E+02
r06	QD-MODE	5.076869E+04	5.177887E+04	4.442427E+02
	MFDC-MODE	4.958437E+04	5.034825E+04	4.598598E+02
	MLS-MODE	4.870594E+04	5.064304E+04	1.270508E+03

Table B.12: Function values obtained by MLS-MODE, Q-MODE, D-MODE, QD-MODE and MFDC-MODE for r07-r10

problem	Alg.	Best	Mean	Std.
r07	D-MODE	1.064496E+06	1.070949E+06	2.620949E+03
	Q-MODE	1.056977E+06	1.061533E+06	2.508282E+03
	QD-MODE	1.065570E+06	1.070672E+06	2.948745E+03
	MFDC-MODE	1.058648E+06	1.068715E+06	3.571217E+03
	MLS-MODE	1.054462E+06	1.068496E+06	4.924136E+03
r08	D-MODE	9.157478E+05	9.163686E+05	5.202005E+02
	Q-MODE	9.157221E+05	9.162300E+05	3.913204E+02
	QD-MODE	9.157536E+05	9.161650E+05	3.142032E+02
	MFDC-MODE	9.157581E+05	9.161236E+05	2.928442E+02
	MLS-MODE	9.157841E+05	9.161801E+05	2.622739E+02
r09	D-MODE	9.173792E+05	9.204819E+05	1.336856E+03
	Q-MODE	9.178093E+05	9.204829E+05	1.385589E+03
	QD-MODE	9.174963E+05	9.200026E+05	1.403220E+03
	MFDC-MODE	9.184978E+05	9.209209E+05	8.304877E+02
	MLS-MODE	9.197099E+05	9.242760E+05	1.612274E+03
r10	D-MODE	9.304478E+05	9.369086E+05	3.531980E+03
	Q-MODE	9.291824E+05	9.338445E+05	3.424019E+03
	QD-MODE	9.305926E+05	9.359744E+05	3.601523E+03
	MFDC-MODE	9.301181E+05	9.339471E+05	2.817914E+03
	MLS-MODE	9.268975E+05	9.348549E+05	4.451490E+03

Table B.13: Function values obtained by MLS-MODE, EHCT-DE, AIS-ZHY, ISMOADE-CMA, SAMO-DE, ECHT-EP2, ϵDEg, AH-DEa, SAMO-GA, APF-GA and rank-iMDDE for CEC2006 for g01-g04

problem	Alg.	Mean	Std.
g01	MLS-MODE	-1.50000E+01	0.00000E+00
	EHCT_DE	-1.50000E+01	0.00000E+00
	AIS-ZHY	-1.50000E+01	0.00000E+00
	ISMOADE-CMA	-1.50000E+01	0.00000E+00
	SAMO-DE	-1.500000E+01	0.000000E+00
	ECHT-EP2	-1.50000E+01	0.00000E+00
	ϵDEg	-1.50000E+01	0.00000E+00
	AH-DEa	-1.500000E+01	0.000000E+00
	SAMO-GA	-1.500000E+01	0.000000E+00
	APF-GA	-1.500000E+01	0.000000E+00
	rank-iMDDE	-1.50000E+01	0.00000E+00
g02	MLS-MODE	-8.036191E-01	4.509747E-16
	EHCT_DE	-7.936387E-01	1.120000E-02
	AIS-ZHY	-8.021930E-01	5.190000E-10
	ISMOADE-CMA	-7.92440E-01	2.80000E-02
	SAMO-DE	-7.987000E-01	8.801E-03
	ECHT-EP2	-7.998220E-01	1.260000E-02
	ϵDEg	-8.036191E-01	1.750000E-08
	AH-DEa	-8.007000E-01	3.912E-03
	SAMO-GA	-7.960000E-01	5.803E-03
	APF-GA	-8.035000E-01	1.000E-04
	rank-iMDDE	-8.02012E-01	1.11335E-02
g03	MLS-MODE	-1.0005000E+00	0.000000E+00
	EHCT_DE	-1.0005000E+00	0.000000E+00
	AIS-ZHY	-1.0005000E+00	1.7700000E-11
	ISMOADE-CMA	-1.0005000E+00	0.000000E+00
	SAMO-DE	-1.0005000E+00	0.000000E+00
	ECHT-EP2	-1.0005000E+00	0.000000E+00
	ϵDEg	-1.0005000E+00	2.9600000E-31
	AH-DEa	-1.0005000E+00	0.000000E+00
	SAMO-GA	-1.0005000E+00	0.000000E+00
	APF-GA	-1.0005000E+00	0.000000E+00
	rank-iMDDE	-1.0005000E+00	2.9600000E-31
g04	MLS-MODE	-3.06655390E+04	0.00000000E+00
	EHCT_DE	-3.06655390E+04	0.00000000E+00
	AIS-ZHY	-3.06655390E+04	3.69000000E-13
	ISMOADE-CMA	-3.06655390E+04	0.00000000E+00
	SAMO-DE	-3.06655386E+04	0.00000000E+00
	ECHT-EP2	-3.06655390E+04	0.00000000E+00
	ϵDEg	-3.06655390E+04	0.00000000E+00
	AH-DEa	-3.06655387E+04	0.000000E+00
	SAMO-GA	-3.06655387E+04	0.000000E+00
	APF-GA	-3.067000E+04	1.000000E-04
	rank-iMDDE	-3.06655387E+04	7.31261000E-12

Table B.14: Function values obtained by MLS-MODE, EHCT-DE, AIS-ZHY, ISMOADE-CMA, SAMO-DE, ECHT-EP2, ϵDEg, AH-DEa, SAMO-GA, APF-GA and rank-iMDDE for CEC2006 for g05-g08

problem	Alg.	Mean	Std.
g05	MLS-MODE	5.1264970E+03	0.000000E+00
	EHCT_DE	5.126497E+03	0.000000E+00
	AIS-ZHY	5.126498E+03	1.700000E-02
	ISMOADE-CMA	5.126497E+03	0.000000E+00
	SAMO-DE	5.126497E+03	0.000000E+00
	ECHT-EP2	5.126497E+03	0.000000E+00
	ϵDEg	5.126497E+03	0.000000E+00
	AH-DEa	5.126000E+03	0.000000E+00
	SAMO-GA	5.128000E+03	1.117000E+00
	APF-GA	5.128000E+03	1.432000E+00
	rank-iMDDE	5.12650E+03	7.31261E-12
g06	MLS-MODE	-6.9618140E+03	0.0000000E+00
	EHCT_DE	-6.96181400E+03	0.00000000E+00
	AIS-ZHY	-6.96181385E+03	1.90000000E-12
	ISMOADE-CMA	-6.96181388E+03	0.00000000E+00
	SAMO-DE	-6.96181388E+03	0.00000000E+00
	ECHT-EP2	-6.96181400E+03	0.00000000E+00
	ϵDEg	-6.96181388E+03	0.00000000E+00
	AH-DEa	-6.9618140E+03	0.000000E+00
	SAMO-GA	-6.9618140E+03	0.000000E+00
	APF-GA	-6.9618140E+03	4.600000E-12
	rank-iMDDE	-6.96181E+03	1.37111E-11
g07	MLS-MODE	2.4306210E+01	0.000000E+00
	EHCT_DE	2.430620E+01	1.140000E-10
	AIS-ZHY	2.435570E+01	8.200000E-03
	ISMOADE-CMA	2.4306210E+01	0.000000E+00
	SAMO-DE	2.430960E+01	1.590000E-03
	ECHT-EP2	2.430630E+01	3.190000E-05
	ϵDEg	2.430620E+01	2.180000E-15
	AH-DEa	2.431000E+01	0.000000E+00
	SAMO-GA	2.441000E+01	4.591000E-02
	APF-GA	2.431000E+01	0.000000E+00
	rank-iMDDE	2.43062E+01	7.74285E-07
g08	MLS-MODE	-9.5825040E-02	0.000000E+00
	EHCT_DE	-9.582500E-02	0.000000E+00
	AIS-ZHY	-9.582500E-02	0.000000E+00
	ISMOADE-CMA	-9.582500E-02	0.000000E+00
	SAMO-DE	-9.582504E-02	0.000000E+00
	ECHT-EP2	-9.582500E-02	2.610000E-08
	ϵDEg	-9.582500E-02	1.230000E-32
	AH-DEa	-9.582500E-02	0.000000E+00
	SAMO-GA	-9.582500E-02	0.000000E+00
	APF-GA	-9.582500E-02	0.000000E+00
	rank-iMDDE	-9.58250E-02	8.36862E-17

Table B.15: Function values obtained by MLS-MODE, EHCT-DE, AIS-ZHY, ISMOADE-CMA, SAMO-DE, ECHT-EP2, ϵDEg, AH-DEa, SAMO-GA, APF-GA and rank-iMDDE for CEC2006 for g09-g12

problem	Alg.	Mean	Std.
g09	MLS-MODE	6.8063010E+02	0.000000E+00
	EHCT_DE	6.8063010E+02	0.000000E+00
	AIS-ZHY	6.806500E+02	1.200000E-08
	ISMOADE-CMA	6.8063010E+02	0.000000E+00
	SAMO-DE	6.8063010E+02	1.160000E-05
	ECHT-EP2	6.8063010E+02	0.000000E+00
	ϵDEg	6.8063010E+02	0.000000E+00
	AH-DEa	6.8063010E+02	0.000000E+00
	SAMO-GA	6.8063010E+02	1.457000E-03
	APF-GA	6.8063010E+02	0.000000E+00
	rank-iMDDE	6.80630E+02	2.285190E-13
g10	MLS-MODE	7.0492480E+03	6.0989160E-10
	EHCT_DE	7.049248E+03	4.180000E-07
	AIS-ZHY	7.049570E+03	4.500000E-04
	ISMOADE-CMA	7.049248E+03	5.420000E-06
	SAMO-DE	7.059813E+03	7.860000E+00
	ECHT-EP2	7.049249E+03	6.600000E-04
	ϵDEg	7.049248E+03	4.240000E-13
	AH-DEa	7.0492480E+03	1.688000E-09
	SAMO-GA	7.144000E+03	6.786000E+01
	APF-GA	7.078000E+03	5.124000E+01
	rank-iMDDE	7.04925E+03	1.50134E-05
g11	MLS-MODE	7.4990000E-01	0.000000E+00
	EHCT_DE	7.499000E-01	0.000000E+00
	AIS-ZHY	7.499000E-01	1.400000E-08
	ISMOADE-CMA	7.499000E-01	0.000000E+00
	SAMO-DE	7.499000E-01	0.000000E+00
	ECHT-EP2	7.499000E-01	0.000000E+00
	ϵDEg	7.499000E-01	0.000000E+00
	AH-DEa	7.499000E-01	0.000000E+00
	SAMO-GA	7.499000E-01	0.000000E+00
	APF-GA	7.499000E-01	0.000000E+00
	rank-iMDDE	7.49900E-01	1.78531E-15
g12	MLS-MODE	-1.0000000E+00	0.0000000E+00
	EHCT_DE	-1.000000E+00	0.000000E+00
	AIS-ZHY	-1.000000E+00	0.000000E+00
	ISMOADE-CMA	-1.000000E+00	0.000000E+00
	SAMO-DE	-1.000000E+00	0.000000E+00
	ECHT-EP2	-1.000000E+00	0.000000E+00
	ϵDEg	-1.000000E+00	0.000000E+00
	AH-DEa	-1.000000E+00	0.000000E+00
	SAMO-GA	-1.000000E+00	0.000000E+00
	APF-GA	-1.000000E+00	0.000000E+00
	rank-iMDDE	-1.00000E+00	0.00000E+00

Table B.16: Function values obtained by MLS-MODE, EHCT-DE, AIS-ZHY, ISMOADE-CMA, SAMO-DE, ECHT-EP2, ϵDEg, AH-DEa, SAMO-GA, APF-GA and rank-iMDDE for CEC2006 for g13-g16

problem	Alg.	Mean	Std.
g13	MLS-MODE	5.39415E-02	0.000000E+00
	EHCT_DE	5.39415E-02	0.000000E+00
	AIS-ZHY	5.39415E-02	7.800000E-10
	ISMOADE-CMA	5.39415E-02	0.000000E+00
	SAMO-DE	5.39415E-02	1.750000E-08
	ECHT-EP2	5.39415E-02	1.000000E-12
	ϵDEg	5.39415E-02	0.000000E+00
	AH-DEa	5.39415E-02	0.000000E+00
	SAMO-GA	5.403000E-02	5.941000E-05
	APF-GA	5.39415E-02	0.000000E+00
	rank-iMDDE	5.39415E-02	3.03523E-07
g14	MLS-MODE	-4.7764890E+01	2.6066690E-14
	EHCT_DE	-4.776489E+01	3.260000E-13
	AIS-ZHY	-4.776488E+01	1.000000E-12
	ISMOADE-CMA	-4.776489E+01	0.000000E+00
	SAMO-DE	-4.768115E+01	4.040000E-02
	ECHT-EP2	-4.776480E+01	2.720000E-05
	ϵDEg	-4.776489E+01	1.390000E-15
	AH-DEa	-4.776000E+01	3.894000E-05
	SAMO-GA	-4.776489E+01	3.159000E-01
	APF-GA	-4.776000E+01	1.000000E-04
	rank-iMDDE	-4.77649E+01	2.23259E-05
g15	MLS-MODE	9.6171500E+02	0.000000E+00
	EHCT_DE	9.617150E+02	0.000000E+00
	AIS-ZHY	9.617150E+02	0.000000E+00
	ISMOADE-CMA	9.617150E+02	0.000000E+00
	SAMO-DE	9.617150E+02	0.000000E+00
	ECHT-EP2	9.617150E+02	2.010000E-13
	ϵDEg	9.617150E+02	0.000000E+00
	AH-DEa	9.617150E+02	0.000000E+00
	SAMO-GA	9.617150E+02	5.524000E-05
	APF-GA	9.617150E+02	0.000000E+00
	rank-iMDDE	9.61715E+02	7.99817E-13
g16	MLS-MODE	-1.9051550E+00	0.000000E+00
	EHCT_DE	-1.905155E+00	0.000000E+00
	AIS-ZHY	-1.905155E+00	0.000000E+00
	ISMOADE-CMA	-1.905155E+00	0.000000E+00
	SAMO-DE	-1.905155E+00	0.000000E+00
	ECHT-EP2	-1.905155E+00	1.120000E-10
	ϵDEg	-1.905155E+00	1.580000E-30
	AH-DEa	-1.905155E+00	0.000000E+00
	SAMO-GA	-1.905155E+00	6.952000E-07
	APF-GA	-1.905155E+00	0.000000E+00
	rank-iMDDE	-1.905155E+00	1.562140E-15

Table B.17: Function values obtained by MLS-MODE, EHCT-DE, AIS-ZHY, ISMOADE-CMA, SAMO-DE, ECHT-EP2, ϵDEg, AH-DEa, SAMO-GA, APF-GA and rank-iMDDE for CEC2006 for g17-g21

problem	Alg.	Mean	Std.
g17	MLS-MODE	8.853540E+03	0.000000E+00
	EHCT_DE	8.853540E+03	0.000000E+00
	AIS-ZHY	8.853540E+03	1.900000E-09
	ISMOADE-CMA	8.853540E+03	0.000000E+00
	SAMO-DE	8.853540E+03	1.150000E-05
	ECHT-EP2	8.853540E+03	2.130000E-08
	ϵDEg	8.853540E+03	1.210000E-27
	AH-DEa	8.858000E+03	1.847000E+01
	SAMO-GA	8.854000E+03	1.740000E-01
	APF-GA	8.888000E+03	2.903000E+01
	rank-iMDDE	8.85354E+03	1.31206E+00
g18	MLS-MODE	-8.6602540E-01	5.1068250E-15
	EHCT_DE	-8.660240E-01	5.150000E-06
	AIS-ZHY	-8.660250E-01	1.300000E-15
	ISMOADE-CMA	-8.660250E-01	0.000000E+00
	SAMO-DE	-8.660240E-01	7.040000E-07
	ECHT-EP2	-8.660250E-01	1.000000E-09
	ϵDEg	-8.660250E-01	2.180000E-17
	AH-DEa	-8.6602540E-01	0.000000E+00
	SAMO-GA	-8.655000E-01	4.080000E-04
	APF-GA	-8.659000E-01	0.000000E+00
	rank-iMDDE	-8.66025E-01	8.62135E-08
g19	MLS-MODE	3.2655590E+01	1.7525120E-14
	EHCT_DE	3.265654E+01	7.760000E-04
	AIS-ZHY	3.265559E+01	0.000000E+00
	ISMOADE-CMA	3.265559E+01	6.460000E-07
	SAMO-DE	3.275734E+01	6.150000E-02
	ECHT-EP2	3.266230E+01	3.400000E-03
	ϵDEg	3.265560E+01	1.260000E-05
	AH-DEa	3.457000E+01	2.524000E+00
	SAMO-GA	3.643000E+01	1.037000E+00
	APF-GA	3.266000E+01	0.000000E+00
	rank-iMDDE	3.26556E+01	1.70016E-03
g21	MLS-MODE	1.937245E+02	0.000000E+00
	EHCT_DE	1.937245E+02	0.000000E+00
	AIS-ZHY	1.967245E+02	1.100000E+00
	ISMOADE-CMA	1.937245E+02	0.000000E+00
	SAMO-DE	1.937714E+02	1.960000E-02
	ECHT-EP2	1.937438E+02	1.650000E-02
	ϵDEg	1.937245E+02	3.340000E-14
	AH-DEa	1.939000E+02	6.977000E-01
	SAMO-GA	2.461000E+02	1.492000E+01
	APF-GA	1.995000E+02	3.866000E+00
	rank-iMDDE	1.93725E+02	1.68174E-10

Table B.18: Function values obtained by MLS-MODE, EHCT-DE, AIS-ZHY, ISMOADE-CMA, SAMO-DE, ECHT-EP2, ϵDEg, AH-DEa, SAMO-GA, APF-GA and rank-iMDDE for CEC2006 for g23-g24

problem	Alg.	Mean	Std.
g23	MLS-MODE	-4.0005120E+02	1.9083780E-02
	EHCT_DE	-4.000546E+02	2.180000E-03
	AIS-ZHY	-3.998743E+02	2.000000E+00
	ISMOADE-CMA	-3.956240E+02	7.790000E+00
	SAMO-DE	-3.608177E+02	1.960000E+01
	ECHT-EP2	-3.732178E+02	3.370000E+01
	ϵDEg	-4.000551E+02	1.110000E-14
	AH-DEa	-3.444000E+02	7.782000E+01
	SAMO-GA	-1.948000E+02	5.328000E+01
	APF-GA	-3.948000E+02	3.866000E+00
	rank-iMDDE	-3.78181E+02	8.20199E-05
g24	MLS-MODE	-5.5080130E+00	0.000000E+00
	EHCT_DE	-5.508013E+00	0.000000E+00
	AIS-ZHY	-5.508013E+00	0.000000E+00
	ISMOADE-CMA	-5.508013E+00	0.000000E+00
	SAMO-DE	-5.508013E+00	0.000000E+00
	ECHT-EP2	-5.508013E+00	0.000000E+00
	ϵDEg	-5.508013E+00	2.520000E-29
	AH-DEa	-5.508000E+00	0.000000E+00
	SAMO-GA	-5.508000E+00	0.000000E+00
	APF-GA	-5.508000E+00	0.000000E+00
	rank-iMDDE	-5.50801E+00	1.78531E-15

Table B.19: Function values obtained by MLS-MODE, DEbavDBmax , SAMODE, DE-DBmax, SAMO-DE, eABC, Co-CLPSO, SAMO-GA,ϵDEg, and ECHT-ARMOR-DE for CEC2010 for c01-c03

		10D			30D		
problem	Alg.	Best	Mean	Std.	Best	Mean	Std.
c01	MLS-MODE	-7.4731036E-01	-7.4731036E-01	4.2142800E-15	-8.2188435E-01	-8.2180624E-01	3.8996670E-04
	DEbavDBmax	-7.473104E-01	-7.463993E-01	2.552887E-03	-8.218843E-01	-8.135627E-01	7.854595E-03
	SAMODE	-7.473104E-01	-7.470402E-01	1.350638E-03	-8.218838E-01	-8.143674E-01	4.766000E-03
	ϵDEg	-7.473104E-01	-7.467701E-01	1.869859E-03	-8.218255E-01	-8.208687E-01	7.103893E-04
	DE-DBmax	-7.473104E-01	-7.463993E-01	2.552887E-03	-8.218843E-01	-8.146829E-01	7.216108E-03
	eABC	-7.472710E-01	-7.162570E-01	2.689780E-02	-8.162650E-01	-7.305540E-01	4.875890E-02
	Co-CLPSO	-7.473100E-01	-7.335800E-01	1.784800E-02	-8.068800E-01	-7.159800E-01	5.025200E-02
	SAMO-GA	-7.473093E-01	-7.470230E-01	1.347658E-03	-8.217813E-01	-8.115299E-01	7.247940E-03
	ECHT-ARMOR-DE	-7.473000E-01	-7.470000E+00	1.400000E-03	-8.180600E-01	-7.899200E-01	2.510000E-02
c02	MLS-MODE	-2.2355979E+00	-2.2123734E+00	1.1919289E-02	-2.1390923E+00	-2.0099045E+00	8.8624158E-02
	DEbavDBmax	-2.277710E+00	-2.198366E+00	8.389727E-02	-2.280973E+00	-2.276646E+00	3.150424E-03
	SAMODE	-2.277709E+00	-2.276842E+00	1.154957E-03	-2.280962E+00	-2.276111E+00	3.706000E-03
	ϵDEg	-2.277702E+00	-2.269502E+00	2.389779E-02	-2.169248E+00	-2.151424E+00	1.197582E-02
	DE-DBmax	-2.277711E+00	-2.182633E+00	1.190727E-01	-2.280973E+00	-2.275178E+00	4.677619E-03
	eABC	-2.155150E+00	-1.248950E-01	1.583590E+00	-1.236470E-01	2.556470E+00	9.429490E-01
	Co-CLPSO	-2.277700E+00	-2.266600E+00	1.461600E-02	-2.280900E+00	-2.202900E+00	1.926700E-01
	SAMO-GA	-2.277497E+00	-2.272609E+00	2.267800E-03	-2.265137E+00	-2.252288E+00	5.747300E-03
	ECHT-ARMOR-DE	-2.277700E+00	-2.277000E+03	3.300000E-03	-2.260700E+00	-2.170600E+00	7.360000E-02
c03	MLS-MODE	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00	1.9003142E-25	4.4111754E-25
	DEbavDBmax	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	6.382170E-25	7.133478E-25
	SAMODE	0.000000E+00	4.173000E-23	1.640510E-22	4.599600E-24	4.826000E-22	1.146000E-21
	ϵDEg	0.000000E+00	0.000000E+00	0.000000E+00	2.867347E+01	2.883785E+01	8.047159E-01
	DE-DBmax	0.000000E+00	2.686721E-26	1.023925E-25	0.000000E+00	1.508909E-16	6.314477E-16
	eABC	8.788210E+11	-	1.009670E+12	-	-	-
	Co-CLPSO	2.474800E-13	3.550200E-01	1.775100E+00	-	-	-
	SAMO-GA	6.492724E-22	1.190663E-10	2.074947E-10	5.481000E-19	2.255000E-07	8.154000E-07
	ECHT-ARMOR-DE	0.000000E+00	0.000000E+00	0.000000E+00	2.580100E-24	2.638000E+01	7.940000E+00

Table B.20: Function values obtained by MLS-MODE, DEbavDBmax , SAMODE, DE-DBmax, SAMO-DE, eABC, Co-CLPSO, SAMO-GA,ϵDEg, and ECHT-ARMOR-DE for CEC2010 for c04-c06

		10D			30D		
problem	Alg.	Best	Mean	Std.	Best	Mean	Std.
c04	MLS-MODE	-1.0000000E-05	-1.0000000E-05	0.0000000E+00	-3.3333333E-06	-3.3333333E-06	1.1749496E-15
	DEbavDBmax	-1.000000E-05	-1.000000E-05	0.000000E+00	-3.332962E-06	-3.330868E-06	2.772148E-09
	SAMODE	-1.000000E-05	-1.000000E-05	1.446083E-11	-3.248000E-06	-2.411300E-06	4.492340E-07
	ϵDEg	-9.992345E-06	-9.918452E-06	1.546730E-07	4.698111E-03	8.162973E-03	3.067785E-03
	DE-DBmax	-1.000000E-05	-1.000000E-05	0.000000E+00	-3.333311E-06	-3.331620E-06	1.710576E-09
	eABC	-	-	-	-	-	-
	Co-CLPSO	-1.000000E-05	-9.338500E-06	1.074800E-06	-2.930000E-06	1.126900E-01	5.633500E-01
	SAMO-GA	-1.000000E-05	-9.934308E-06	5.118391E-08	-2.644300E-06	1.470915E-03	7.359431E-03
	ECHT-ARMOR-DE	-1.000000E-05	-1.000000E-05	0.000000E+00	-3.332600E-06	8.371300E-02	2.890000E-01
c05	MLS-MODE	-4.8361062E+02	-4.8361062E+02	3.4809343E-13	-4.8361062E+02	-4.8361062E+02	1.2749496E-14
	DEbavDBmax	-4.836106E+02	-4.836106E+02	4.948079E-06	-4.836106E+02	-4.836106E+02	7.160225E-09
	SAMODE	-4.836106E+02	-4.836106E+02	4.144280E-06	-4.836106E+02	-4.836106E+02	5.389910E-06
	ϵDEg	-4.836106E+02	-4.836106E+02	3.890350E-13	-4.531307E+02	-4.495460E+02	2.899105E+00
	DE-DBmax	-4.836106E+02	-4.835891E+02	3.316223E-02	-4.836106E+02	-4.836106E+02	3.598556E-08
	eABC	4.659550E+01	3.652470E+02	1.172050E+02	1.463110E+02	3.718920E+02	7.888540E+01
	Co-CLPSO	-4.836100E+02	-4.836000E+02	1.957700E-02	-4.836000E+02	-3.124900E+02	8.833200E+01
	SAMO-GA	-4.836106E+02	-4.016976E+02	1.112600E+02	-4.784754E+02	-4.716434E+02	3.783287E+00
	ECHT-ARMOR-DE	-4.836100E+02	-4.836100E+02	0.000000E+00	-4.812200E+02	-4.333500E+02	1.460000E+02
c06	MLS-MODE	-5.7866237E+02	-5.7866237E+02	4.2790171E-13	-5.3063786E+02	-5.3026981E+02	5.9692277E-01
	DEbavDBmax	-5.755047E+02	-5.755047E+02	7.677759E-07	-5.306379E+02	-5.305663E+02	2.346488E-01
	SAMODE	-5.786624E+02	-5.786562E+02	9.352190E-03	-5.306368E+02	-5.306155E+02	1.288050E-02
	ϵDEgg	-5.786581E+02	-5.786528E+02	3.627169E-03	-5.285750E+02	-5.279068E+02	4.748378E-01
	DE-DBmax	-5.755047E+02	-5.755047E+02	3.512664E-05	-5.306379E+02	-5.306316E+02	1.817352E-02
	eABC	2.384680E+02	4.381680E+02	8.595360E+01	3.259620E+02	4.738410E+02	6.302590E+01
	Co-CLPSO	-5.786600E+02	-5.786600E+02	5.728900E-04	-2.860100E+02	-2.447000E+02	3.948100E+01
	SAMO-GA	-5.786619E+02	-5.777351E+02	3.260495E+00	-5.249595E+02	-5.208121E+02	3.169611E+00
	ECHT-ARMOR-DE	-5.786600E+02	-5.786600E+02	4.000000E-13	-5.246500E+02	-4.893100E+02	1.320000E+02

Table B.21: Function values obtained by MLS-MODE, DEbavDBmax , SAMODE, DE-DBmax, SAMO-DE, eABC, Co-CLPSO, SAMO-GA,ϵDEg, and ECHT-ARMOR-DE for CEC2010 for c07-c09

		10D			30D		
problem	Alg.	Best	Mean	Std.	Best	Mean	Std.
c07	MLS-MODE	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00	3.0805556E-27	5.4724404E-27
	DEbavDBmax	0.000000E+00	1.250285E-28	2.274570E-28	0.000000E+00	1.594650E-01	7.973248E-01
	SAMODE	0.000000E+00	7.762750E-23	3.880800E-22	9.495250E-23	1.782790E-13	3.628040E-13
	ϵDEg	0.000000E+00	0.000000E+00	0.000000E+00	1.147112E-15	2.603632E-15	1.233430E-15
	DE-DBmax	0.000000E+00	1.250285E-28	2.274570E-28	0.000000E+00	3.189299E-01	1.103846E+00
	eABC	1.044680E-03	7.160940E+01	5.191130E+01	1.761440E-01	1.332900E+02	2.058280E+02
	Co-CLPSO	1.071100E-09	7.973200E-01	1.627500E+00	3.786100E-11	1.116300E+00	1.826900E+00
	SAMO-GA	0.000000E+00	7.856600E-23	1.208100E-22	3.236800E-28	3.236800E-28	4.576600E-44
	ECHT-ARMOR-DE	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	1.078900E-25	2.200000E-25
c08	MLS-MODE	0.0000000E+00	8.7384827E+00	4.4599415E+00	0.0000000E+00	4.3791390E-27	8.8083879E-27
	DEbavDBmax	0.000000E+00	9.216872E+00	3.595848E+00	1.012710E-26	4.200829E-26	4.917721E-26
	SAMODE	0.000000E+00	2.520090E-25	1.260047E-24	6.425810E-21	1.032930E-09	2.373300E-09
	ϵDEg	0.000000E+00	6.727528E+00	5.560648E+00	2.518693E-14	7.831464E-14	4.855177E-14
	DE-DBmax	1.938261E-23	1.045963E+01	2.182499E+00	4.643926E-28	8.265845E-26	1.185239E-25
	eABC	2.870260E-01	4.107890E+02	9.356030E+02	1.600450E-02	1.501960E+02	7.148770E+01
	Co-CLPSO	9.644200E-10	6.087600E-01	1.425500E+00	4.311400E-14	4.751700E+01	1.125900E+02
	SAMO-GA	3.654300E-25	3.648700E-23	5.693200E-23	3.236790E-28	1.594860E-24	6.776570E-24
	ECHT-ARMOR-DE	0.000000E+00	7.526200E+00	5.000000E+00	0.000000E+00	2.010100E+01	4.700000E+01
c09	MLS-MODE	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00	2.5357772E-25	1.1769998E-24
	DEbavDBmax	0.000000E+00	4.538184E-26	1.839539E-25	0.000000E+00	4.302495E-26	4.474786E-26
	SAMODE	0.000000E+00	5.089752E+00	2.410828E+01	1.186170E-20	6.079667E+00	1.432599E+01
	ϵDEg	0.000000E+00	0.000000E+00	0.000000E+00	2.770665E-16	1.072140E+01	2.821923E+01
	DE-DBmax	0.000000E+00	4.248994E-26	1.564360E-25	0.000000E+00	4.771583E-26	4.869925E-26
	eABC	2.289850E+10	2.019340E+12	1.810830E+12	2.613740E+12	1.607560E+13	9.287410E+12
	Co-CLPSO	3.755100E-16	1.993800E+10	9.968800E+10	1.969500E+02	1.482200E+08	2.450900E+08
	SAMO-GA	3.654300E-25	3.648700E-23	5.693200E-23	3.236790E-28	1.594860E-24	6.776570E-24
	ECHT-ARMOR-DE	0.000000E+00	1.763300E-01	8.800000E-01	0.000000E+00	4.611000E+00	2.310000E+01

Table B.22: Function values obtained by MLS-MODE, DEbavDBmax , SAMODE, DE-DBmax, SAMO-DE, eABC, Co-CLPSO, SAMO-GA,ϵDEg, and ECHT-ARMOR-DE for CEC2010 for c010-c12

		10D			30D		
problem	Alg.	Best	Mean	Std.	Best	Mean	Std.
c10	MLS-MODE	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00	2.9614474E-27	9.6411285E-27
	DEbavDBmax	0.000000E+00	1.333516E-26	3.338908E-26	1.506488E-26	4.412282E-20	1.211799E-19
	SAMODE	0.000000E+00	4.467656E-01	1.546298E+00	9.769000E-21	1.960780E+01	2.123749E+01
	ϵDEg	0.000000E+00	0.000000E+00	0.000000E+00	3.252002E+01	3.326175E+01	4.545577E-01
	DE-DBmax	0.000000E+00	2.165980E-24	1.081388E-23	1.370196E-25	1.540962E-19	2.992547E-19
	eABC	4.493580E+09	1.746200E+12	2.583000E+12	9.358200E+11	1.498640E+13	9.773360E+12
	Co-CLPSO	2.396700E-15	4.974300E+10	2.487100E+11	3.196700E+01	1.395100E+09	5.843800E+09
	SAMO-GA	1.124000E-19	1.732124E+02	2.636938E+02	4.164404E+01	6.031654E+03	7.177385E+03
	ECHT-ARMOR-DE	0.000000E+00	0.000000E+00	0.000000E+00	6.020900E-13	6.553600E+01	1.070000E+02
c11	MLS-MODE	-1.5227132E-03	-1.5227132E-03	6.3511486E-18	-3.9234392E-04	-3.9234367E-04	2.0125659E-10
	DEbavDBmax	-1.522713E-03	-1.522713E-03	1.461597E-14	-3.923439E-04	-3.923436E-04	3.984682E-10
	SAMODE	-1.522710E-03	-1.522710E-03	3.667610E-09	-3.923000E-04	-3.869000E-04	6.149660E-06
	ϵDEg	-1.522710E-03	-1.522710E-03	6.341035E-11	-3.268462E-04	-2.863882E-04	2.707605E-05
	DE-DBmax	-1.522713E-03	-1.522713E-03	6.880150E-16	-3.923439E-04	1.895262E-03	6.322610E-03
	eABC	-	-	-	-	-	-
	Co-CLPSO	-	-	-	-	-	-
	SAMO-GA	-6.205000E-04	-5.262000E-04	4.904300E-05	-1.360000E-04	-1.260000E-04	4.887000E-06
	ECHT-ARMOR-DE	-1.522700E-03	-	4.400000E-02	-3.923400E-04	-	5.280000E-03
c12	MLS-MODE	-1.1602762E+02	-1.6819684E+01	3.7509007E+01	-1.9926346E-01	-1.9926346E-01	5.0128642E-09
	DEbavDBmax	-2.202514E-01	-2.202502E-01	6.620404E-07	-1.992635E-01	-1.992635E-01	1.630986E-09
	SAMODE	-5.700899E+02	-1.166134E+02	1.830005E+02	-1.992598E-01	-1.992573E-01	1.321890E-06
	ϵDEg	-5.700899E+02	-3.367349E+02	1.782166E+02	-1.991453E-01	-	2.889253E+02
	DE-DBmax	-2.202509E-01	-2.202483E-01	2.733060E-06	-1.992635E-01	-1.992635E-01	1.401959E-09
	eABC	-5.700360E+02	-1.800890E+02	2.757580E+02	-8.826380E+02	-	-
	Co-CLPSO	-1.263900E+01	-2.336900E+00	2.432900E+01	-1.992600E-01	-1.991100E-01	1.184000E-04
	SAMO-GA	-5.697441E+02	-5.588351E+01	1.351800E+02	-8.535971E+01	-3.493162E+00	1.705561E+01
	ECHT-ARMOR-DE	-1.992500E-01	-1.992500E-01	1.600000E-13	-1.992600E-01	-1.607600E-01	1.930000E-01

Table B.23: Function values obtained by MLS-MODE, DEbavDBmax , SAMODE, DE-DBmax, SAMO-DE, eABC, Co-CLPSO, SAMO-GA,ϵDEg, and ECHT-ARMOR-DE for CEC2010 for c13-c15

		10D			30D		
problem	Alg.	Best	Mean	Std.	Best	Mean	Std.
c13	MLS-MODE	-6.8429363E+01	-6.8416666E+01	8.3825697E-03	-6.1630347E+01	-5.8816532E+01	1.4972294E+00
	DEbavDBmax	-6.842937E+01	-6.750889E+01	1.414780E+00	-6.583213E+01	-6.018059E+01	5.066890E+00
	SAMODE	-6.842937E+01	-6.842937E+01	1.542877E-07	-6.842940E+01	-6.819178E+01	3.891641E-01
	ϵDEg	-6.842937E+01	-6.842936E+01	1.025960E-06	-6.642473E+01	-6.535310E+01	5.733005E-01
	DE-DBmax	-6.842937E+01	-6.755373E+01	1.275098E+00	-6.575545E+01	-6.002365E+01	5.274321E+00
	eABC	-6.842890E+01	-6.568060E+01	2.502700E+00	-6.756870E+01	-6.485590E+01	1.382130E+00
	Co-CLPSO	-6.842936E+01	-6.397445E+01	2.134080E+00	-6.275200E+01	-6.077400E+01	1.117600E+00
	SAMO-GA	-6.842803E+01	-6.837658E+01	1.558530E-01	-6.637738E+01	-6.309871E+01	1.286011E+00
	ECHT-ARMOR-DE	-6.842900E+01	-6.716900E+01	2.100000E+00	-6.741600E+01	-6.464600E+01	1.970000E+00
c14	MLS-MODE	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00	3.0384713E-27	7.2647679E-27
	DEbavDBmax	0.000000E+00	3.303876E-27	4.718921E-27	0.000000E+00	1.010587E-25	8.052860E-26
	SAMODE	0.000000E+00	1.206380E-21	2.435928E-21	1.747000E-22	1.196910E-08	2.569420E-08
	ϵDEg	0.000000E+00	0.000000E+00	0.000000E+00	5.015863E-14	3.089407E-13	5.608409E-13
	DE-DBmax	0.000000E+00	3.393617E-27	3.469890E-27	1.391551E-27	3.189299E-01	1.103846E+00
	eABC	3.141520E-04	8.004100E+10	2.366080E+11	3.134910E-01	9.947190E+03	1.916060E+04
	Co-CLPSO	5.780000E-12	3.189300E-01	1.103800E+00	3.288340E-09	6.152420E-02	3.073560E-01
	SAMO-GA	1.876000E-22	3.872582E+02	1.244818E+03	4.145243E+00	1.762866E+03	5.465005E+03
	ECHT-ARMOR-DE	0.000000E+00	0.000000E+00	0.000000E+00	1.580900E-27	6.613500E+02	2.470000E+03
c15	MLS-MODE	0.0000000E+00	6.1089389E-29	3.0544694E-28	0.0000000E+00	6.0483543E-28	1.1788763E-27
	DEbavDBmax	0.000000E+00	2.958427E-25	1.086319E-24	5.026464E-26	1.868275E-22	2.968697E-22
	SAMODE	0.000000E+00	7.053800E-04	2.441385E-03	5.842400E-18	2.112810E+00	4.510670E+00
	ϵDEg	0.000000E+00	1.798980E-01	8.813156E-01	2.160345E+01	2.160376E+01	1.104834E-04
	DE-DBmax	2.600086E-27	1.714139E-22	8.450532E-22	1.323884E-24	2.113119E-20	6.666822E-20
	eABC	4.216110E+09	2.565760E+13	2.862760E+13	1.781140E+12	3.785130E+13	3.440730E+13
	Co-CLPSO	3.046900E-12	2.988500E+00	3.314700E+00	5.749900E-12	5.105900E+01	9.175900E+01
	SAMO-GA	2.825000E-20	8.574500E+02	1.792600E+03	8.384400E-01	1.749600E+04	3.411400E+04
	ECHT-ARMOR-DE	0.000000E+00	2.824600E+00	1.600000E+00	1.171600E-04	3.131600E+08	1.200000E+09

Table B.24: Function values obtained by MLS-MODE, DEbavDBmax , SAMODE, DE-DBmax, SAMO-DE, eABC, Co-CLPSO, SAMO-GA,ϵDEg, and ECHT-ARMOR-DE for CEC2010 for c16-c18

		10D			30D		
problem	Alg.	Best	Mean	Std.	Best	Mean	Std.
	MLS-MODE	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00
	DEbavDBmax	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
	SAMODE	0.000000E+00	6.469600E-03	1.087042E-02	0.000000E+00	4.160700E-03	7.677900E-03
	ϵDEg	0.000000E+00	3.702054E-01	3.710479E-01	0.000000E+00	2.168404E-21	1.062297E-20
c16	DE-DBmax	0.000000E+00	1.332800E-01	3.301736E-01	0.000000E+00	0.000000E+00	0.000000E+00
	eABC	0.000000E+00	8.347410E-02	9.106390E-02	6.252940E-02	8.207220E-01	2.569880E-01
	Co-CLPSO	0.000000E+00	5.986100E-03	1.331500E-02	0.000000E+00	5.240300E-16	4.672200E-16
	SAMO-GA	0.000000E+00	1.403400E-03	4.880000E-03	0.000000E+00	3.737000E-03	6.284000E-03
	ECHT-ARMOR-DE	0.000000E+00	2.847800E-02	5.000000E-02	0.000000E+00	0.000000E+00	0.000000E+00
	MLS-MODE	3.3716223E-27	4.4542095E-24	1.1601496E-23	3.3560550E-22	2.3915346E-14	1.1228612E-13
	DEbavDBmax	1.346610E-29	7.053446E-19	1.079196E-18	1.061228E-18	1.811932E-12	5.281940E-12
	SAMODE	0.000000E+00	1.265501E-23	3.217991E-23	0.000000E+00	1.022630E-10	1.455150E-10
	ϵDEg	1.463180E-17	1.249561E-01	1.937197E-01	2.165719E-01	6.326487E+00	4.986691E+00
c17	DE-DBmax	3.098066E-28	1.154546E-05	4.146688E-05	7.114167E-20	6.849019E-12	1.639527E-11
	eABC	2.533850E-02	3.242900E+00	6.831360E+00	3.267520E+00	2.680100E+01	1.634550E+01
	Co-CLPSO	7.667700E-17	3.798600E-01	4.528400E-01	1.578700E-01	1.391900E+00	4.262100E+00
	SAMO-GA	0.000000E+00	1.270700E-02	1.331029E-02	0.000000E+00	1.343600E-02	1.616200E-02
	ECHT-ARMOR-DE	0.000000E+00	3.697800E-33	3.100000E-33	3.356400E-16	4.033600E-01	3.510000E-01
	MLS-MODE	0.0000000E+00	3.2170734E-31	4.1468535E-31	1.4791142E-31	8.9491903E-05	1.1722656E-04
	DEbavDBmax	4.239528E-25	3.885478E-24	4.336668E-24	1.030097E-18	2.829429E-01	1.356367E+00
	SAMODE	2.236640E-15	1.617890E-14	3.820340E-14	1.375370E-14	4.738410E-14	6.573500E-14
	ϵDEg	3.731440E-20	9.678765E-19	1.811234E-18	1.226054E+00	8.754569E+01	1.664753E+02
c18	DE-DBmax	8.320589E-25	5.257827E-24	9.154472E-24	1.631766E-17	2.901024E-04	9.250217E-04
	eABC	3.098470E-03	3.470230E+02	3.710760E+02	1.960850E-01	2.933360E+02	3.528430E+02
	Co-CLPSO	7.780400E-21	2.319200E-01	9.955900E-01	6.004700E-02	1.087700E+01	3.716100E+01
	SAMO-GA	4.359000E-17	1.053095E-02	1.540100E-02	2.818290E-01	7.535728E+00	1.051703E+01
	ECHT-ARMOR-DE	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00

Table B.25: Function values obtained by MLS-MODE, ADE , EPSDE, SAMODE, DE-Acr, CDELS and CDASA for CEC2011 for r01-r05

problem	Alg.	Best	Mean	Std.
r01	MLS-MODE	1.5444180E+04	1.5444180E+04	6.1846980E-07
	ADE	1.5445380E+04	1.5482070E+04	3.0473660E+01
	EPSDE	1.5400000E+04	1.5500000E+04	1.5500000E+01
	SAMODE	1.5444190E+04	1.5444230E+04	3.7333110E-02
	DE-Acr	1.5445000E+04	1.5446000E+04	8.4385000E-01
	CDELS	1.5446550E+04	1.5446999E+04	1.5449634E+04
	CDASA	1.5461000E+04	1.5511000E+04	4.2337000E+01
r02	MLS-MODE	1.8050910E+04	1.8104920E+04	3.7112340E+01
	ADE	1.8224720E+04	1.8550260E+04	1.4151470E+02
	EPSDE	1.8100000E+04	1.8100000E+04	4.3900000E+01
	SAMODE	1.8046840E+04	1.8185590E+04	1.0714080E+02
	DE-Acr	1.8028000E+04	1.8096000E+04	3.9913000E+01
	CDELS	1.9007720E+04	1.9120010E+04	1.9185830E+04
	CDASA	1.8942000E+04	1.9323000E+04	2.1267000E+02
r03	MLS-MODE	3.2797750E+04	3.2890930E+04	4.7772960E+01
	ADE	3.2744200E+04	3.2859400E+04	6.8413650E+01
	EPSDE	3.2600000E+04	3.2700000E+04	3.5900000E+01
	SAMODE	3.2721950E+04	3.2737580E+04	7.7779080E+00
	DE-Acr	3.2730000E+04	3.2790000E+04	3.0447000E+01
	CDELS	3.2928030E+04	3.2958290E+04	3.3031150E+04
	CDASA	3.2966000E+04	3.3181000E+04	1.0457000E+02
r04	MLS-MODE	1.2281680E+05	1.2448750E+05	7.2852280E+02
	ADE	1.2406900E+05	1.2553230E+05	8.3625570E+02
	EPSDE	1.2800000E+05	1.3100000E+05	2.4700000E+03
	SAMODE	1.2278753E+05	1.2409967E+05	8.0725815E+02
	DE-Acr	1.2313000E+05	1.2389000E+05	4.5430000E+02
	CDELS	1.3590694E+05	1.3771068E+05	1.4118462E+05
	CDASA	1.3242000E+05	1.4666000E+05	6.3435000E+03
r05	MLS-MODE	1.6769140E+06	1.7812770E+06	4.9848520E+04
	ADE	1.8906710E+06	1.9250980E+06	2.3429040E+04
	EPSDE	1.9100000E+06	1.9200000E+06	1.1800000E+04
	SAMODE	1.8891308E+06	1.9876716E+06	1.2556507E+05
	DE-Acr	1.7124000E+06	1.8386000E+06	4.5074000E+04
	CDELS	1.9209047E+06	1.9464404E+06	1.9573575E+06
	CDASA	1.8828000E+06	2.0375000E+06	2.4110000E+05

Table B.26: Function values obtained by MLS-MODE, ADE , EPSDE, SAMODE, DE-Acr, CDELS and CDASA for CEC2011 for r06-r10

problem	Alg.	Best	Mean	Std.
r06	MLS-MODE	4.8705940E+04	5.0643040E+04	1.2705080E+03
	ADE	5.0170600E+04	5.4181520E+04	4.8669640E+03
	EPSDE	5.1100000E+04	5.2200000E+04	7.2400000E+02
	SAMODE	5.0981400E+04	5.3278380E+04	2.0317090E+03
	DE-Acr	4.4847000E+04	4.6321000E+04	1.3962000E+03
	CDELS	6.8146400E+04	7.3806300E+04	8.0638850E+04
	CDASA	5.1210000E+04	5.2017000E+04	3.9028000E+02
r07	MLS-MODE	1.0544620E+06	1.0684960E+06	4.9241360E+03
	ADE	1.0785260E+06	1.0866840E+06	4.4556550E+03
	EPSDE	1.0600000E+06	1.0700000E+06	2.1300000E+03
	SAMODE	1.0675010E+06	1.0707990E+06	1.2727460E+03
	DE-Acr	1.0354000E+06	1.0490000E+06	2.9044000E+05
	CDELS	2.4228390E+06	2.4461757E+06	2.5662654E+06
	CDASA	1.2683000E+06	1.2717000E+06	1.8724000E+03
r08	MLS-MODE	9.1578410E+05	9.1618010E+05	2.6227390E+02
	ADE	9.2482150E+05	9.3060260E+05	3.5027700E+03
	EPSDE	9.3900000E+05	9.4300000E+05	2.6300000E+03
	SAMODE	9.2435370E+05	9.2580000E+05	6.7159130E+02
	DE-Acr	9.2280000E+05	9.2393000E+05	7.4822000E+02
	CDELS	9.8301754E+05	1.0188345E+06	1.2755968E+06
	CDASA	9.4142000E+05	9.4569000E+05	3.6946000E+03
r09	MLS-MODE	9.1970990E+05	9.2427600E+05	1.6122740E+03
	ADE	9.2832270E+05	9.9296430E+05	1.9642450E+05
	EPSDE	9.4000000E+05	9.9000000E+05	4.1400000E+04
	SAMODE	9.2432330E+05	9.2602220E+05	7.2979820E+02
	DE-Acr	9.2717000E+05	9.3001000E+05	1.6287000E+03
	CDELS	1.4593034E+06	1.5374957E+06	1.8722742E+06
	CDASA	1.1044000E+06	1.4012000E+06	1.7944000E+05
r10	MLS-MODE	9.2689750E+05	9.3485490E+05	4.4514900E+03
	ADE	9.2714540E+05	9.3052880E+05	3.0638130E+03
	EPSDE	9.3900000E+05	9.4300000E+05	2.6300000E+03
	SAMODE	9.2437627E+05	9.2584595E+05	7.6154770E+02
	DE-Acr	9.2303000E+05	9.2382000E+05	5.1463000E+02
	CDELS	1.0026198E+06	1.0295312E+06	1.2755968E+06
	CDASA	9.3967000E+05	9.4887000E+05	1.8056000E+04

Appendix C

Table C.1: Function values obtained by MJADE, MCoDE and MOXDE with and without R2S for CEC2006 and CEC2011

P	Criteria	MJADE-R2S	MJADE	MCoDE-R2S	MCoDE	MOXDE-R2S	MOXDE
g01	best	-1.5000E+01	-1.5000E+01	-1.5000E+01	-1.5000E+01	-1.5000E+01	-1.5000E+01
	Average	-1.5000E+01	-1.5000E+01	-1.5000E+01	-1.5000E+01	-1.4554E+01	-1.4898E+01
	St.d	0.0000E+00	0.0000E+00	0.0000E+00	1.0000E-10	8.8037E-01	5.0938E-01
g02	best	-8.0362E-01	-8.0253E-01	-8.0361E-01	-7.9982E-01	-8.0362E-01	-8.0362E-01
	Average	-8.0358E-01	-8.0072E-01	-8.0228E-01	-7.9335E-01	-7.9649E-01	-7.9682E-01
	St.d	7.6037E-05	9.3017E-04	3.6465E-03	3.4972E-03	1.0332E-02	8.5005E-03
g03	best	-1.0005E+00	-8.5371E-01	-1.0005E+00	-5.4855E-01	-1.0005E+00	-8.4378E-01
	Average	-1.0005E+00	-7.2806E-01	-1.0004E+00	-2.3569E-01	-1.0005E+00	-6.6064E-01
	St.d	0.0000E+00	8.2835E-02	6.5150E-04	1.2422E-01	0.0000E+00	1.0773E-01
g04	best	-3.0666E+04	-3.0666E+04	-3.0666E+04	-3.0666E+04	-3.0666E+04	-3.0666E+04
	Average	-3.0666E+04	-3.0666E+04	-3.0666E+04	-3.0666E+04	-3.0666E+04	-3.0666E+04
	St.d	0.0000E+00	4.0000E-12	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
g05	best	5.1265E+03	5.1265E+03	5.1265E+03	5.1384E+03	5.1265E+03	5.1265E+03
	Average	5.1265E+03	5.1859E+03	5.1265E+03	5.2228E+03	5.1265E+03	5.1614E+03
	St.d	0.0000E+00	5.4893E+01	7.4000E-09	8.4883E+01	0.0000E+00	6.3668E+01
g06	best	-6.9618E+03	-6.9618E+03	-6.9618E+03	-6.9618E+03	-6.9618E+03	-6.9618E+03
	Average	-6.9618E+03	-6.9618E+03	-6.9618E+03	-6.9618E+03	-6.9618E+03	-6.9618E+03
	St.d	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
g07	best	2.4306E+01	2.4306E+01	2.4307E+01	2.4312E+01	2.4306E+01	2.4306E+01
	Average	2.4306E+01	2.4306E+01	2.4308E+01	2.4315E+01	2.4306E+01	2.4306E+01
	St.d	0.0000E+00	0.0000E+00	4.0714E-04	2.3077E-03	0.0000E+00	0.0000E+00
g08	best	-9.5825E-02	-9.5825E-02	-9.5825E-02	-9.5825E-02	-9.5825E-02	-9.5825E-02
	Average	-9.5830E-02	-9.5825E-02	-9.5830E-02	-9.5830E-02	-9.5830E-02	-9.5830E-02
	St.d	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
g09	best	6.8063E+02	6.8063E+02	6.8063E+02	6.8063E+02	6.8063E+02	6.8063E+02
	Average	6.8063E+02	6.8063E+02	6.8063E+02	6.8063E+02	6.8063E+02	6.8063E+02
	St.d	0.0000E+00	0.0000E+00	6.0000E-10	2.8200E-08	0.0000E+00	0.0000E+00
g10	best	7.0492E+03	7.0492E+03	7.0502E+03	7.0535E+03	7.0492E+03	7.0492E+03
	Average	7.0492E+03	7.0492E+03	7.0510E+03	7.0579E+03	7.0506E+03	7.0492E+03
	St.d	1.0000E-10	2.0000E-12	3.8705E-01	2.2982E+00	6.9515E+00	0.0000E+00
g11	best	7.4990E-01	7.4990E-01	7.4990E-01	7.4990E-01	7.4990E-01	7.4990E-01
	Average	7.4990E-01	7.4991E-01	7.4990E-01	7.5234E-01	7.4990E-01	7.5067E-01
	St.d	0.0000E+00	2.8035E-05	0.0000E+00	9.7174E-03	0.0000E+00	2.8554E-03
g12	best	-1.0000E+00	-1.0000E+00	-1.0000E+00	-1.0000E+00	-1.0000E+00	-1.0000E+00
	Average	-1.0000E+00	-1.0000E+00	-1.0000E+00	-1.0000E+00	-1.0000E+00	-1.0000E+00
	St.d	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
g13	best	5.3942E-02	8.2929E-01	5.3942E-02	6.1239E-01	5.3942E-02	5.6604E-01
	Average	5.3940E-02	9.6981E-01	5.3940E-02	1.0862E+00	5.3940E-02	8.7747E-01
	St.d	0.0000E+00	4.2789E-02	1.7400E-08	6.2340E-01	0.0000E+00	1.4934E-01
g14	best	-4.7765E+01	-4.7765E+01	-4.7763E+01	-4.7764E+01	-4.7765E+01	-4.7765E+01
	Average	-4.7765E+01	-4.7765E+01	-4.7758E+01	-4.7763E+01	-4.7765E+01	-4.7765E+01
	St.d	0.0000E+00	0.0000E+00	4.1054E-03	8.5333E-04	0.0000E+00	0.0000E+00
g15	best	9.6172E+02	9.6172E+02	9.6172E+02	9.6172E+02	9.6172E+02	9.6172E+02
	Average	9.6172E+02	9.6215E+02	9.6172E+02	9.6230E+02	9.6172E+02	9.6201E+02
	St.d	0.0000E+00	7.3211E-01	0.0000E+00	7.5150E-01	0.0000E+00	7.0245E-01
g16	best	-1.9052E+00	-1.9052E+00	-1.9052E+00	-1.9052E+00	-1.9052E+00	-1.9052E+00
	Average	-1.9052E+00	-1.9052E+00	-1.9052E+00	-1.9052E+00	-1.9052E+00	-1.9052E+00
	St.d	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
g17	best	8.8535E+03	8.8561E+03	8.8535E+03	8.8786E+03	8.8535E+03	8.8605E+03
	Average	8.8565E+03	8.9243E+03	8.8556E+03	8.9457E+03	8.8565E+03	8.9256E+03
	St.d	1.4812E+01	2.9491E+01	2.2013E+00	2.3899E+01	1.4812E+01	2.3980E+01
g18	best	-8.6603E-01	-8.6603E-01	-8.6589E-01	-8.6386E-01	-8.6603E-01	-8.6603E-01
	Average	-8.3546E-01	-8.5838E-01	-8.6564E-01	-8.6160E-01	-8.2017E-01	-8.6603E-01
	St.d	7.1482E-02	3.8209E-02	1.7955E-04	1.5537E-03	8.3274E-02	0.0000E+00
g19	best	3.2656E+01	3.2656E+01	3.2684E+01	3.2951E+01	3.2656E+01	3.2656E+01
	Average	3.2656E+01	3.2656E+01	3.2717E+01	3.3045E+01	3.2656E+01	3.2656E+01
	St.d	0.0000E+00	0.0000E+00	1.4406E-02	7.7338E-02	0.0000E+00	0.0000E+00
g21	best	1.9372E+02	1.9372E+02	1.9372E+02	1.9381E+02	1.9372E+02	1.9372E+02
	Average	2.3040E+02	2.3040E+02	2.5060E+02	4.1038E+02	2.5076E+02	2.5243E+02
	St.d	6.0021E+01	6.0022E+01	5.8583E+01	2.7323E+02	6.4214E+01	6.2981E+01

g23	best	-4.0006E+02	-4.0006E+02	-3.8573E+02	-2.2555E+02	-4.0006E+02	-4.0006E+02
	Average	-4.0006E+02	-3.9741E+02	-3.5557E+02	-1.5979E+01	-3.8805E+02	-3.9936E+02
	St.d	6.0000E-10	4.2920E+00	2.0513E+01	1.2491E+02	6.0002E+01	2.0494E+00
g24	best	-5.5080E+00	-5.5080E+00	-5.5080E+00	-5.5080E+00	-5.5080E+00	-5.5080E+00
	Average	-5.5080E+00	-5.5080E+00	-5.5080E+00	-5.5080E+00	-5.5080E+00	-5.5080E+00
	St.d	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
r01	best	1.5444E+04	1.5444E+04	1.5444E+04	1.5447E+04	1.5444E+04	1.5444E+04
	Average	1.5444E+04	1.5449E+04	1.5444E+04	1.5454E+04	1.5444E+04	1.5451E+04
	St.d	1.0855E-06	8.9928E+00	1.1299E-02	5.8502E+00	6.2430E-07	1.1709E+01
r02	best	1.8028E+04	1.8075E+04	1.8265E+04	1.8477E+04	1.8035E+04	1.8019E+04
	Average	1.8120E+04	1.8146E+04	1.8474E+04	1.8629E+04	1.8118E+04	1.8113E+04
	St.d	5.0666E+01	4.4871E+01	9.1791E+01	6.3145E+01	5.2259E+01	4.8783E+01
r03	best	3.2692E+04	3.2994E+04	3.2761E+04	3.3032E+04	3.2692E+04	3.2982E+04
	Average	3.2694E+04	3.3056E+04	3.2816E+04	3.3151E+04	3.2696E+04	3.3089E+04
	St.d	9.5705E+00	4.6986E+01	3.4841E+01	7.0589E+01	1.3250E+01	4.5181E+01
r04	best	1.2145E+05	1.2543E+05	1.2375E+05	1.3098E+05	1.2145E+05	1.3662E+05
	Average	1.2157E+05	1.2789E+05	1.2407E+05	1.3325E+05	1.2157E+05	1.4391E+05
	St.d	8.5076E+01	9.5821E+02	2.1117E+02	1.2851E+03	8.5076E+01	4.0558E+03
r05	best	1.6649E+06	1.8692E+06	1.7537E+06	1.9291E+06	1.6649E+06	1.9014E+06
	Average	1.6685E+06	1.9044E+06	1.7703E+06	1.9459E+06	1.6685E+06	1.9444E+06
	St.d	3.4583E+03	2.5853E+04	6.2869E+03	8.4284E+03	3.4583E+03	2.1101E+04
r06	best	4.5730E+04	5.0962E+04	4.9265E+04	5.1100E+04	4.5730E+04	5.1042E+04
	Average	4.6586E+04	5.2136E+04	5.0198E+04	5.1919E+04	4.6586E+04	5.2014E+04
	St.d	5.5149E+02	5.7874E+02	5.6253E+02	4.2422E+02	5.5149E+02	4.8711E+02
r07	best	9.2271E+05	1.0724E+06	9.6681E+05	1.0711E+06	9.2271E+05	1.0714E+06
	Average	9.4068E+05	1.0749E+06	9.7703E+05	1.0733E+06	1.0751E+06	1.0751E+06
	St.d	8.0356E+03	1.5285E+03	5.3698E+03	1.4421E+03	8.0356E+03	2.0023E+03
r08	best	9.1692E+05	9.1569E+05	9.1779E+05	9.1811E+05	9.1692E+05	9.1567E+05
	Average	9.1733E+05	9.1600E+05	9.1862E+05	9.1868E+05	9.1733E+05	9.1602E+05
	St.d	3.8198E+02	2.8022E+02	3.5551E+02	2.9870E+02	3.8198E+02	2.5123E+02
r09	best	9.1749E+05	9.1993E+05	9.2211E+05	9.2994E+05	9.1749E+05	9.2222E+05
	Average	9.1837E+05	9.2095E+05	9.2433E+05	9.3437E+05	9.1837E+05	9.2841E+05
	St.d	5.2993E+02	5.8315E+02	1.0444E+03	2.1156E+03	5.2993E+02	8.3284E+03
r10	best	9.2737E+05	9.3680E+05	9.3245E+05	9.3977E+05	9.2737E+05	9.3870E+05
	Average	9.2926E+05	9.4338E+05	9.3444E+05	9.4356E+05	9.2926E+05	9.4472E+05
	St.d	9.6017E+02	3.4985E+03	1.1954E+03	2.3003E+03	9.6017E+02	3.3143E+03

Table C.2: Function values obtained by FRORI, SaDE and GAMPC with and without R2S for CEC2006 and CEC2011

P	Criteria	FRORI-R2S	FRORI	SaDE-R2S	SaDE	GAMPC-R2S	GAMPC
g01	best	-1.5000E+01	-1.5000E+01	-1.5000E+01	-1.5000E+01	-1.5000E+01	-1.5000E+01
	Average	-1.5000E+01	-1.5000E+01	-1.5000E+01	-1.5000E+01	-1.5000E+01	-1.5000E+01
	St.d	0.0000E+00	0.0000E+00	9.1400E-08	6.1700E-08	0.0000E+00	0.0000E+00
g02	best	-8.0362E-01	-8.0355E-01	-8.0333E-01	-8.0308E-01	-8.0362E-01	-8.0362E-01
	Average	-8.0282E-01	-8.0292E-01	-8.0095E-01	-7.9402E-01	-7.9206E-01	-7.8907E-01
	St.d	2.7803E-03	2.3488E-03	3.0965E-03	3.6435E-03	1.3263E-02	1.5853E-02
g03	best	-1.0005E+00	-9.8901E-01	-1.0005E+00	-6.6102E-01	-1.0005E+00	-1.0005E+00
	Average	-1.0005E+00	-9.6684E-01	-1.0005E+00	-5.2465E-01	-1.0005E+00	-4.4515E-01
	St.d	0.0000E+00	1.0652E-02	0.0000E+00	9.7210E-02	0.0000E+00	2.1450E-01
g04	Best	-3.0666E+04	-3.0666E+04	-3.0666E+04	-3.0666E+04	-3.0666E+04	-3.0666E+04
	Average	-3.0666E+04	-3.0666E+04	-3.0665E+04	-3.0666E+04	-3.0666E+04	-3.0666E+04
	St.D	0.0000E+00	0.0000E+00	8.7176E-02	1.3280E-07	4.0000E-12	4.0000E-12
g05	Best	5.1265E+03	5.1265E+03	5.1265E+03	5.1266E+03	5.1265E+03	5.1265E+03
	Average	5.1265E+03	5.1265E+03	5.1265E+03	5.1536E+03	5.1265E+03	5.2742E+03
	St.D	0.0000E+00	0.0000E+00	0.0000E+00	3.0343E+01	3.0000E-12	1.8027E+02
g06	Best	-6.9618E+03	-6.9618E+03	-6.9618E+03	-6.9618E+03	-7.5933E+03	-6.9618E+03
	Average	-6.9618E+03	-6.9618E+03	-6.9618E+03	-6.9618E+03	-6.9871E+03	-6.9618E+03
	St.D	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	1.2630E+02	0.0000E+00
g07	Best	2.4306E+01	2.4306E+01	2.4306E+01	2.4306E+01	2.4306E+01	2.4306E+01
	Average	2.4306E+01	2.4306E+01	2.4306E+01	2.4306E+01	2.4306E+01	2.4306E+01
	St.D	2.0000E-10	2.1069E-05	1.6700E-07	1.3695E-06	9.6280E-09	4.4000E-11
g08	Best	-9.5825E-02	-9.5825E-02	-9.5825E-02	-9.5825E-02	-9.5825E-02	-9.5825E-02
	Average	-9.5830E-02	-9.5830E-02	-9.5830E-02	-9.5830E-02	-9.5825E-02	-9.5825E-02
	St.D	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
g09	Best	6.8063E+02	6.8063E+02	6.8063E+02	6.8063E+02	6.8063E+02	6.8063E+02
	Average	6.8063E+02	6.8063E+02	6.8063E+02	6.8063E+02	6.8063E+02	6.8063E+02
	St.D	0.0000E+00	0.0000E+00	7.5932E-05	0.0000E+00	0.0000E+00	0.0000E+00
g10	Best	7.0492E+03	7.0493E+03	7.0492E+03	7.0493E+03	7.0492E+03	7.0492E+03
	Average	7.0492E+03	7.0493E+03	7.0493E+03	7.0493E+03	7.0492E+03	7.0492E+03
	St.D	7.0060E-07	7.4005E-03	1.7907E-02	3.3504E-02	3.3802E-07	3.0000E-12
g11	Best	7.4990E-01	7.4990E-01	7.4990E-01	7.4990E-01	7.4990E-01	7.4990E-01
	Average	7.4990E-01	7.4990E-01	7.4990E-01	7.5103E-01	7.4990E-01	8.0388E-01
	St.D	0.0000E+00	0.0000E+00	0.0000E+00	5.1181E-03	0.0000E+00	6.5354E-02
g12	Best	-1.0000E+00	-1.0000E+00	-1.0000E+00	-1.0000E+00	-1.0000E+00	-1.0000E+00
	Average	-1.0000E+00	-1.0000E+00	-1.0000E+00	-1.0000E+00	-1.0000E+00	-1.0000E+00
	St.D	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
g13	Best	5.3942E-02	5.3942E-02	5.3942E-02	8.0826E-01	5.3942E-02	5.3942E-02
	Average	5.3940E-02	5.3950E-02	5.3940E-02	9.7581E-01	5.3942E-02	3.5032E-01
	St.D	0.0000E+00	2.9450E-05	0.0000E+00	4.6030E-02	0.0000E+00	2.1094E-01
g14	Best	-4.7765E+01	-4.7765E+01	-4.7765E+01	-4.7765E+01	-4.7765E+01	-4.7765E+01
	Average	-4.7765E+01	-4.7765E+01	-4.7765E+01	-4.7765E+01	-4.7765E+01	-4.7765E+01
	St.D	1.4600E-08	4.9300E-07	1.7000E-09	0.0000E+00	1.0554E-08	1.2000E-11
g15	Best	9.6172E+02	9.6172E+02	9.6172E+02	9.6172E+02	9.6172E+02	9.6172E+02
	Average	9.6172E+02	9.6172E+02	9.6172E+02	9.6216E+02	9.6172E+02	9.6261E+02
	St.D	0.0000E+00	0.0000E+00	0.0000E+00	6.7043E-01	1.0000E-12	1.0765E+00
g16	Best	-1.9052E+00	-1.9052E+00	-1.9052E+00	-1.9052E+00	-1.9052E+00	-1.9052E+00
	Average	-1.9052E+00	-1.9052E+00	-1.9052E+00	-1.9052E+00	-1.9052E+00	-1.9052E+00
	St.D	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	5.0000E-09	0.0000E+00
g17	Best	8.8535E+03	8.8568E+03	8.8535E+03	8.9395E+03	8.8535E+03	8.8535E+03
	Average	8.8535E+03	8.8632E+03	8.8605E+03	8.9465E+03	8.8535E+03	8.9013E+03
	St.D	1.9800E-08	1.7868E+01	2.3900E+01	4.1999E+00	2.0000E-12	3.5988E+01
g18	Best	-8.6602E-01	-8.6579E-01	-8.6603E-01	-8.6603E-01	-8.6603E-01	-8.6603E-01
	Average	-8.6602E-01	-8.6527E-01	-8.2017E-01	-8.5073E-01	-8.5159E-01	-8.6603E-01
	St.D	3.8497E-06	4.3520E-04	8.3274E-02	5.2895E-02	6.3787E-02	0.0000E+00
g19	Best	3.2656E+01	3.2662E+01	3.2702E+01	3.2667E+01	3.2656E+01	3.2656E+01
	Average	3.2656E+01	3.2667E+01	3.3055E+01	3.2705E+01	3.2656E+01	3.2656E+01
	St.D	1.2707E-06	3.2269E-03	3.1541E-01	4.0410E-02	1.0498E-06	9.3340E-07
g21	Best	1.9372E+02	1.9372E+02	1.9373E+02	1.9377E+02	1.9372E+02	1.9372E+02
	Average	1.9896E+02	2.0104E+02	2.2945E+02	2.4011E+02	2.3040E+02	2.5659E+02
	St.D	2.6196E+01	2.6013E+01	2.9040E+01	2.4323E+01	6.0022E+01	6.6786E+01

g23	Best	-4.0005E+02	-3.8046E+02	-4.0004E+02	-3.9967E+02	-4.0006E+02	-4.0006E+02
	Average	-3.9961E+02	-3.5983E+02	-2.6421E+02	-3.7861E+02	-3.7605E+02	-3.8805E+02
	St.D	2.1845E+00	1.4478E+01	1.4262E+02	1.3756E+01	8.3075E+01	6.0002E+01
g24	Best	-5.5080E+00	-5.5080E+00	-5.5080E+00	-5.5080E+00	-5.5080E+00	-5.5080E+00
	Average	-5.5080E+00	-5.5080E+00	-5.5080E+00	-5.5080E+00	-5.5080E+00	-5.5080E+00
	St.D	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
r01	Best	1.5444E+04	1.5444E+04	1.5444E+04	1.5444E+04	1.5444E+04	1.5444E+04
	Average	1.5444E+04	1.5444E+04	1.5444E+04	1.5445E+04	1.5444E+04	1.5454E+04
	St.D	1.2969E-05	1.5014E+05	5.2395E-06	9.8578E-01	4.8713E-06	1.2148E+01
r02	Best	1.8538E+04	1.8677E+04	1.8075E+04	1.8075E+04	1.8058E+04	1.8045E+04
	Average	1.8632E+04	1.8597E+04	1.8106E+04	1.8111E+04	1.8191E+04	1.8158E+04
	St.D	4.6636E+01	1.5010E+05	3.0565E+01	4.1176E+01	1.2538E+02	6.3008E+01
r03	Best	3.2695E+04	3.3136E+04	3.2693E+04	3.2998E+04	3.2692E+04	3.2875E+04
	Average	3.2697E+04	3.2962E+04	3.2736E+04	3.3089E+04	3.2695E+04	3.3033E+04
	St.D	1.7909E+00	1.5011E+05	6.9628E+01	2.6260E+01	9.6839E+00	7.8394E+01
r04	Best	1.2282E+05	1.3883E+05	1.2182E+05	1.2780E+05	1.2158E+05	1.3831E+05
	Average	1.2357E+05	1.2872E+05	1.2218E+05	1.3010E+05	1.2277E+05	1.4757E+05
	St.D	5.1464E+02	1.5015E+05	2.4371E+02	1.8145E+03	1.8188E+03	5.6931E+03
r05	Best	1.3304E+06	1.8548E+06	1.6878E+06	1.8936E+06	1.7378E+06	1.9246E+06
	Average	1.3442E+06	1.7615E+06	1.6949E+06	1.9074E+06	1.7949E+06	1.9470E+06
	St.D	1.0115E+04	1.5015E+05	5.1015E+03	9.3519E+03	3.4442E+04	1.1278E+04
r06	Best	4.5242E+04	4.8891E+04	4.5516E+04	5.1343E+04	4.8082E+04	5.1001E+04
	Average	4.6649E+04	4.7759E+04	4.6488E+04	5.2176E+04	4.9279E+04	5.2543E+04
	St.D	7.1377E+02	1.5015E+05	4.2946E+02	4.9450E+02	5.4345E+02	5.9771E+02
r07	Best	7.0105E+05	9.5458E+05	1.0377E+06	1.0702E+06	1.0659E+06	1.0728E+06
	Average	7.1769E+05	9.3263E+05	1.0431E+06	1.0731E+06	1.0713E+06	1.0753E+06
	St.D	8.9975E+03	2.0020E+05	2.3511E+03	1.6103E+03	2.2509E+03	1.5009E+03
r08	Best	9.1639E+05	9.1755E+05	9.1662E+05	9.1601E+05	9.1598E+05	9.1583E+05
	Average	9.1670E+05	9.1712E+05	9.1690E+05	9.1630E+05	9.1682E+05	9.1619E+05
	St.D	2.2248E+02	1.5003E+05	2.3659E+02	1.3147E+02	4.1547E+02	2.3447E+02
r09	Best	9.1766E+05	9.2294E+05	9.1848E+05	9.2181E+05	9.1707E+05	9.1852E+05
	Average	9.1917E+05	9.2168E+05	9.1940E+05	9.2332E+05	9.2311E+05	9.2713E+05
	St.D	8.7250E+02	1.5008E+05	5.3074E+02	6.1649E+02	5.0612E+03	6.8705E+03
r10	best	9.2522E+05	9.2833E+05	9.2592E+05	9.3861E+05	9.2818E+05	9.3808E+05
	Average	9.2685E+05	9.2722E+05	9.2660E+05	9.4174E+05	9.3226E+05	9.4608E+05
	St.d	9.8155E+02	2.0020E+05	5.5665E+02	1.6099E+03	2.3048E+03	4.3681E+03

Table C.3: Function values obtained by MJADE, MCoDE and MOXDE with and without R2S for CEC2010 for 10D

Problem	Criteria	MJADE-R2S	MJADE	MCoDE-R2S	MCoDE	MOXDE-R2S	MOXDE
c01	best	-7.4731E-01	-7.4731E-01	-7.4730E-01	-7.4710E-01	-8.2188E-01	-7.4731E-01
	Average	-7.4529E-01	-7.4731E-01	-7.4540E-01	-7.4658E-01	-8.0572E-01	-7.4586E-01
	St.d	5.2900E-03	0.0000E+00	5.3771E-03	5.7556E-04	1.2450E-02	3.0000E-03
c02	best	-2.2777E+00	-6.7604E-01	-5.6383E-01	2.8896E-01	-2.2609E+00	9.0190E-01
	Average	-2.0103E+00	2.9705E+00	1.3785E+00	3.5073E+00	-2.2423E+00	3.3490E+00
	St.d	4.0944E-01	1.4316E+00	1.3243E+00	1.1931E+00	1.2930E-02	1.1986E+00
c03	best	0.0000E+00	0.0000E+00	2.5268E-04	1.6998E+08	2.8673E+01	7.6128E+02
	Average	0.0000E+00	0.0000E+00	7.1360E-01	9.1783E+13	7.3609E+01	1.2796E+14
	St.d	0.0000E+00	0.0000E+00	2.4573E+00	1.9906E+14	4.5431E+01	2.5638E+14
c04	best	-1.0000E-05	-1.0000E-05	5.5452E-04	2.2302E-02	1.6791E-01	2.4890E-04
	Average	9.9009E+00	6.6700E-02	1.0933E-03	7.5533E-02	1.3119E+00	5.2282E-04
	St.d	1.6262E+01	2.3520E-01	2.8754E-04	4.9194E-02	1.8201E+00	1.7000E-04
c05	best	-2.6391E+02	1.4904E+02	1.3972E+02	5.4644E+01	1.4623E+02	2.3473E+02
	Average	3.6320E+01	3.7530E+02	3.2406E+02	4.4302E+02	3.4648E+02	4.4720E+02
	St.d	1.4938E+02	1.1848E+02	1.1175E+02	1.1419E+02	8.7585E+01	8.7287E+01
c06	best	-5.8089E+02	7.9357E+01	-1.6323E+01	5.2111E+01	2.3282E+01	1.8963E+02
	Average	-7.0538E+01	3.4958E+02	3.2729E+02	4.5242E+02	3.6310E+02	4.6521E+02
	St.d	1.9221E+02	1.1561E+02	1.5422E+02	1.3325E+02	1.1472E+02	1.2585E+02
c07	best	0.0000E+00	0.0000E+00	4.5772E-01	1.8932E+00	2.7981E-07	6.2171E-08
	Average	0.0000E+00	0.0000E+00	1.1778E+00	4.2065E+00	1.5437E+07	1.5147E-06
	St.d	0.0000E+00	0.0000E+00	6.5928E-01	1.2638E+00	4.2689E+07	0.0000E+00
c08	best	0.0000E+00	0.0000E+00	5.1869E+00	5.1425E+01	3.3706E-01	8.7724E-04
	Average	9.6105E+00	8.8980E+00	1.6222E+05	1.6966E+02	1.2329E+03	8.3539E+00
	St.d	3.0696E+00	4.1981E+00	8.1098E+05	7.4939E+01	3.0737E+03	4.1679E+00
c09	best	0.0000E+00	1.2392E+12	2.0543E+11	2.7624E+12	3.6393E+10	4.2051E+12
	Average	3.7515E+11	7.6337E+12	5.1062E+12	1.1508E+13	2.1969E+12	1.3033E+13
	St.d	9.4435E+11	4.8842E+12	5.1497E+12	6.1047E+12	2.9369E+12	7.3894E+12
c10	best	0.0000E+00	9.1809E+11	7.9872E+11	1.3036E+12	1.7457E+11	2.9996E+12
	Average	4.4535E+10	8.8018E+12	3.2566E+12	1.0492E+13	3.4309E+12	9.9174E+12
	St.d	1.3280E+11	5.4561E+12	2.0610E+12	6.3057E+12	4.6492E+12	4.3962E+12
c11	best	-3.7220E-01	-1.5227E-03	-4.2034E+00	-1.9172E-01	-1.7421E-03	-1.5126E-03
	Average	1.4000E-01	-1.5200E-03	-1.1449E-01	-7.8724E-02	1.4168E-02	-1.4898E-03
	St.d	4.8259E-01	0.0000E+00	8.8884E-01	6.0429E-02	1.0500E-02	2.0000E-05
c12	best	-1.5441E+03	-3.0378E+02	-5.7007E+02	-3.0223E+02	-6.7960E+02	-8.2850E+01
	Average	-6.6746E+02	-1.2343E+01	-1.3407E+02	-1.1663E+01	-3.6917E+01	-5.1274E+00
	St.d	3.8228E+02	6.0717E+01	1.9344E+02	6.0724E+01	1.3870E+02	1.8555E+01
c13	best	-6.8429E+01	-6.8429E+01	-6.8428E+01	-6.7542E+01	-6.5033E+01	-6.8429E+01
	Average	-6.8315E+01	-6.8270E+01	-6.8405E+01	-6.5202E+01	-6.1279E+01	-6.6470E+01
	St.d	5.7018E-01	1.3735E-01	2.7052E-02	9.6944E-01	2.3874E+00	3.3536E+00
c14	best	0.0000E+00	9.8250E+08	3.1282E+08	6.7373E+11	1.2366E+01	7.8486E+11
	Average	1.7335E+08	7.9387E+10	3.1014E+11	5.0417E+12	2.6880E+06	5.4895E+12
	St.d	8.4496E+08	1.1947E+11	9.0084E+11	3.9508E+12	1.0460E+07	3.4284E+12
c15	best	4.8105E+11	2.5942E+12	2.0333E+13	1.0261E+13	4.0810E+13	1.1071E+13
	Average	4.4552E+13	3.5452E+13	9.4355E+13	6.2344E+13	9.4942E+13	6.6123E+13
	St.d	5.1189E+13	1.8185E+13	5.4568E+13	3.3763E+13	4.4549E+13	4.1323E+13
c16	best	0.0000E+00	8.7866E-01	2.8031E-02	9.8017E-01	0.0000E+00	9.9867E-01
	Average	0.0000E+00	9.9428E-01	6.2085E-01	1.0657E+00	2.7415E-02	1.0502E+00
	St.d	0.0000E+00	3.2130E-02	3.2090E-01	3.0494E-02	9.6750E-02	2.8970E-02
c17	best	0.0000E+00	5.6812E+01	6.3408E-02	2.2103E+02	2.2131E-01	2.0317E+02
	Average	1.5608E-01	2.3986E+02	2.2442E+01	6.7148E+02	4.0382E+01	5.7688E+02
	St.d	3.7892E-01	1.4535E+02	5.3793E+01	3.2379E+02	7.0749E+01	2.4107E+02
c18	best	0.0000E+00	3.3132E+03	9.6192E+02	4.5765E+03	7.1509E+01	5.2067E+03
	Average	5.4925E+02	1.0507E+04	4.7928E+03	1.4863E+04	6.0808E+03	1.3756E+04
	St.d	1.2888E+03	4.8276E+03	2.1971E+03	7.2381E+03	5.4365E+03	6.1163E+03

Table C.4: Function values obtained by FRORI, SaDE and GAMPC with and without R2S for CEC2010 for 10D

Problem	Criteria	FRORI-R2S	FRORI	SaDE-R2S	SaDE	GAMPC-R2S	GAMPC
c01	best	-7.4731E-01	-7.4731E-01	-7.4731E-01	-7.4731E-01	-7.4731E-01	-7.4731E-01
	Average	-7.4064E-01	-7.4677E-01	-7.4731E-01	-7.4721E-01	-7.2513E-01	-7.2513E-01
	St.d	4.2500E-03	1.8700E-03	0.0000E+00	1.7000E-04	2.4539E-02	2.4539E-02
c02	best	-2.2777E+00	-2.1564E+00	-4.7524E-01	-2.2419E+00	-2.2777E+00	-7.9110E-02
	Average	-2.0556E+00	-1.9329E+00	3.0074E+00	-1.9582E+00	-1.8726E+00	2.8027E+00
	St.d	1.4048E-01	5.3027E-01	1.1222E+00	3.9838E-01	1.3383E+00	1.3912E+00
c03	best	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
	Average	0.0000E+00	0.0000E+00	1.6193E+12	2.5182E+01	3.5502E-01	1.1897E+14
	St.d	0.0000E+00	0.0000E+00	2.4424E+12	1.3839E+01	1.7751E+00	2.1466E+14
c04	best	-1.0000E-05	-1.0000E-05	-9.9999E-06	-9.4983E-06	-1.0000E-05	-1.0000E-05
	Average	-1.0000E-05	-1.0000E-05	2.7134E-02	1.2432E+00	-1.0000E-05	9.3835E-02
	St.d	0.0000E+00	0.0000E+00	1.3568E-01	2.8164E+00	0.0000E+00	2.6450E-01
c05	best	-4.8361E+02	-4.8361E+02	2.1842E+02	-2.4591E+02	4.0151E+01	3.8301E+01
	Average	-4.8361E+02	-4.8361E+02	3.9346E+02	-1.2556E+02	3.6165E+02	3.6592E+02
	St.d	0.0000E+00	1.0000E-05	9.2251E+01	1.6456E+02	1.6840E+02	1.5055E+02
c06	best	-5.7866E+02	-5.7866E+02	1.0678E+02	-4.9368E+02	-5.7866E+02	1.3171E+02
	Average	-5.7729E+02	-5.7866E+02	3.9100E+02	-3.5756E+02	3.0143E+02	3.9697E+02
	St.d	4.0979E-01	8.4000E-04	1.2231E+02	7.8839E+01	3.2138E+02	1.3605E+02
c07	best	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
	Average	0.0000E+00	0.0000E+00	2.6052E-01	1.6789E-01	4.7839E-01	4.7839E-01
	St.d	0.0000E+00	0.0000E+00	5.4954E-01	6.0840E-01	1.3222E+00	1.3222E+00
c08	best	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
	Average	8.3832E+00	8.2271E+00	3.5013E+00	2.2048E+01	8.9702E+00	8.9702E+00
	St.d	4.0584E+00	4.7210E+00	5.2092E+00	5.7192E+01	7.9635E+00	7.9635E+00
c09	best	0.0000E+00	0.0000E+00	2.5333E+11	1.9653E-08	0.0000E+00	1.3770E+12
	Average	1.7633E-01	2.7320E+01	7.8619E+12	9.5034E+01	2.6208E+12	9.1879E+12
	St.d	8.8163E-01	3.7799E+01	4.7386E+12	1.0057E+02	3.9740E+12	6.6750E+12
c10	best	0.0000E+00	0.0000E+00	3.8698E+11	2.7110E+00	0.0000E+00	1.5984E+12
	Average	0.0000E+00	4.0057E+01	6.3719E+12	8.1224E+01	3.5303E+12	1.0222E+13
	St.d	0.0000E+00	8.3452E+00	4.6609E+12	6.7192E+01	4.5032E+12	4.3672E+12
c11	best	-1.5227E-03	-2.9449E-01	-2.6912E-02	-3.3588E-01	-8.7342E-02	-8.7342E-02
	Average	-7.0000E-05	-1.3240E-02	3.9768E-03	1.4292E-01	4.3885E-01	-4.9555E-03
	St.d	7.2600E-03	5.8590E-02	1.3150E-02	6.8907E-01	2.2557E+00	1.7164E-02
c12	best	-8.8704E+02	-5.7009E+02	-1.0639E+01	-8.8704E+02	-8.8704E+02	-4.2313E+02
	Average	-8.8220E+02	-3.5131E+02	1.2667E-01	-7.0456E+02	-7.3613E+02	-6.0586E+01
	St.d	1.4285E+01	2.3993E+02	3.4706E+00	2.8457E+02	3.0879E+02	1.4010E+02
c13	best	-6.8429E+01	-6.8429E+01	-6.8417E+01	-6.8429E+01	-6.8429E+01	-6.8429E+01
	Average	-6.7343E+01	-6.8429E+01	-6.7815E+01	-6.8376E+01	-6.3711E+01	-6.3711E+01
	St.d	5.3836E+00	0.0000E+00	2.2020E-01	1.2322E-01	2.0241E+00	2.0241E+00
c14	best	0.0000E+00	0.0000E+00	1.2106E+11	2.4574E-03	0.0000E+00	0.0000E+00
	Average	0.0000E+00	0.0000E+00	5.7268E+11	6.0659E-01	2.4363E+09	2.4363E+09
	St.d	0.0000E+00	0.0000E+00	3.7262E+11	6.4674E-01	1.2181E+10	1.2181E+10
c15	best	0.0000E+00	0.0000E+00	1.5195E+12	0.0000E+00	5.3223E+12	5.3223E+12
	Average	3.0951E+00	3.0855E+00	1.8005E+13	1.0615E+00	3.3224E+13	3.3224E+13
	St.d	1.3527E+00	1.3744E+00	1.4528E+13	1.7442E+00	1.7034E+13	1.7034E+13
c16	best	0.0000E+00	0.0000E+00	1.0088E+00	0.0000E+00	0.0000E+00	1.1348E-02
	Average	0.0000E+00	1.7380E-02	1.0342E+00	0.0000E+00	1.5478E-01	6.5004E-01
	St.d	0.0000E+00	4.3340E-02	1.3250E-02	0.0000E+00	1.7613E-01	4.0442E-01
c17	best	0.0000E+00	0.0000E+00	7.2933E+01	0.0000E+00	0.0000E+00	2.8531E+01
	Average	2.4100E-03	1.1365E-01	3.4038E+02	1.6567E-01	4.1944E-01	4.9083E+02
	St.d	8.2000E-03	2.6848E-01	1.9447E+02	3.1660E-01	9.9827E-01	2.9329E+02
c18	best	0.0000E+00	0.0000E+00	1.5014E+03	0.0000E+00	0.0000E+00	2.0741E+03
	Average	0.0000E+00	0.0000E+00	9.2596E+03	1.7442E+00	4.8374E+03	1.1310E+04
	St.d	0.0000E+00	0.0000E+00	4.7018E+03	3.6312E+00	7.0530E+03	5.1203E+03

Table C.5: Function values obtained by MJADE, MCoDE and MOXDE with and without R2S for CEC2010 for 30D

Problem	Criteria	MJADE-R2S	MJADE	MCoDE-R2S	MCoDE	MOXDE-R2S	MOXDE
c01	best	-8.2188E-01	-8.2028E-01	-8.2188E-01	-8.1188E-01	-8.2188E-01	-8.2186E-01
	Average	-8.1884E-01	-8.1851E-01	-8.1903E-01	-8.0269E-01	-8.0572E-01	-8.1594E-01
	St.d	3.3000E-03	1.0600E-03	2.7004E-03	5.0923E-03	1.2450E-02	7.7000E-03
c02	best	-2.2807E+00	1.9259E+00	-1.8655E+00	2.9092E+00	-2.2609E+00	2.3162E+00
	Average	-2.2702E+00	3.0030E+00	-1.2785E+00	3.9574E+00	-2.2423E+00	4.4347E+00
	St.d	8.4000E-03	5.0236E-01	3.0969E-01	4.2029E-01	1.2930E-02	8.3621E-01
c03	best	0.0000E+00	1.5951E+11	3.0113E+01	3.9044E+09	2.8673E+01	1.0685E+11
	Average	1.0000E-04	1.3018E+13	1.1853E+02	2.0792E+13	7.3609E+01	1.0177E+15
	St.d	4.9000E-04	1.3674E+13	4.0758E+01	4.2487E+13	4.5431E+01	9.6758E+14
c04	best	-1.6641E+00	-3.3333E-06	4.2779E-02	1.7617E-02	1.6791E-01	1.9145E-02
	Average	8.9271E+00	1.8334E+00	1.3748E+00	1.4696E-01	1.3119E+00	4.2161E-01
	St.d	1.0413E+01	6.0058E+00	2.1426E+00	1.2475E-01	1.8201E+00	2.8696E-01
c05	best	-1.7385E+02	3.7216E+02	2.5062E+02	4.7150E+02	1.4623E+02	3.3424E+02
	Average	1.9452E+02	5.1420E+02	4.2385E+02	5.3463E+02	3.4648E+02	5.2850E+02
	St.d	1.5427E+02	4.9667E+01	7.0301E+01	3.0407E+01	8.7585E+01	5.6977E+01
c06	best	-1.9382E+02	4.0309E+02	3.5733E+02	5.1301E+02	2.3282E+01	3.5902E+02
	Average	1.1422E+02	5.2287E+02	4.8261E+02	5.6380E+02	3.6310E+02	5.5232E+02
	St.d	1.9705E+02	6.0737E+01	6.1308E+01	2.5159E+01	1.1472E+02	5.2312E+01
c07	best	0.0000E+00	0.0000E+00	1.9633E+01	2.4942E+01	2.7981E-07	6.1750E+00
	Average	4.7839E-01	4.7839E-01	2.0373E+07	3.1394E+01	1.5437E+07	1.6185E+01
	St.d	1.3222E+00	1.3222E+00	7.0728E+07	1.3626E+01	4.2689E+07	1.9577E+01
c08	best	0.0000E+00	0.0000E+00	2.4770E+01	8.6880E+03	3.3706E-01	1.8225E+01
	Average	2.2044E+02	5.9815E+01	1.9627E+07	2.7205E+04	1.2329E+03	1.2475E+02
	St.d	4.2871E+02	1.7672E+02	7.7824E+07	1.2878E+04	3.0737E+03	3.1195E+02
c09	best	0.0000E+00	9.8877E+12	1.5969E+12	1.3067E+13	3.6393E+10	1.6787E+13
	Average	7.9184E+11	2.8973E+13	6.1463E+12	4.1919E+13	2.1969E+12	4.0761E+13
	St.d	1.9356E+12	1.2011E+13	4.5959E+12	1.4552E+13	2.9369E+12	1.3102E+13
c10	best	3.1309E+01	1.1382E+13	2.7680E+12	2.2768E+13	1.7457E+11	1.4301E+13
	Average	2.1642E+12	2.4775E+13	9.0669E+12	4.0089E+13	3.4309E+12	3.8270E+13
	St.d	5.3144E+12	8.3408E+12	6.3115E+12	1.2379E+13	4.6492E+12	1.1574E+13
c11	best	-3.0787E-04	-3.9234E-04	-2.6345E-02	-3.9646E-02	-1.7421E-03	-2.0838E-03
	Average	4.1150E-02	1.1300E-03	4.8227E-02	-1.7251E-02	1.4168E-02	3.1271E-02
	St.d	2.0760E-02	5.2800E-03	7.8317E-02	1.2085E-02	1.0500E-02	1.4190E-02
c12	best	-3.3682E+03	-1.9926E-01	-9.0392E+02	-1.0660E+02	-6.7960E+02	-1.9925E-01
	Average	-1.7505E+03	-1.9926E-01	-5.1517E+01	-1.0725E+00	-3.6917E+01	5.8771E+00
	St.d	1.1057E+03	0.0000E+00	2.0168E+02	2.4652E+01	1.3870E+02	2.4607E+01
c13	best	-6.8429E+01	-6.6541E+01	-6.8429E+01	-5.7684E+01	-6.5033E+01	-6.5611E+01
	Average	-6.5867E+01	-6.5341E+01	-6.8390E+01	-5.4370E+01	-6.1279E+01	-6.3387E+01
	St.d	1.6725E+00	8.5065E-01	1.5502E-01	1.2987E+00	2.3874E+00	1.2132E+00
c14	best	0.0000E+00	0.0000E+00	2.6332E+07	1.2532E+13	1.2366E+01	7.8839E+12
	Average	9.8075E+05	1.0958E+09	8.5911E+09	3.5324E+13	2.6880E+06	2.5483E+13
	St.d	3.4831E+06	2.1367E+09	1.6304E+10	1.1312E+13	1.0460E+07	1.0502E+13
c15	best	2.1603E+01	3.5006E+13	2.1630E+13	9.0749E+13	4.0810E+13	1.2240E+14
	Average	9.8781E+13	1.0381E+14	1.6530E+14	2.3371E+14	9.4942E+13	2.2995E+14
	St.d	5.7336E+13	4.3934E+13	1.1241E+14	6.7049E+13	4.4549E+13	5.2312E+13
c16	best	0.0000E+00	1.0456E+00	7.8000E-11	1.1091E+00	0.0000E+00	1.0799E+00
	Average	0.0000E+00	1.0734E+00	4.3310E-09	1.1790E+00	2.7415E-02	1.1292E+00
	St.d	0.0000E+00	1.5060E-02	1.9608E-08	3.4530E-02	9.6750E-02	3.2360E-02
c17	best	0.0000E+00	3.7835E+02	1.2189E-02	1.1301E+03	2.2131E-01	1.1314E+03
	Average	2.5243E-01	7.3509E+02	7.5812E+00	1.9347E+03	4.0382E+01	1.7881E+03
	St.d	2.9517E-01	2.3478E+02	1.5809E+01	5.1045E+02	7.0749E+01	3.3231E+02
c18	best	0.0000E+00	6.7264E+03	3.6159E+03	2.4254E+04	7.1509E+01	1.9401E+04
	Average	2.4315E+03	1.8047E+04	1.2626E+04	3.4018E+04	6.0808E+03	3.4594E+04
	St.d	4.7555E+03	6.1590E+03	6.1116E+03	4.9550E+03	5.4365E+03	8.9789E+03

Table C.6: Function values obtained by FRORI, SaDE and GAMPC with and without R2S for CEC2010 for 30D

Problem	Criteria	FRORI-R2S	FRORI	SaDE-R2S	SaDE	GAMPC-R2S	GAMPC
c01	best	-8.2188E-01	-8.2186E-01	-8.2096E-01	-8.1804E-01	-8.2188E-01	-8.2188E-01
	Average	-8.2010E-01	-8.1954E-01	-8.1755E-01	-8.1793E-01	-7.9294E-01	-7.9193E-01
	St.d	2.3400E-03	2.7900E-03	2.1400E-03	4.0000E-05	1.9906E-02	1.9412E-02
c02	best	-2.0971E+00	-2.0540E+00	-2.2077E+00	3.3416E+00	-2.2726E+00	3.4082E+00
	Average	-2.0265E+00	-2.0015E+00	-2.1019E+00	3.9279E+00	-2.2519E+00	4.3779E+00
	St.d	4.0050E-02	4.1710E-02	4.1590E-02	3.5800E-01	3.9014E-02	5.2499E-01
c03	best	0.0000E+00	0.0000E+00	3.7275E+01	2.6711E+12	0.0000E+00	2.2513E+07
	Average	0.0000E+00	2.7527E+01	5.3481E+01	1.3364E+13	1.7001E+00	4.9244E+13
	St.d	0.0000E+00	5.7347E+00	9.9579E+00	7.4735E+12	8.5007E+00	6.8614E+13
c04	best	-3.0155E-06	-3.3333E-06	6.9984E+00	1.9672E-01	2.3802E-03	2.4993E-03
	Average	0.0000E+00	0.0000E+00	3.5903E+01	1.5468E+00	6.1171E-01	6.8649E+00
	St.d	1.0000E-05	0.0000E+00	1.7274E+01	2.2462E+00	6.7104E-01	1.2668E+01
c05	best	-4.8361E+02	-4.8305E+02	-2.0451E+02	3.4584E+02	4.5441E+02	3.8579E+02
	Average	-4.8361E+02	-4.8216E+02	-1.4604E+02	4.6744E+02	5.2808E+02	5.0918E+02
	St.d	0.0000E+00	1.0092E+00	1.3875E+02	6.2522E+01	3.7056E+01	5.6067E+01
c06	best	-5.3064E+02	-5.2962E+02	-1.9106E+02	4.2758E+02	4.5949E+02	3.8411E+02
	Average	-5.3064E+02	-5.2849E+02	-1.0098E+02	5.3579E+02	5.5631E+02	5.5213E+02
	St.d	2.0700E-03	1.0250E+00	3.3722E+01	4.8312E+01	4.0248E+01	4.9792E+01
c07	best	0.0000E+00	0.0000E+00	8.1494E+00	6.2755E+00	0.0000E+00	0.0000E+00
	Average	1.1092E+07	0.0000E+00	3.1043E+01	2.4564E+01	4.7839E-01	4.7839E-01
	St.d	3.8390E+07	0.0000E+00	2.2553E+01	1.8751E+01	1.3222E+00	1.3222E+00
c08	best	0.0000E+00	0.0000E+00	1.6065E+01	5.5460E+00	0.0000E+00	0.0000E+00
	Average	1.1759E+01	3.9280E+00	2.6311E+02	1.7532E+02	3.5933E+02	3.5599E+02
	St.d	2.7838E+01	1.9640E+01	7.7430E+02	3.8144E+02	7.5827E+02	7.5970E+02
c09	best	0.0000E+00	0.0000E+00	6.9057E+01	5.2901E+12	0.0000E+00	1.7162E+13
	Average	0.0000E+00	2.0311E+02	1.0789E+06	2.3023E+13	1.6482E+13	3.6549E+13
	St.d	0.0000E+00	8.3727E+02	2.5796E+06	1.1961E+13	1.5558E+13	1.1175E+13
c10	best	6.5470E+00	3.1310E+01	7.1281E+01	1.8188E+12	0.0000E+00	1.5179E+13
	Average	6.5470E+00	3.1321E+01	2.1117E+06	2.3065E+13	1.4739E+13	3.5793E+13
	St.d	1.0000E-05	1.4550E-02	4.5796E+06	1.1001E+13	1.6596E+13	1.0174E+13
c11	best	-1.6721E-02	-3.9234E-04	-1.6287E+00	-4.5147E-02	-3.9216E-04	-3.9234E-04
	Average	1.9930E-02	-3.9000E-04	-1.8936E-01	1.0804E-02	1.9630E-03	4.9454E-03
	St.d	1.3470E-02	0.0000E+00	7.0524E-01	4.1880E-02	6.3287E-03	8.7359E-03
c12	best	-1.5280E+02	-1.2131E+02	-9.1848E+02	-2.2038E+02	-5.7192E+02	-1.9926E-01
	Average	-1.0120E+01	-5.0438E+00	-4.4742E+02	-1.5673E+00	-3.7261E+01	-1.9926E-01
	St.d	3.0764E+01	2.4223E+01	4.1750E+02	4.8439E+01	1.6099E+02	1.1688E-07
c13	best	-6.8428E+01	-6.8428E+01	-6.8429E+01	-6.8186E+01	-6.4908E+01	-6.4908E+01
	Average	-6.6472E+01	-6.8389E+01	-6.8291E+01	-6.7756E+01	-6.2380E+01	-6.2380E+01
	St.d	8.5631E+00	1.0834E-01	6.7939E-01	1.9539E-01	1.5458E+00	1.5458E+00
c14	best	0.0000E+00	0.0000E+00	1.1754E+00	1.8942E+01	0.0000E+00	0.0000E+00
	Average	1.5946E-01	1.5946E-01	3.1266E+01	2.1450E+03	1.9795E+05	1.9795E+05
	St.d	7.9732E-01	7.9732E-01	2.6813E+01	1.0215E+04	7.7267E+05	7.7267E+05
c15	best	2.1603E+01	2.1603E+01	4.4942E+00	2.9457E+13	7.3110E+13	7.3110E+13
	Average	2.1604E+01	2.1604E+01	4.2988E+01	6.5947E+13	1.4010E+14	1.4010E+14
	St.d	9.0000E-05	9.0000E-05	3.0610E+01	2.4612E+13	3.3911E+13	3.3911E+13
c16	best	0.0000E+00	0.0000E+00	0.0000E+00	1.0927E+00	0.0000E+00	1.0838E+00
	Average	0.0000E+00	5.3000E-04	6.0200E-10	1.1309E+00	4.4158E-02	1.1410E+00
	St.d	0.0000E+00	2.6300E-03	0.0000E+00	2.5650E-02	1.8440E-01	2.8925E-02
c17	best	0.0000E+00	0.0000E+00	6.9643E-03	5.3205E+02	1.8351E+00	9.6259E+02
	Average	6.9998E-01	1.1333E-01	5.8986E-01	1.3146E+03	4.2820E+02	1.8612E+03
	St.d	1.6847E+00	2.7722E-01	9.4236E-01	5.3577E+02	5.1177E+02	4.7196E+02
c18	best	0.0000E+00	0.0000E+00	2.5852E-01	1.3728E+04	2.2600E-10	1.9605E+04
	Average	0.0000E+00	6.4572E-01	3.2206E+01	2.0953E+04	1.6110E+04	2.7162E+04
	St.d	0.0000E+00	1.9996E+00	3.8816E+01	4.9146E+03	1.1335E+04	4.8482E+03

Table C.7: Function values obtained by proposed algorithm with different CS values of $CS = 1$, 25, 50, 75 and 100 generations

	CS=1		CS=25		CS=50		CS=75		CS=100	
	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
g01	-1.476563E+01	4.784160E-01	-1.50000E+01	0.00000E+00	-1.50000E+01	0.00000E+00	-1.50000E+01	0.00000E+00	-1.50000E+01	0.00000E+00
g02	-8.034874E-01	4.503080E-04	-8.03584E-01	7.46577E-05	-8.03584E-01	7.46577E-05	-8.03579E-01	7.57183E-05	-8.03579E-01	7.57183E-05
g03	-1.000500E+00	0.000000E+00	-1.00050E+00	0.00000E+00	-1.00050E+00	0.00000E+00	-1.00050E+00	0.00000E+00	-1.00050E+00	0.00000E+00
g04	-3.066554E+04	4.000000E-12	-3.06655E+04	4.00000E-12	-3.06655E+04	4.00000E-12	-3.06655E+04	4.00000E-12	-3.06655E+04	4.00000E-12
g05	5.151055E+03	2.221757E+01	5.12650E+03	3.00000E-12	5.12650E+03	3.00000E-12	5.12650E+03	3.00000E-12	5.12650E+03	3.00000E-12
g06	-6.961814E+03	0.000000E+00	-6.96181E+03	0.00000E+00	-6.96181E+03	0.00000E+00	-6.96181E+03	0.00000E+00	-6.96181E+03	0.00000E+00
g07	2.464802E+01	1.709034E+00	2.43062E+01	6.49500E-09	2.43062E+01	0.00000E+00	2.43062E+01	0.00000E+00	2.43062E+01	6.00000E-12
g08	-9.582504E-02	0.000000E+00	-9.58250E-02	0.00000E+00	-9.58250E-02	0.00000E+00	-9.58250E-02	0.00000E+00	-9.58250E-02	0.00000E+00
g09	6.806301E+02	0.000000E+00	6.80630E+02	0.00000E+00	6.80630E+02	0.00000E+00	6.80630E+02	0.00000E+00	6.80630E+02	0.00000E+00
g10	7.049248E+03	1.330000E-10	7.04925E+03	3.00000E-12	7.04925E+03	3.00000E-12	7.04925E+03	4.00000E-12	7.04925E+03	3.00000E-12
g11	7.499000E-01	0.000000E+00	7.49900E-01	0.00000E+00	7.49900E-01	0.00000E+00	7.49900E-01	0.00000E+00	7.49900E-01	0.00000E+00
g12	-9.996006E-01	3.056963E-04	-1.00000E+00	0.00000E+00	-1.00000E+00	0.00000E+00	-1.00000E+00	0.00000E+00	-1.00000E+00	0.00000E+00
g13	5.394151E-02	0.000000E+00	5.39415E-02	0.00000E+00	5.39415E-02	0.00000E+00	5.39415E-02	0.00000E+00	5.39415E-02	0.00000E+00
g14	-4.776489E+01	0.000000E+00	-4.77649E+01	0.00000E+00	-4.77649E+01	0.00000E+00	-4.77649E+01	0.00000E+00	-4.77649E+01	0.00000E+00
g15	9.617150E+02	1.000000E-12	9.61715E+02	1.00000E-12	9.61715E+02	1.00000E-12	9.61715E+02	1.00000E-12	9.61715E+02	1.00000E-12
g16	-1.905155E+00	0.000000E+00	-1.90516E+00	0.00000E+00	-1.90516E+00	0.00000E+00	-1.90516E+00	0.00000E+00	-1.90516E+00	0.00000E+00
g17	8.860013E+03	2.251566E+01	8.86001E+03	2.25157E+01	8.86001E+03	2.25157E+01	8.86001E+03	2.25157E+01	8.86001E+03	2.25157E+01
g18	-7.325384E-01	1.425095E-01	-7.97706E-01	1.01180E-01	-7.49903E-01	1.71863E-01	-7.83775E-01	1.21678E-01	-7.60720E-01	1.57780E-01
g19	3.265559E+01	5.890000E-10	3.26556E+01	7.00000E-11	3.26556E+01	9.60000E-11	3.26556E+01	9.60000E-11	3.26556E+01	9.90000E-11
g21	2.303989E+02	6.002149E+01	2.30399E+02	6.00215E+01	2.30399E+02	6.00215E+01	2.30399E+02	6.00215E+01	2.30399E+02	6.00215E+01
g23	-4.000551E+02	3.060520E-07	-4.00055E+02	1.27000E-10	-4.00055E+02	1.27000E-10	-4.00055E+02	1.27000E-10	-4.00055E+02	1.27000E-10
g24	-5.508013E+00	0.000000E+00	-5.50801E+00	0.00000E+00	-5.50801E+00	0.00000E+00	-5.50801E+00	0.00000E+00	-5.50801E+00	0.00000E+00

Table C.8: Function values obtained by proposed algorithm with different FEs_c values of $FEs_c = 0.3$, 0.5, 0.7, and 0.9 FEs_{MAX}

	$FEs_c = 0.3FEs_{MAX}$		$FEs_c = 0.5FEs_{MAX}$		$FEs_c = 0.7FEs_{MAX}$		$FEs_c = 0.9FEs_{MAX}$	
	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
g01	-1.500000E+01	0.000000E+00	-1.500000E+01	0.000000E+00	-1.500000E+01	1.000000E-12	-1.499280E+01	1.122375E-02
g02	-8.026617E-01	3.386871E-03	-8.035836E-01	7.465768E-05	-8.035844E-01	7.493343E-05	-8.035832E-01	7.448197E-05
g03	-1.000500E+00	0.000000E+00	-1.000500E+00	0.000000E+00	-1.000500E+00	0.000000E+00	-1.000500E+00	5.921140E-07
g04	-3.066554E+04	4.000000E-12	-3.066554E+04	4.000000E-12	-3.066554E+04	4.000000E-12	-3.066554E+04	1.249165E-03
g05	5.126497E+03	3.000000E-12	5.126497E+03	3.000000E-12	5.126497E+03	3.000000E-12	5.126656E+03	5.260210E-01
g06	-6.961814E+03	0.000000E+00	-6.961814E+03	0.000000E+00	-6.961814E+03	0.000000E+00	-6.949107E+03	3.199674E+01
g07	2.431032E+01	1.422335E-02	2.430621E+01	6.495000E-09	2.430621E+01	9.000000E-12	2.431964E+01	2.392446E-02
g08	-9.582504E-02	0.000000E+00	-9.582504E-02	0.000000E+00	-9.582504E-02	0.000000E+00	-9.582504E-02	0.000000E+00
g09	6.806322E+02	1.049121E-02	6.806301E+02	0.000000E+00	6.806301E+02	0.000000E+00	6.806411E+02	1.729010E-02
g10	7.049248E+03	4.000000E-12	7.049248E+03	3.000000E-12	7.049248E+03	4.240000E-10	7.054611E+03	2.919199E+00
g11	7.499000E-01	0.000000E+00	7.499000E-01	0.000000E+00	7.499000E-01	0.000000E+00	7.499000E-01	0.000000E+00
g12	-1.000000E+00	0.000000E+00	-1.000000E+00	0.000000E+00	-1.000000E+00	0.000000E+00	-1.000000E+00	0.000000E+00
g13	5.394151E-02	0.000000E+00	5.394151E-02	0.000000E+00	5.394151E-02	0.000000E+00	5.396878E-02	5.528193E-05
g14	-4.776489E+01	0.000000E+00	-4.776489E+01	0.000000E+00	-4.776489E+01	0.000000E+00	-4.776151E+01	1.808974E-03
g15	9.617150E+02	1.000000E-12	9.617150E+02	1.000000E-12	9.617150E+02	1.000000E-12	9.617150E+02	1.481909E-06
g16	-1.905155E+00	0.000000E+00	-1.905155E+00	0.000000E+00	-1.905155E+00	0.000000E+00	-1.905141E+00	1.580695E-05
g17	8.867267E+03	3.120840E+01	8.860013E+03	2.251566E+01	8.856496E+03	1.481157E+01	8.860543E+03	2.439862E+01
g18	-7.799195E-01	9.734265E-02	-7.977059E-01	1.011804E-01	-7.520819E-01	2.042593E-01	-5.656166E-01	2.320782E-01
g19	3.265565E+01	2.305376E-04	3.265559E+01	7.000000E-11	3.265559E+01	1.400000E-11	3.266023E+01	7.628188E-03
g21	2.310374E+02	5.969931E+01	2.303989E+02	6.002149E+01	2.303985E+02	6.002181E+01	2.303986E+02	6.002188E+01
g23	-3.965959E+02	9.054982E+00	-4.000551E+02	1.270000E-10	-4.000551E+02	1.342200E-08	-3.996159E+02	3.680200E-01
g24	-5.508013E+00	0.000000E+00	-5.508013E+00	0.000000E+00	-5.508013E+00	0.000000E+00	-5.508013E+00	4.760000E-10

Appendix D

Table D.1: Function values obtained by Var1, Var2 and Var3

Problem		Var1	Var2	Var3
c01	Best	-7.4731036E-01	-7.4731036E-01	-7.4731036E-01
	Mean	-7.4731036E-01	-7.4731036E-01	-7.4731036E-01
	Std.	1.7044364E-14	4.2951518E-09	4.2126954E-15
c02	Best	-2.2757892E+00	-2.2776142E+00	-2.2629924E+00
	Mean	-2.1826480E+00	-2.1892114E+00	-2.1994706E+00
	Std.	1.3343611E-01	1.9214218E-01	2.0108304E-02
c03	Best	1.2534573E-22	1.0609288E-16	0.0000000E+00
	Mean	4.6152872E+00	5.3253314E+00	0.0000000E+00
	Std.	4.5256615E+00	4.4377762E+00	0.0000000E+00
c04	Best	-1.0000000E-05	-9.9999296E-06	-1.0000000E-05
	Mean	-6.2396532E-06	-9.0760634E-06	-1.0000000E-05
	Std.	1.3565220E-05	4.6012250E-06	0.0000000E+00
c05	Best	-4.8361062E+02	-4.8361062E+02	-4.8361062E+02
	Mean	-4.8359738E+02	-4.8358682E+02	-4.8361062E+02
	Std.	2.3975582E-02	5.2893881E-02	3.5059842E-13
c06	Best	-5.7866237E+02	-5.7866237E+02	-5.7866237E+02
	Mean	-5.7863219E+02	-5.7865931E+02	-5.7860234E+02
	Std.	5.9729751E-02	7.3512507E-03	2.7507558E-01
c07	Best	1.2889326E-17	1.5558047E-14	0.0000000E+00
	Mean	6.9695739E-15	8.6994386E-11	0.0000000E+00
	Std.	8.5946129E-15	1.3308933E-10	0.0000000E+00
c08	Best	3.0898267E-20	7.6778018E-19	0.0000000E+00
	Mean	5.6973213E+00	6.0197595E+00	8.1968334E+00
	Std.	4.8352056E+00	4.9760659E+00	4.5961673E+00
c09	Best	5.7677413E-18	3.1402380E-15	0.0000000E+00
	Mean	1.1967838E-11	6.1989162E-01	0.0000000E+00
	Std.	5.9498535E-11	1.5951491E+00	0.0000000E+00
c10	Best	4.7716082E-20	4.6085354E-18	0.0000000E+00
	Mean	2.8633987E-02	1.3192800E+01	0.0000000E+00
	Std.	1.4316979E-01	3.6585559E+01	0.0000000E+00
c11	Best	-1.5227132E-03	-1.5227132E-03	-1.5227132E-03
	Mean	-1.5227132E-03	-1.5227124E-03	-1.5227132E-03
	Std.	1.2424959E-12	1.6725789E-09	1.4407112E-17
c12	Best	-5.6838283E+02	-5.6838283E+02	-3.0548877E+02
	Mean	-6.6966693E+01	-8.4840376E+01	-3.4183177E+01
	Std.	1.3607554E+02	1.3376205E+02	8.7249340E+01
c13	Best	-6.8429365E+01	-6.8429365E+01	-6.8429365E+01
	Mean	-6.8335459E+01	-6.8306542E+01	-6.8429295E+01
	Std.	2.3571787E-01	1.5732544E-01	3.0535111E-04
c14	Best	8.4138513E-14	1.6716963E-11	0.0000000E+00
	Mean	8.2686246E-08	1.5951589E-04	0.0000000E+00
	Std.	3.5864485E-07	5.7217915E-04	0.0000000E+00

c15	Best	2.9163009E-16	2.2031668E-11	0.0000000E+00
	Mean	1.1761112E+00	1.6491965E+00	2.1738482E-27
	Std.	1.7483424E+00	1.9056737E+00	9.9021887E-27
c16	Best	0.0000000E+00	0.0000000E+00	0.0000000E+00
	Mean	0.0000000E+00	8.6597396E-16	0.0000000E+00
	Std.	0.0000000E+00	4.2610007E-15	0.0000000E+00
c17	Best	1.5819620E-19	1.3491504E-17	9.8774013E-29
	Mean	6.0354440E-13	4.3876936E-02	2.5313067E-22
	Std.	2.6064242E-12	2.1938468E-01	1.0078392E-21
c18	Best	4.8440270E-21	2.6472429E-18	0.0000000E+00
	Mean	1.5302277E-18	3.9576223E-15	7.3364064E-31
	Std.	4.3928235E-18	6.2515289E-15	1.8450015E-30

Table D.2: Function values obtained by the proposed MEA-LS-R2S with different CS values of CS= 25, 50, 75 and 100 generations

Problem	criteria	CS=25	CS=50	CS=75	CS=100
c01	Best	-7.4731036E-01	-7.4731036E-01	-7.4731036E-01	-7.4731036E-01
	Mean	-7.4731036E-01	-7.4731036E-01	-7.4731036E-01	-7.4731036E-01
	Std.	2.6814434E-16	5.8095631E-15	4.1943074E-15	2.8214377E-16
c02	Best	-2.2366334E+00	-2.2396573E+00	-2.2285037E+00	-2.2332659E+00
	Mean	-2.2059516E+00	-2.1996897E+00	-2.2035334E+00	-2.1997141E+00
	Std.	1.6278231E-02	1.9563403E-02	1.2078798E-02	1.9346828E-02
c03	Best	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00
	Mean	0.0000000E+00	0.0000000E+00	6.4785044E-27	8.0133122E-26
	Std.	0.0000000E+00	0.0000000E+00	2.2422708E-26	2.2460370E-25
c04	Best	-1.0000000E-05	-1.0000000E-05	-1.0000000E-05	-1.0000000E-05
	Mean	-1.0000000E-05	-1.0000000E-05	-1.0000000E-05	-1.0000000E-05
	Std.	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00
c05	Best	-4.8361062E+02	-4.8361062E+02	-4.8361062E+02	-4.8361062E+02
	Mean	-4.8361062E+02	-4.8361062E+02	-4.8361062E+02	-4.8361062E+02
	Std.	3.4809343E-13	1.5653457E-13	2.7433453E-13	9.0756183E-13
c06	Best	-5.7866237E+02	-5.7866237E+02	-5.7866237E+02	-5.7866237E+02
	Mean	-5.7866237E+02	-5.7866237E+02	-5.7866237E+02	-5.7866158E+02
	Std.	3.6545205E-13	3.6692269E-13	5.5533722E-12	2.8471911E-03
c07	Best	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00
	Mean	0.0000000E+00	1.0529321E-28	2.9970601E-28	1.2638084E-27
	Std.	0.0000000E+00	3.1964204E-28	6.6008625E-28	4.9586700E-27
c08	Best	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00
	Mean	9.1761442E+00	9.7437742E+00	1.0051467E+01	1.0474381E+01
	Std.	4.0880393E+00	3.2426056E+00	3.0260314E+00	2.1845406E+00
c09	Best	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00
	Mean	0.0000000E+00	6.1089389E-29	3.0342746E-28	3.4735123E-28
	Std.	0.0000000E+00	3.0544694E-28	1.2792927E-27	8.7394677E-28
c10	Best	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00
	Mean	5.0487098E-29	4.0945036E-28	7.4619931E-28	1.3202376E-26
	Std.	2.5243549E-28	1.7908399E-27	2.2503728E-27	3.4953462E-26
c11	Best	-1.5227132E-03	-1.5227132E-03	-1.5227132E-03	-1.5227132E-03
	Mean	-1.5227132E-03	-1.5227132E-03	-1.5227132E-03	-1.5227132E-03
	Std.	6.6938692E-18	7.1153736E-18	5.4665369E-18	1.1295233E-17
c12	Best	-1.1602762E+02	-1.1602762E+02	-1.1432711E+02	-1.2667138E+01
	Mean	-1.6388441E+01	-5.7620655E+00	-5.2630758E+00	-1.1966772E+00
	Std.	3.7445613E+01	2.3196732E+01	2.2857855E+01	3.4522031E+00
c13	Best	-6.8429365E+01	-6.8429365E+01	-6.8429365E+01	-6.8429120E+01
	Mean	-6.8414095E+01	-6.8410227E+01	-6.8408192E+01	-6.8414287E+01
	Std.	1.5047902E-02	1.8960127E-02	2.0436101E-02	1.4365403E-02
c14	Best	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00
	Mean	2.2361998E-27	5.6397496E-26	5.4851329E-26	8.3210487E-25
	Std.	2.7739946E-27	2.4420816E-25	1.5510197E-25	2.9497895E-24

c15	Best	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00
	Mean	2.0895348E-27	1.7989779E-01	1.2831548E-26	3.2386337E-26
	Std.	7.1983746E-27	8.9948895E-01	3.1477882E-26	6.0381004E-26
c16	Best	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00
	Mean	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00
	Std.	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00
c17	Best	2.6987375E-25	2.5582616E-25	1.6000546E-26	1.4602517E-25
	Mean	9.4055111E-24	1.2492963E-22	1.6181065E-23	3.6501219E-23
	Std.	1.1756918E-23	4.8168630E-22	3.1710419E-23	1.1656528E-22
c18	Best	2.3111159E-31	2.9582284E-31	7.1490520E-31	2.9582284E-31
	Mean	4.4031997E-30	4.7048157E-30	1.9274337E-29	1.2036711E-28
	Std.	7.2563135E-30	5.7723172E-30	2.7678567E-29	2.1132285E-28

Table D.3: Function values obtained from MEA-LS-R2S with and without linear population size reduction

Problem	V1			V2		
	Best	Mean	Std.	Best	Mean	Std.
c01	-7.473104E-01	-7.473104E-01	2.681443E-16	-7.473104E-01	-7.473104E-01	3.668219E-16
c02	-2.236633E+00	-2.205952E+00	1.627823E-02	-2.205184E+00	-2.168576E+00	2.004337E-02
c03	0.000000E+00	0.000000E+00	0.000000E+00	4.054737E-18	1.412732E-16	1.979497E-16
c04	-1.000000E-05	-1.000000E-05	0.000000E+00	-9.999993E-06	-9.999972E-06	1.502910E-11
c05	-4.836106E+02	-4.836106E+02	3.480934E-13	-4.836106E+02	-4.834679E+02	5.209582E-01
c06	-5.786624E+02	-5.786624E+02	3.654521E-13	-5.786624E+02	-5.786624E+02	2.133904E-07
c07	0.000000E+00	0.000000E+00	0.000000E+00	3.198405E-19	5.900787E-18	1.176939E-17
c08	0.000000E+00	9.176144E+00	4.088039E+00	6.140176E-21	9.788016E+00	3.256664E+00
c09	0.000000E+00	0.000000E+00	0.000000E+00	1.833633E-18	1.957954E-17	1.937901E-17
c10	0.000000E+00	5.048710E-29	2.524355E-28	4.370675E-18	9.900511E-17	1.418788E-16
c11	-1.522713E-03	-1.522713E-03	6.693869E-18	-1.522713E-03	-1.522713E-03	1.555434E-15
c12	-1.160276E+02	-1.638844E+01	3.744561E+01	-1.160276E+02	-1.239006E+01	3.162959E+01
c13	-6.842937E+01	-6.841410E+01	1.504790E-02	-6.823191E+01	-6.772957E+01	4.379487E-01
c14	0.000000E+00	2.236200E-27	2.773995E-27	4.401067E-17	8.581458E-15	3.695477E-14
c15	0.000000E+00	2.089535E-27	7.198375E-27	1.581785E-17	7.978741E-16	1.000109E-15
c16	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
c17	2.698738E-25	9.405511E-24	1.175692E-23	1.463279E-16	2.689475E-15	2.383620E-15
c18	2.311116E-31	4.403200E-30	7.256314E-30	7.894231E-22	8.773713E-21	9.222488E-21

Table D.4: Function values obtained by the proposed MEA-LS-R2S with different NP^{init} values of NP^{init}= 50, 75, 100 and 125 individuals

Problem	criteria	$NP^{init} = 50$	$NP^{init} = 75$	$NP^{init} = 100$	$NP^{init} = 125$
c01	Best	-7.473104E-01	-7.473104E-01	-7.473104E-01	-7.473104E-01
	Mean	-7.447855E-01	-7.470402E-01	-7.470402E-01	-7.470402E-01
	Std.	6.110261E-03	1.350628E-03	1.350628E-03	1.350628E-03
c02	Best	-2.277710E+00	-2.277710E+00	-2.269389E+00	-2.241026E+00
	Mean	-2.276648E+00	-2.263310E+00	-2.234157E+00	-2.220710E+00
	Std.	5.287305E-03	1.587400E-02	1.474365E-02	1.700886E-02
c03	Best	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
	Mean	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
	Std.	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
c04	Best	-1.000000E-05	-1.000000E-05	-1.000000E-05	-1.000000E-05
	Mean	-1.000000E-05	-1.000000E-05	-1.000000E-05	-1.000000E-05
	Std.	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
c05	Best	-4.836106E+02	-4.836106E+02	-4.836106E+02	-4.836106E+02
	Mean	-4.836106E+02	-4.836106E+02	-4.836106E+02	-4.836106E+02
	Std.	3.480934E-13	3.480934E-13	3.480934E-13	3.480934E-13
c06	Best	-5.786624E+02	-5.786624E+02	-5.786624E+02	-5.786624E+02
	Mean	-5.780678E+02	-5.786624E+02	-5.786624E+02	-5.786624E+02
	Std.	7.432283E-01	3.979039E-13	4.085878E-13	3.756256E-13
c07	Best	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
	Mean	1.594632E-01	0.000000E+00	0.000000E+00	0.000000E+00
	Std.	7.973158E-01	0.000000E+00	0.000000E+00	0.000000E+00
c08	Best	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
	Mean	1.411399E+01	8.731391E+00	8.152591E+00	1.048913E+01
	Std.	2.418676E+01	5.690794E+00	4.570200E+00	2.186477E+00
c09	Best	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
	Mean	1.763559E-01	2.653057E-01	0.000000E+00	0.000000E+00
	Std.	8.817793E-01	1.326529E+00	0.000000E+00	0.000000E+00
c10	Best	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
	Mean	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
	Std.	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
c11	Best	-1.522713E-03	-1.522713E-03	-1.522713E-03	-1.522713E-03
	Mean	-1.522713E-03	-1.522713E-03	-1.522713E-03	-1.522713E-03
	Std.	4.517559E-15	5.800059E-18	6.045339E-18	6.394495E-18
c12	Best	-1.992458E-01	-4.231332E+02	-3.054888E+02	-3.037844E+02
	Mean	-1.992458E-01	-2.362370E+01	-8.058275E+01	-5.674957E+01
	Std.	5.405345E-10	9.037024E+01	1.217887E+02	1.019679E+02
c13	Best	-6.842937E+01	-6.842937E+01	-6.842937E+01	-6.842937E+01
	Mean	-6.842908E+01	-6.842904E+01	-6.842273E+01	-6.842140E+01
	Std.	1.107314E-03	1.009323E-03	2.312649E-02	9.359518E-03
c14	Best	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
	Mean	1.594632E-01	0.000000E+00	0.000000E+00	0.000000E+00
	Std.	7.973158E-01	0.000000E+00	0.000000E+00	0.000000E+00

c15	Best	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
	Mean	5.396934E-01	0.000000E+00	0.000000E+00	0.000000E+00
	Std.	1.491634E+00	0.000000E+00	0.000000E+00	0.000000E+00
c16	Best	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
	Mean	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
	Std.	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
c17	Best	0.000000E+00	0.000000E+00	0.000000E+00	8.196758E-30
	Mean	7.395571E-34	1.232595E-33	2.173460E-29	1.208299E-25
	Std.	2.044028E-33	2.516024E-33	6.693253E-29	3.292050E-25
c18	Best	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
	Mean	2.822643E-32	2.428212E-32	1.503766E-32	1.836567E-32
	Std.	2.829381E-32	3.187913E-32	2.637809E-32	2.779723E-32

Table D.5: Function values obtained by the proposed MEA-LS-R2S with different NP^{min} values of NP^{min}= 4, 10, 30 and 40 individuals

Problem	criteria	$NP^{min} = 4$	$NP^{min} = 10$	$NP^{min} = 30$	$NP^{min} = 40$
c01	Best	-7.473104E-01	-7.473104E-01	-7.473104E-01	-7.473104E-01
	Mean	-7.473104E-01	-7.470402E-01	-7.473104E-01	-7.470402E-01
	Std.	2.533726E-16	1.350628E-03	2.254874E-16	1.350628E-03
c02	Best	-2.277710E+00	-2.277710E+00	-2.277710E+00	-2.269389E+00
	Mean	-2.271055E+00	-2.262074E+00	-2.252877E+00	-2.234157E+00
	Std.	1.121433E-02	1.667956E-02	1.889332E-02	1.474365E-02
c03	Best	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
	Mean	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
	Std.	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
c04	Best	-1.000000E-05	-1.000000E-05	-1.000000E-05	-1.000000E-05
	Mean	-1.000000E-05	-1.000000E-05	-1.000000E-05	-1.000000E-05
	Std.	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
c05	Best	-4.836106E+02	-4.836106E+02	-4.836106E+02	-4.836106E+02
	Mean	-4.836106E+02	-4.836106E+02	-4.836106E+02	-4.836106E+02
	Std.	3.480934E-13	3.480934E-13	3.480934E-13	3.480934E-13
c06	Best	-5.786624E+02	-5.786624E+02	-5.786624E+02	-5.786624E+02
	Mean	-5.786624E+02	-5.786624E+02	-5.786624E+02	-5.786624E+02
	Std.	6.345408E-12	5.921000E-13	4.372388E-13	4.085878E-13
c07	Best	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
	Mean	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
	Std.	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
c08	Best	0.000000E+00	3.986579E+00	0.000000E+00	0.000000E+00
	Mean	9.613806E+00	1.066334E+01	9.758722E+00	8.152591E+00
	Std.	3.624085E+00	1.390992E+00	3.279714E+00	4.570200E+00
c09	Best	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
	Mean	0.000000E+00	2.089906E-01	6.877943E-01	0.000000E+00
	Std.	0.000000E+00	1.044953E+00	3.438971E+00	0.000000E+00
c10	Best	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
	Mean	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
	Std.	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
c11	Best	-1.522713E-03	-1.522713E-03	-1.522713E-03	-1.522713E-03
	Mean	-1.522713E-03	-1.522713E-03	-1.522713E-03	-1.522713E-03
	Std.	5.696431E-17	3.507778E-17	7.601796E-18	6.045339E-18
c12	Best	-3.054888E+02	-3.054888E+02	-3.054888E+02	-3.054888E+02
	Mean	-8.508055E+01	-7.938110E+01	-7.538300E+01	-8.058275E+01
	Std.	1.209995E+02	1.222434E+02	1.228049E+02	1.217887E+02
c13	Best	-6.842937E+01	-6.842937E+01	-6.842937E+01	-6.842937E+01
	Mean	-6.842912E+01	-6.842920E+01	-6.842740E+01	-6.842273E+01
	Std.	5.096311E-04	3.786048E-04	5.845673E-03	2.312649E-02
c14	Best	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
	Mean	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
	Std.	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00

c15	Best	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
	Mean	1.798978E-01	1.798978E-01	0.000000E+00	0.000000E+00
	Std.	8.994890E-01	8.994890E-01	0.000000E+00	0.000000E+00
c16	Best	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
	Mean	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
	Std.	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
c17	Best	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
	Mean	3.451266E-33	2.711709E-33	1.380507E-32	2.173460E-29
	Std.	3.122304E-33	3.122304E-33	5.626789E-32	6.693253E-29
c18	Best	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
	Mean	8.443277E-33	1.639352E-32	1.756448E-32	1.503766E-32
	Std.	1.840915E-32	2.725448E-32	2.681957E-32	2.637809E-32

Table D.6: Function values obtained by MEA-LS-R2S, MLS-MODE, MJADE-R2S, EHCT-DE, AIS-ZHY, ISMOADE-CMA, SAMO-DE, ECHT-EP2, ϵDEg, AH-DEa and SAMO-GA for CEC2006 for g01-g04

problem	Alg.	Mean	Std.
g01	MEA-LS-R2S	-1.50000E+01	0.00000E+00
	MJADE-R2S	-1.5000E+01	0.00000E+00
	MLS-MODE	-1.50000E+01	0.00000E+00
	EHCT_DE	-1.50000E+01	0.00000E+00
	AIS-ZHY	-1.50000E+01	0.00000E+00
	ISMOADE-CMA	-1.50000E+01	0.00000E+00
	SAMO-DE	-1.500000E+01	0.000000E+00
	ECHT-EP2	-1.50000E+01	0.00000E+00
	ϵDEg	-1.50000E+01	0.00000E+00
	AH-DEa	-1.500000E+01	0.000000E+00
	SAMO-GA	-1.500000E+01	0.000000E+00
g02	MEA-LS-R2S	-8.036191E-01	4.163894E-07
	MJADE-R2S	8.0358E-01	7.6037E-05
	MLS-MODE	-8.036191E-01	4.509747E-16
	EHCT_DE	-7.936387E-01	1.120000E-02
	AIS-ZHY	-8.021930E-01	5.190000E-10
	ISMOADE-CMA	-7.92440E-01	2.80000E-02
	SAMO-DE	-7.987000E-01	8.801E- 03
	ECHT-EP2	-7.998220E-01	1.260000E-02
	ϵDEg	-8.036191E-01	1.750000E-08
	AH-DEa	-8.007000E-01	3.912E-03
	SAMO-GA	-7.960000E-01	5.803E- 03
g03	MEA-LS-R2S	-1.0005001E+00	0.000000E+00
	MJADE-R2S	-1.000500E+00	0.000000E+00
	MLS-MODE	-1.0005000E+00	0.000000E+00
	EHCT_DE	-1.0005000E+00	0.000000E+00
	AIS-ZHY	-1.0005000E+00	1.7700000E-11
	ISMOADE-CMA	-1.0005000E+00	0.000000E+00
	SAMO-DE	-1.0005000E+00	0.000000E+00
	ECHT-EP2	-1.0005000E+00	0.000000E+00
	ϵDEg	-1.0005000E+00	2.9600000E-31
	AH-DEa	-1.0005000E+00	0.000000E+00
	SAMO-GA	-1.0005000E+00	0.000000E+00
g04	MEA-LS-R2S	-3.0665539E+04	0.00000000E+00
	MJADE-R2S	3.06665539E+04	0.0000000E+00
	MLS-MODE	-3.06655390E+04	0.00000000E+00
	EHCT_DE	-3.06655390E+04	0.00000000E+00
	AIS-ZHY	-3.06655390E+04	3.69000000E-13
	ISMOADE-CMA	-3.06655390E+04	0.00000000E+00
	SAMO-DE	-3.06655386E+04	0.00000000E+00
	ECHT-EP2	-3.06655390E+04	0.00000000E+00
	ϵDEg	-3.06655390E+04	0.00000000E+00
	AH-DEa	-3.06655387E+04	0.000000E+00
	SAMO-GA	-3.06655387E+04	0.000000E+00

Table D.7: Function values obtained by MEA-LS-R2S, MLS-MODE, MJADE-R2S, EHCT-DE, AIS-ZHY, ISMOADE-CMA, SAMO-DE, ECHT-EP2, ϵDEg, AH-DEa and SAMO-GA for CEC2006 for g05-g08

problem	Alg.	Mean	Std.
g05	MEA-LS-R2S	5.126497E+03	0.0000000E+00
	MJADE-R2S	5.126497E+03	0.0000000E+00
	MLS-MODE	5.1264970E+03	0.000000E+00
	EHCT_DE	5.126497E+03	0.000000E+00
	AIS-ZHY	5.126498E+03	1.700000E-02
	ISMOADE-CMA	5.126497E+03	0.000000E+00
	SAMO-DE	5.126497E+03	0.000000E+00
	ECHT-EP2	5.126497E+03	0.000000E+00
	ϵDEg	5.126497E+03	0.000000E+00
	AH-DEa	5.126000E+03	0.000000E+00
	SAMO-GA	5.128000E+03	1.117000E+00
g06	MEA-LS-R2S	-6.961814E+03	0.0000000E+00
	MJADE-R2S	-6.961814E+03	0.0000000E+00
	MLS-MODE	-6.9618140E+03	0.0000000E+00
	EHCT_DE	-6.96181400E+03	0.00000000E+00
	AIS-ZHY	-6.96181385E+03	1.90000000E-12
	ISMOADE-CMA	-6.96181388E+03	0.00000000E+00
	SAMO-DE	-6.96181388E+03	0.00000000E+00
	ECHT-EP2	-6.96181400E+03	0.00000000E+00
	ϵDEg	-6.96181388E+03	0.00000000E+00
	AH-DEa	-6.9618140E+03	0.000000E+00
	SAMO-GA	-6.9618140E+03	0.000000E+00
g07	MEA-LS-R2S	2.430621E+01	0.0000000E+00
	MJADE-R2S	2.430621E+01	0.0000000E+00
	MLS-MODE	2.4306210E+01	0.000000E+00
	EHCT_DE	2.430620E+01	1.140000E-10
	AIS-ZHY	2.435570E+01	8.200000E-03
	ISMOADE-CMA	2.4306210E+01	0.000000E+00
	SAMO-DE	2.430960E+01	1.590000E-03
	ECHT-EP2	2.430630E+01	3.190000E-05
	ϵDEg	2.430620E+01	2.180000E-15
	AH-DEa	2.431000E+01	0.000000E+00
	SAMO-GA	2.441000E+01	4.591000E-02
g08	MEA-LS-R2S	-9.582504E-02	0.0000000E+00
	MJADE-R2S	-9.5825040E-02	0.0000000E+00
	MLS-MODE	-9.5825040E-02	0.000000E+00
	EHCT_DE	-9.582500E-02	0.000000E+00
	AIS-ZHY	-9.582500E-02	0.000000E+00
	ISMOADE-CMA	-9.582500E-02	0.000000E+00
	SAMO-DE	-9.582504E-02	0.000000E+00
	ECHT-EP2	-9.582500E-02	2.610000E-08
	ϵDEg	-9.582500E-02	1.230000E-32
	AH-DEa	-9.582500E-02	0.000000E+00
	SAMO-GA	-9.582500E-02	0.000000E+00

Table D.8: Function values obtained by MEA-LS-R2S, MLS-MODE, MJADE-R2S, EHCT-DE, AIS-ZHY, ISMOADE-CMA, SAMO-DE, ECHT-EP2, ϵDEg, AH-DEa and SAMO-GA for CEC2006 for g09-g12

problem	Alg.	Mean	Std.
g09	MEA-LS-R2S	6.806301E+02	0.000000E+00
	MJADE-R2S	6.806301E+02	0.000000E+00
	MLS-MODE	6.8063010E+02	0.000000E+00
	EHCT_DE	6.8063010E+02	0.000000E+00
	AIS-ZHY	6.806500E+02	1.200000E-08
	ISMOADE-CMA	6.8063010E+02	0.000000E+00
	SAMO-DE	6.8063010E+02	1.160000E-05
	ECHT-EP2	6.8063010E+02	0.000000E+00
	ϵDEg	6.8063010E+02	0.000000E+00
	AH-DEa	6.8063010E+02	0.000000E+00
	SAMO-GA	6.8063010E+02	1.457000E-03
g10	MEA-LS-R2S	7.0492480E+03	0.000000E+00
	MJADE-R2S	7.0492480E+03	1.0000E-10
	MLS-MODE	7.0492480E+03	6.0989160E-10
	EHCT_DE	7.049248E+03	4.180000E-07
	AIS-ZHY	7.049570E+03	4.500000E-04
	ISMOADE-CMA	7.049248E+03	5.420000E-06
	SAMO-DE	7.059813E+03	7.860000E+00
	ECHT-EP2	7.049249E+03	6.600000E-04
	ϵDEg	7.049248E+03	4.240000E-13
	AH-DEa	7.0492480E+03	1.688000E-09
	SAMO-GA	7.144000E+03	6.786000E+01
g11	MEA-LS-R2S	7.499000E-01	0.000000E+00
	MJADE-R2S	7.4990000E-01	0.000000E+00
	MLS-MODE	7.4990000E-01	0.000000E+00
	EHCT_DE	7.499000E-01	0.000000E+00
	AIS-ZHY	7.499000E-01	1.400000E-08
	ISMOADE-CMA	7.499000E-01	0.000000E+00
	SAMO-DE	7.499000E-01	0.000000E+00
	ECHT-EP2	7.499000E-01	0.000000E+00
	ϵDEg	7.499000E-01	0.000000E+00
	AH-DEa	7.499000E-01	0.000000E+00
	SAMO-GA	7.499000E-01	0.000000E+00
g12	MEA-LS-R2S	-1.000000E+00	0.000000E+00
	MJADE-R2S	-1.0000000E+00	0.0000000E+00
	MLS-MODE	-1.0000000E+00	0.0000000E+00
	EHCT_DE	-1.000000E+00	0.000000E+00
	AIS-ZHY	-1.000000E+00	0.000000E+00
	ISMOADE-CMA	-1.000000E+00	0.000000E+00
	SAMO-DE	-1.000000E+00	0.000000E+00
	ECHT-EP2	-1.000000E+00	0.000000E+00
	ϵDEg	-1.000000E+00	0.000000E+00
	AH-DEa	-1.000000E+00	0.000000E+00
	SAMO-GA	-1.000000E+00	0.000000E+00

Table D.9: Function values obtained by MEA-LS-R2S, MLS-MODE, MJADE-R2S, EHCT-DE, AIS-ZHY, ISMOADE-CMA, SAMO-DE, ECHT-EP2, ϵDEg, AH-DEa and SAMO-GA for CEC2006 for g13-g16

problem	Alg.	Mean	Std.
g13	MEA-LS-R2S	5.394150E-02	0.000000E+00
	MJADE-R2S	5.394150E-02	0.000000E+00
	MLS-MODE	5.39415E-02	0.000000E+00
	EHCT_DE	5.39415E-02	0.000000E+00
	AIS-ZHY	5.39415E-02	7.800000E-10
	ISMOADE-CMA	5.39415E-02	0.000000E+00
	SAMO-DE	5.39415E-02	1.750000E-08
	ECHT-EP2	5.39415E-02	1.000000E-12
	ϵDEg	5.39415E-02	0.000000E+00
	AH-DEa	5.39415E-02	0.000000E+00
	SAMO-GA	5.403000E-02	5.941000E-05
g14	MEA-LS-R2S	-4.7764890E+01	0.000000E+00
	MJADE-R2S	-4.7764890E+01	0.000000E+00
	MLS-MODE	-4.7764890E+01	2.6066690E-14
	EHCT_DE	-4.776489E+01	3.260000E-13
	AIS-ZHY	-4.776488E+01	1.000000E-12
	ISMOADE-CMA	-4.776489E+01	0.000000E+00
	SAMO-DE	-4.768115E+01	4.040000E-02
	ECHT-EP2	-4.776480E+01	2.720000E-05
	ϵDEg	-4.776489E+01	1.390000E-15
	AH-DEa	-4.776000E+01	3.894000E-05
	SAMO-GA	-4.776489E+01	3.159000E-01
g15	MEA-LS-R2S	9.617150E+02	0.000000E+00
	MJADE-R2S	9.617150E+02	0.000000E+00
	MLS-MODE	9.6171500E+02	0.000000E+00
	EHCT_DE	9.617150E+02	0.000000E+00
	AIS-ZHY	9.617150E+02	0.000000E+00
	ISMOADE-CMA	9.617150E+02	0.000000E+00
	SAMO-DE	9.617150E+02	0.000000E+00
	ECHT-EP2	9.617150E+02	2.010000E-13
	ϵDEg	9.617150E+02	0.000000E+00
	AH-DEa	9.617150E+02	0.000000E+00
	SAMO-GA	9.617150E+02	5.524000E-05
g16	MEA-LS-R2S	-1.905155E+00	0.000000E+00
	MJADE-R2S	-1.905155E+00	0.000000E+00
	MLS-MODE	-1.9051550E+00	0.000000E+00
	EHCT_DE	-1.905155E+00	0.000000E+00
	AIS-ZHY	-1.905155E+00	0.000000E+00
	ISMOADE-CMA	-1.905155E+00	0.000000E+00
	SAMO-DE	-1.905155E+00	0.000000E+00
	ECHT-EP2	-1.905155E+00	1.120000E-10
	ϵDEg	-1.905155E+00	1.580000E-30
	AH-DEa	-1.905155E+00	0.000000E+00
	SAMO-GA	-1.905155E+00	6.952000E-07

Table D.10: Function values obtained by MEA-LS-R2S, MLS-MODE, MJADE-R2S, EHCT-DE, AIS-ZHY, ISMOADE-CMA, SAMO-DE, ECHT-EP2, ϵDEg, AH-DEa and SAMO-GA for CEC2006 for g17-g21

problem	Alg.	Mean	Std.
g17	MEA-LS-R2S	8.853540E+03	0.000000E+00
	MJADE-R2S	8.8565E+03	1.4812E+01
	MLS-MODE	8.853540E+03	0.000000E+00
	EHCT_DE	8.853540E+03	0.000000E+00
	AIS-ZHY	8.853540E+03	1.900000E-09
	ISMOADE-CMA	8.853540E+03	0.000000E+00
	SAMO-DE	8.853540E+03	1.150000E-05
	ECHT-EP2	8.853540E+03	2.130000E-08
	ϵDEg	8.853540E+03	1.210000E-27
	AH-DEa	8.858000E+03	1.847000E+01
	SAMO-GA	8.854000E+03	1.740000E-01
g18	MEA-LS-R2S	-8.660254E-01	0.000000E+00
	MJADE-R2S	-8.3546E-01	7.1482E-02
	MLS-MODE	-8.6602540E-01	5.1068250E-15
	EHCT_DE	-8.660240E-01	5.150000E-06
	AIS-ZHY	-8.660250E-01	1.300000E-15
	ISMOADE-CMA	-8.660250E-01	0.000000E+00
	SAMO-DE	-8.660240E-01	7.040000E-07
	ECHT-EP2	-8.660250E-01	1.000000E-09
	ϵDEg	-8.660250E-01	2.180000E-17
	AH-DEa	-8.6602540E-01	0.000000E+00
	SAMO-GA	-8.655000E-01	4.080000E-04
g19	MEA-LS-R2S	3.265559E+01	0.000000E+00
	MJADE-R2S	3.265559E+01	0.000000E+00
	MLS-MODE	3.2655590E+01	1.7525120E-14
	EHCT_DE	3.265654E+01	7.760000E-04
	AIS-ZHY	3.265559E+01	0.000000E+00
	ISMOADE-CMA	3.265559E+01	6.460000E-07
	SAMO-DE	3.275734E+01	6.150000E-02
	ECHT-EP2	3.266230E+01	3.400000E-03
	ϵDEg	3.265560E+01	1.260000E-05
	AH-DEa	3.457000E+01	2.524000E+00
	SAMO-GA	3.643000E+01	1.037000E+00
g21	MEA-LS-R2S	1.937245E+02	0.000000E+00
	MJADE-R2S	2.3040E+02	6.0021E+01
	MLS-MODE	1.937245E+02	0.000000E+00
	EHCT_DE	1.937245E+02	0.000000E+00
	AIS-ZHY	1.967245E+02	1.100000E+00
	ISMOADE-CMA	1.937245E+02	0.000000E+00
	SAMO-DE	1.937714E+02	1.960000E-02
	ECHT-EP2	1.937438E+02	1.650000E-02
	ϵDEg	1.937245E+02	3.340000E-14
	AH-DEa	1.939000E+02	6.977000E-01
	SAMO-GA	2.461000E+02	1.492000E+01

Table D.11: Function values obtained by MEA-LS-R2S, MLS-MODE, MJADE-R2S, EHCT-DE, AIS-ZHY, ISMOADE-CMA, SAMO-DE, ECHT-EP2, ϵDEg, AH-DEa and SAMO-GA for CEC2006 for g23-g24

problem	Alg.	Mean	Std.
g23	MEA-LS-R2S	-4.000551E+02	0.000000E+00
	MJADE-R2S	-4.000520E+02	3.180000E-04
	MLS-MODE	-4.0005120E+02	1.9083780E-02
	EHCT_DE	-4.000546E+02	2.180000E-03
	AIS-ZHY	-3.998743E+02	2.000000E+00
	ISMOADE-CMA	-3.956240E+02	7.790000E+00
	SAMO-DE	-3.608177E+02	1.960000E+01
	ECHT-EP2	-3.732178E+02	3.370000E+01
	ϵDEg	-4.000551E+02	1.110000E-14
	AH-DEa	-3.444000E+02	7.782000E+01
	SAMO-GA	-1.948000E+02	5.328000E+01
g24	MEA-LS-R2S	-5.508013E+00	0.000000E+00
	MJADE-R2S	-5.5080130E+00	0.000000E+00
	MLS-MODE	-5.5080130E+00	0.000000E+00
	EHCT_DE	-5.508013E+00	0.000000E+00
	AIS-ZHY	-5.508013E+00	0.000000E+00
	ISMOADE-CMA	-5.508013E+00	0.000000E+00
	SAMO-DE	-5.508013E+00	0.000000E+00
	ECHT-EP2	-5.508013E+00	0.000000E+00
	ϵDEg	-5.508013E+00	2.520000E-29
	AH-DEa	-5.508000E+00	0.000000E+00
	SAMO-GA	-5.508000E+00	0.000000E+00

Table D.12: Function values obtained by MEA-LS-R2S, MJADE-R2S, MLS-MODE, DEbavDBmax , SAMODE, DE-DBmax, SAMO-DE, eABC, Co-CLPSO, SAMO-GA,ϵDEg, and ECHT-ARMOR-DE for CEC2010 for c01-c03

		10D			30D		
problem	Alg.	Best	Mean	Std.	Best	Mean	Std.
	MEA-LS-R2S	-7.4731036E-01	-7.4731036E-01	2.5337258E-16	-8.2188433E-01	-8.2149710E-01	9.6427287E-04
	MJADE-R2S	-7.4731000E-01	-7.4529000E-01	5.2900000E-03	-8.2188000E-01	-8.1884000E-01	3.3000000E-03
	MLS-MODE	-7.4731036E-01	-7.4731036E-01	4.2142800E-15	-8.2188435E-01	-8.2180624E-01	3.8996670E-04
	DEbavDBmax	-7.473104E-01	-7.463993E-01	2.552887E-03	-8.218843E-01	-8.135627E-01	7.854595E-03
	SAMODE	-7.473104E-01	-7.470402E-01	1.350638E-03	-8.218838E-01	-8.143674E-01	4.766000E-03
c01	ϵDEag	-7.473104E-01	-7.467701E-01	1.869859E-03	-8.218255E-01	-8.208687E-01	7.103893E-04
	DE-DBmax	-7.473104E-01	-7.463993E-01	2.552887E-03	-8.218843E-01	-8.146829E-01	7.216108E-03
	eABC	-7.472710E-01	-7.162570E-01	2.689780E-02	-8.162650E-01	-7.305540E-01	4.875890E-02
	Co-CLPSO	-7.473100E-01	-7.335800E-01	1.784800E-02	-8.068800E-01	-7.159800E-01	5.025200E-02
	SAMO-GA	-7.473093E-01	-7.470230E-01	1.347658E-03	-8.217813E-01	-8.115299E-01	7.247940E-03
	ECHT-ARMOR-DE	-7.473000E-01	-7.470000E+00	1.400000E-03	-8.180600E-01	-7.899200E-01	2.510000E-02
	MEA-LS-R2S	-2.2777100E+00	-2.2710550E+00	1.1214327E-02	-2.2210704E+00	-1.9697560E+00	1.1252965E-01
	MJADE-R2S	-2.2777000E+00	-2.0103000E+00	4.0944000E-01	-2.2807000E+00	-2.2702000E+00	8.4000000E-03
	MLS-MODE	-2.2355979E+00	-2.2123734E+00	1.1919289E-02	-2.1390923E+00	-2.0099045E+00	8.8624158E-02
	DEbavDBmax	-2.277710E+00	-2.198366E+00	8.389727E-02	-2.280973E+00	-2.276646E+00	3.150424E-03
	SAMODE	-2.277709E+00	-2.276842E+00	1.154957E-03	-2.280962E+00	-2.276111E+00	3.706000E-03
c02	ϵDEag	-2.277702E+00	-2.269502E+00	2.389779E-02	-2.169248E+00	-2.151424E+00	1.197582E-02
	DE-DBmax	-2.277711E+00	-2.182633E+00	1.190727E-01	-2.280973E+00	-2.275178E+00	4.677619E-03
	eABC	-2.155150E+00	-1.248950E-01	1.583590E+00	-1.236470E-01	2.556470E+00	9.429490E-01
	Co-CLPSO	-2.277700E+00	-2.266600E+00	1.461600E-02	-2.280900E+00	-2.202900E+00	1.926700E-01
	SAMO-GA	-2.277497E+00	-2.272609E+00	2.267800E-03	-2.265137E+00	-2.252288E+00	5.747300E-03
	ECHT-ARMOR-DE	-2.277700E+00	-2.277000E+03	3.300000E-03	-2.260700E+00	-2.170600E+00	7.360000E-02
	MEA-LS-R2S	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00
	MJADE-R2S	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00	1.0000000E-04	4.9000000E-04
	MLS-MODE	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00	1.9003142E-25	4.4111754E-25
	DEbavDBmax	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	6.382170E-25	7.133478E-25
	SAMODE	0.000000E+00	4.173000E-23	1.640510E-22	4.599600E-24	4.826000E-22	1.146000E-21
c03	ϵDEag	0.000000E+00	0.000000E+00	0.000000E+00	2.867347E+01	2.883785E+01	8.047159E-01
	DE-DBmax	0.000000E+00	2.686721E-26	1.023925E-25	0.000000E+00	1.508909E-16	6.314477E-16
	eABC	8.788210E+11	-	1.009670E+12	-	-	-
	Co-CLPSO	2.474800E-13	3.550200E-01	1.775100E+00	-	-	-
	SAMO-GA	6.492724E-22	1.190663E-10	2.074947E-10	5.481000E-19	2.255000E-07	8.154000E-07
	ECHT-ARMOR-DE	0.000000E+00	0.000000E+00	0.000000E+00	2.580100E-24	2.638000E+01	7.940000E+00

Table D.13: Function values obtained by MEA-LS-R2S, MJADE-R2S, MLS-MODE, DEbavDBmax , SAMODE, DE-DBmax, SAMO-DE, eABC, Co-CLPSO, SAMO-GA,ϵDEg, and ECHT-ARMOR-DE for CEC2010 for c04-c06

		10D			30D		
problem	Alg.	Best	Mean	Std.	Best	Mean	Std.
c04	MEA-LS-R2S	-1.0000000E-05	-1.0000000E-05	0.0000000E+00	-3.3333333E-06	-3.3333333E-06	8.3741535E-16
	MJADE-R2S	-1.0000000E-05	9.9009000E+00	1.6262000E+01	-1.6641000E-03	8.9271000E+00	1.0413000E+01
	MLS-MODE	-1.0000000E-05	-1.0000000E-05	0.0000000E+00	-3.3333333E-06	-3.3333333E-06	1.1749496E-15
	DEbavDBmax	-1.000000E-05	-1.000000E-05	0.000000E+00	-3.332962E-06	-3.330868E-06	2.772148E-09
	SAMODE	-1.000000E-05	-1.000000E-05	1.446083E-11	-3.248000E-06	-2.411300E-06	4.492340E-07
	ϵDEag	-9.992345E-06	-9.918452E-06	1.546730E-07	4.698111E-03	8.162973E-03	3.067785E-03
	DE-DBmax	-1.000000E-05	-1.000000E-05	0.000000E+00	-3.333311E-06	-3.331620E-06	1.710576E-09
	eABC	-	-	-	-	-	-
	Co-CLPSO	-1.000000E-05	-9.338500E-06	1.074800E-06	-2.930000E-06	1.126900E-01	5.633500E-01
	SAMO-GA	-1.000000E-05	-9.934308E-06	5.118391E-08	-2.644300E-06	1.470915E-03	7.359431E-03
	ECHT-ARMOR-DE	-1.000000E-05	-1.000000E-05	0.000000E+00	-3.332600E-06	8.371300E-02	2.890000E-01
c05	MEA-LS-R2S	-4.8361062E+02	-4.8361062E+02	3.4809343E-13	-4.8361062E+02	-4.8360044E+02	4.8867962E-02
	MJADE-R2S	-2.6391E+02	3.6320E+01	1.4938E+02	-2.6391000E+02	3.6320000E+01	1.4938000E+02
	MLS-MODE	-4.8361062E+02	-4.8361062E+02	3.4809343E-13	-4.8361062E+02	-4.8361062E+02	1.2749496E-14
	DEbavDBmax	-4.836106E+02	-4.836106E+02	4.948079E-06	-4.836106E+02	-4.836106E+02	7.160225E-09
	SAMODE	-4.836106E+02	-4.836106E+02	4.144280E-06	-4.836106E+02	-4.836106E+02	5.389910E-06
	ϵDEag	-4.836106E+02	-4.836106E+02	3.890350E-13	-4.531307E+02	-4.495460E+02	2.899105E+00
	DE-DBmax	-4.836106E+02	-4.835891E+02	3.316223E-02	-4.836106E+02	-4.836106E+02	3.598556E-08
	eABC	4.659550E+01	3.652470E+02	1.172050E+02	1.463110E+02	3.718920E+02	7.888540E+01
	Co-CLPSO	-4.836100E+02	-4.836000E+02	1.957700E-02	-4.836000E+02	-3.124900E+02	8.833200E+01
	SAMO-GA	-4.836106E+02	-4.016976E+02	1.112600E+02	-4.784754E+02	-4.716434E+02	3.783287E+00
	ECHT-ARMOR-DE	-4.836100E+02	-4.836100E+02	0.000000E+00	-4.812200E+02	-4.333500E+02	1.460000E+02
c06	MEA-LS-R2S	-5.7866237E+02	-5.7866237E+02	6.3454079E-12	-5.3061970E+02	-5.3011598E+02	8.0977815E-01
	MJADE-R2S	-5.7866237E+02	-7.0538000E+01	1.9221000E+02	-1.9382000E+02	1.1422000E+02	1.9705000E+02
	MLS-MODE	-5.7866237E+02	-5.7866237E+02	4.2790171E-13	-5.3063786E+02	-5.3026981E+02	5.9692277E-01
	DEbavDBmax	-5.755047E+02	-5.755047E+02	7.677759E-07	-5.306379E+02	-5.305663E+02	2.346488E-01
	SAMODE	-5.786624E+02	-5.786562E+02	9.352190E-03	-5.306368E+02	-5.306155E+02	1.288050E-02
	ϵDEag	-5.786581E+02	-5.786528E+02	3.627169E-03	-5.285750E+02	-5.279068E+02	4.748378E-01
	DE-DBmax	-5.755047E+02	-5.755047E+02	3.512664E-05	-5.306379E+02	-5.306316E+02	1.817352E-02
	eABC	2.384680E+02	4.381680E+02	8.595360E+01	3.259620E+02	4.738410E+02	6.302590E+01
	Co-CLPSO	-5.786600E+02	-5.786600E+02	5.728900E-04	-2.860100E+02	-2.447000E+02	3.948100E+01
	SAMO-GA	-5.786619E+02	-5.777351E+02	3.260495E+00	-5.249595E+02	-5.208121E+02	3.169611E+00
	ECHT-ARMOR-DE	-5.786600E+02	-5.786600E+02	4.000000E-13	-5.246500E+02	-4.893100E+02	1.320000E+02

Table D.14: Function values obtained by MEA-LS-R2S, MJADE-R2S, MLS-MODE, DEbavDBmax, SAMODE, DE-DBmax, SAMO-DE, eABC, Co-CLPSO, SAMO-GA,ϵDEg, and ECHT-ARMOR-DE for CEC2010 for c07-c09

		10D			30D		
problem	Alg.	Best	Mean	Std.	Best	Mean	Std.
c07	MEA-LS-R2S	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00	7.5841876E-28	2.8794751E-27
	MJADE-R2S	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00	4.7839000E-01	1.3222000E+00
	MLS-MODE	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00	3.0805556E-27	5.4724404E-27
	DEbavDBmax	0.000000E+00	1.250285E-28	2.274570E-28	0.000000E+00	1.594650E-01	7.973248E-01
	SAMODE	0.000000E+00	7.762750E-23	3.880800E-22	9.495250E-23	1.782790E-13	3.628040E-13
	ϵDEag	0.000000E+00	0.000000E+00	0.000000E+00	1.147112E-15	2.603632E-15	1.233430E-15
	DE-DBmax	0.000000E+00	1.250285E-28	2.274570E-28	0.000000E+00	3.189299E-01	1.103846E+00
	eABC	1.044680E-03	7.160940E+01	5.191130E+01	1.761440E-01	1.332900E+02	2.058280E+02
	Co-CLPSO	1.071100E-09	7.973200E-01	1.627500E+00	3.786100E-11	1.116300E+00	1.826900E+00
	SAMO-GA	0.000000E+00	7.856600E-23	1.208100E-22	3.236800E-28	3.236800E-28	4.576600E-44
	ECHT-ARMOR-DE	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	1.078900E-25	2.200000E-25
c08	MEA-LS-R2S	0.0000000E+00	9.6138057E+00	3.6240853E+00	0.0000000E+00	2.4028595E-27	4.9698960E-27
	MJADE-R2S	0.0000000E+00	9.6105000E+00	3.0696000E+00	0.0000000E+00	2.2044000E+02	4.2871000E+02
	MLS-MODE	0.0000000E+00	8.7384827E+00	4.4599415E+00	0.0000000E+00	4.3791390E-27	8.8083879E-27
	DEbavDBmax	0.000000E+00	9.216872E+00	3.595848E+00	1.012710E-26	4.200829E-26	4.917721E-26
	SAMODE	0.000000E+00	2.520090E-25	1.260047E-24	6.425810E-21	1.032930E-09	2.373300E-09
	ϵDEag	0.000000E+00	6.727528E+00	5.560648E+00	2.518693E-14	7.831464E-14	4.855177E-14
	DE-DBmax	1.938261E-23	1.045963E+01	2.182499E+00	4.643926E-28	8.265845E-26	1.185239E-25
	eABC	2.870260E-01	4.107890E+02	9.356030E+02	1.600450E-02	1.501960E+02	7.148770E+01
	Co-CLPSO	9.644200E-10	6.087600E-01	1.425500E+00	4.311400E-14	4.751700E+01	1.125900E+02
	SAMO-GA	3.654300E-25	3.648700E-23	5.693200E-23	3.236790E-28	1.594860E-24	6.776570E-24
	ECHT-ARMOR-DE	0.000000E+00	7.526200E+00	5.000000E+00	0.000000E+00	2.010100E+01	4.700000E+01
c09	MEA-LS-R2S	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00	4.2135334E-27	1.4681996E-26
	MJADE-R2S	0.0000000E+00	3.7515000E+11	9.4435000E+11	0.0000000E+00	7.9184000E+11	1.9356000E+12
	MLS-MODE	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00	2.5357772E-25	1.1769998E-24
	DEbavDBmax	0.000000E+00	4.538184E-26	1.839539E-25	0.000000E+00	4.302495E-26	4.474786E-26
	SAMODE	0.000000E+00	5.089752E+00	2.410828E+01	1.186170E-20	6.079667E+00	1.432599E+01
	ϵDEag	0.000000E+00	0.000000E+00	0.000000E+00	2.770665E-16	1.072140E+01	2.821923E+01
	DE-DBmax	0.000000E+00	4.248994E-26	1.564360E-25	0.000000E+00	4.771583E-26	4.869925E-26
	eABC	2.289850E+10	2.019340E+12	1.810830E+12	2.613740E+12	1.607560E+13	9.287410E+12
	Co-CLPSO	3.755100E-16	1.993800E+10	9.968800E+10	1.969500E+02	1.482200E+08	2.450900E+08
	SAMO-GA	3.654300E-25	3.648700E-23	5.693200E-23	3.236790E-28	1.594860E-24	6.776570E-24
	ECHT-ARMOR-DE	0.000000E+00	1.763300E-01	8.800000E-01	0.000000E+00	4.611000E+00	2.310000E+01

Table D.15: Function values obtained by MEA-LS-R2S, MJADE-R2S, MLS-MODE, DEbavDBmax , SAMODE, DE-DBmax, SAMO-DE, eABC, Co-CLPSO, SAMO-GA,ϵDEg, and ECHT-ARMOR-DE for CEC2010 for c10-c12

		10D			30D		
problem	Alg.	Best	Mean	Std.	Best	Mean	Std.
	MEA-LS-R2S	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00	6.5060455E-27	1.6305585E-26
	MJADE-R2S	0.0000000E+00	4.4535000E+10	1.3280000E+11	3.1309000E+01	2.1642000E+12	5.3144000E+12
	MLS-MODE	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00	2.9614474E-27	9.6411285E-27
	DEbavDBmax	0.000000E+00	1.333516E-26	3.338908E-26	1.506488E-26	4.412282E-20	1.211799E-19
	SAMODE	0.000000E+00	4.467656E-01	1.546298E+00	9.769000E-21	1.960780E+01	2.123749E+01
c10	ϵDEag	0.000000E+00	0.000000E+00	0.000000E+00	3.252002E+01	3.326175E+01	4.545577E-01
	DE-DBmax	0.000000E+00	2.165980E-24	1.081388E-23	1.370196E-25	1.540962E-19	2.992547E-19
	eABC	4.493580E+09	1.746200E+12	2.583000E+12	9.358200E+11	1.498640E+13	9.773360E+12
	Co-CLPSO	2.396700E-15	4.974300E+10	2.487100E+11	3.196700E+01	1.395100E+09	5.843800E+09
	SAMO-GA	1.124000E-19	1.732124E+02	2.636938E+02	4.164404E+01	6.031654E+03	7.177385E+03
	ECHT-ARMOR-DE	0.000000E+00	0.000000E+00	0.000000E+00	6.020900E-13	6.553600E+01	1.070000E+02
	MEA-LS-R2S	-1.5227132E-03	-1.5227132E-03	5.6964310E-17	-3.9234394E-04	-3.9234381E-04	9.0895425E-11
	MJADE-R2S	-1.5227132E-03	1.4000000E-01	4.8259000E-01	-3.0787000E-04	4.1150000E-02	2.0760000E-02
	MLS-MODE	-1.5227132E-03	-1.5227132E-03	6.3511486E-18	-3.9234392E-04	-3.9234367E-04	2.0125659E-10
	DEbavDBmax	-1.522713E-03	-1.522713E-03	1.461597E-14	-3.923439E-04	-3.923436E-04	3.984682E-10
	SAMODE	-1.522710E-03	-1.522710E-03	3.667610E-09	-3.923000E-04	-3.869000E-04	6.149660E-06
c11	ϵDEag	-1.522710E-03	-1.522710E-03	6.341035E-11	-3.268462E-04	-2.863882E-04	2.707605E-05
	DE-DBmax	-1.522713E-03	-1.522713E-03	6.880150E-16	-3.923439E-04	1.895262E-03	6.322610E-03
	eABC	-	-	-	-	-	-
	Co-CLPSO	-	-	-	-	-	-
	SAMO-GA	-6.205000E-04	-5.262000E-04	4.904300E-05	-1.360000E-04	-1.260000E-04	4.887000E-06
	ECHT-ARMOR-DE	-1.522700E-03	-	4.400000E-02	-3.923400E-04	-	5.280000E-03
	MEA-LS-R2S	-3.0548877E+02	-8.5080548E+01	1.2099948E+02	-1.9926346E-01	-1.9926346E-01	1.8958744E-09
	MJADE-R2S	-5.700899E+02	-1.232500E-01	3.8228000E+02	-1.991453E-01	-1.7505000E-01	1.1057000E+03
	MLS-MODE	-1.1602762E+02	-1.6819684E+01	3.7509007E+01	-1.9926346E-01	-1.9926346E-01	5.0128642E-09
	DEbavDBmax	-2.202514E-01	-2.202502E-01	6.620404E-07	-1.992635E-01	-1.992635E-01	1.630986E-09
	SAMODE	-5.700899E+02	-1.166134E+02	1.830005E+02	-1.992598E-01	-1.992573E-01	1.321890E-06
c12	ϵDEag	-5.700899E+02	-3.367349E+02	1.782166E+02	-1.991453E-01	-	2.889253E+02
	DE-DBmax	-2.202509E-01	-2.202483E-01	2.733060E-06	-1.992635E-01	-1.992635E-01	1.401959E-09
	eABC	-5.700360E+02	-1.800890E+02	2.757580E+02	-8.826380E+02	-	-
	Co-CLPSO	-1.263900E+01	-2.336900E+00	2.432900E+01	-1.992600E-01	-1.991100E-01	1.184000E-04
	SAMO-GA	-5.697441E+02	-5.588351E+01	1.351800E+02	-8.535971E+01	-3.493162E+00	1.705561E+01
	ECHT-ARMOR-DE	-1.992500E-01	-1.992500E-01	1.600000E-13	-1.992600E-01	-1.607600E-01	1.930000E-01

Table D.16: Function values obtained by MEA-LS-R2S, MJADE-R2S, MLS-MODE, DEbavDBmax , SAMODE, DE-DBmax, SAMO-DE, eABC, Co-CLPSO, SAMO-GA,ϵDEg, and ECHT-ARMOR-DE for CEC2010 for c13-c15

		10D			30D		
problem	Alg.	Best	Mean	Std.	Best	Mean	Std.
	MEA-LS-R2S	-6.8429365E+01	-6.8429118E+01	5.0963112E-04	-6.2363464E+01	-6.0130423E+01	1.2087678E+00
	MJADE-R2S	-6.8429000E+01	-6.8315000E+01	5.7018000E-01	-6.8429000E+01	-6.5867000E+01	1.6725000E+00
	MLS-MODE	-6.8429363E+01	-6.8416666E+01	8.3825697E-03	-6.1630347E+01	-5.8816532E+01	1.4972294E+00
	DEbavDBmax	-6.842937E+01	-6.750889E+01	1.414780E+00	-6.583213E+01	-6.018059E+01	5.066890E+00
	SAMODE	-6.842937E+01	-6.842937E+01	1.542877E-07	-6.842940E+01	-6.819178E+01	3.891641E-01
c13	ϵDEag	-6.842937E+01	-6.842936E+01	1.025960E-06	-6.642473E+01	-6.535310E+01	5.733005E-01
	DE-DBmax	-6.842937E+01	-6.755373E+01	1.275098E+00	-6.575545E+01	-6.002365E+01	5.274321E+00
	eABC	-6.842890E+01	-6.568060E+01	2.502700E+00	-6.756870E+01	-6.485590E+01	1.382130E+00
	Co-CLPSO	-6.842936E+01	-6.397445E+01	2.134080E+00	-6.275200E+01	-6.077400E+01	1.117600E+00
	SAMO-GA	-6.842803E+01	-6.837658E+01	1.558530E-01	-6.637738E+01	-6.309871E+01	1.286011E+00
	ECHT-ARMOR-DE	-6.842900E+01	-6.716900E+01	2.100000E+00	-6.741600E+01	-6.464600E+01	1.970000E+00
	MEA-LS-R2S	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00	3.0652610E-27	1.0423172E-26
	MJADE-R2S	0.0000000E+00	1.7335000E+08	8.4496000E+08	0.0000000E+00	9.8075000E+05	3.4831000E+06
	MLS-MODE	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00	3.0384713E-27	7.2647679E-27
	DEbavDBmax	0.000000E+00	3.303876E-27	4.718921E-27	0.000000E+00	1.010587E-25	8.052860E-26
	SAMODE	0.000000E+00	1.206380E-21	2.435928E-21	1.747000E-22	1.196910E-08	2.569420E-08
c14	ϵDEag	0.000000E+00	0.000000E+00	0.000000E+00	5.015863E-14	3.089407E-13	5.608409E-13
	DE-DBmax	0.000000E+00	3.393617E-27	3.469890E-27	1.391551E-27	3.189299E-01	1.103846E+00
	eABC	3.141520E-04	8.004100E+10	2.366080E+11	3.134910E-01	9.947190E+03	1.916060E+04
	Co-CLPSO	5.780000E-12	3.189300E-01	1.103800E+00	3.288340E-09	6.152420E-02	3.073560E-01
	SAMO-GA	1.876000E-22	3.872582E+02	1.244818E+03	4.145243E+00	1.762866E+03	5.465005E+03
	ECHT-ARMOR-DE	0.000000E+00	0.000000E+00	0.000000E+00	1.580900E-27	6.613500E+02	2.470000E+03
	MEA-LS-R2S	0.0000000E+00	1.7989780E-01	8.9948899E-01	0.0000000E+00	1.4979838E-27	4.6049062E-27
	MJADE-R2S	4.8105000E+11	4.4552000E+13	5.1189000E+13	2.1603000E+01	9.8781000E+13	5.7336000E+13
	MLS-MODE	0.0000000E+00	6.1089389E-29	3.0544694E-28	0.0000000E+00	6.0483543E-28	1.1788763E-27
	DEbavDBmax	0.000000E+00	2.958427E-25	1.086319E-24	5.026464E-26	1.868275E-22	2.968697E-22
	SAMODE	0.000000E+00	7.053800E-04	2.441385E-03	5.842400E-18	2.112810E+00	4.510670E+00
c15	ϵDEag	0.000000E+00	1.798980E-01	8.813156E-01	2.160345E+01	2.160376E+01	1.104834E-04
	DE-DBmax	2.600086E-27	1.714139E-22	8.450532E-22	1.323884E-24	2.113119E-20	6.666822E-20
	eABC	4.216110E+09	2.565760E+13	2.862760E+13	1.781140E+12	3.785130E+13	3.440730E+13
	Co-CLPSO	3.046900E-12	2.988500E+00	3.314700E+00	5.749900E-12	5.105900E+01	9.175900E+01
	SAMO-GA	2.825000E-20	8.574500E+02	1.792600E+03	8.384400E-01	1.749600E+04	3.411400E+04
	ECHT-ARMOR-DE	0.000000E+00	2.824600E+00	1.600000E+00	1.171600E-04	3.131600E+08	1.200000E+09

Table D.17: Function values obtained by MEA-LS-R2S, MJADE-R2S, MLS-MODE, DEbavDBmax , SAMODE, DE-DBmax, SAMO-DE, eABC, Co-CLPSO, SAMO-GA,ϵDEg, and ECHT-ARMOR-DE for CEC2010 for c16-c18

		10D			30D		
problem	Alg.	Best	Mean	Std.	Best	Mean	Std.
c16	MEA-LS-R2S	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00
	MJADE-R2S	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00
	MLS-MODE	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00
	DEbavDBmax	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
	SAMODE	0.000000E+00	6.469600E-03	1.087042E-02	0.000000E+00	4.160700E-03	7.677900E-03
	ϵDEag	0.000000E+00	3.702054E-01	3.710479E-01	0.000000E+00	2.168404E-21	1.062297E-20
	DE-DBmax	0.000000E+00	1.332800E-01	3.301736E-01	0.000000E+00	0.000000E+00	0.000000E+00
	eABC	0.000000E+00	8.347410E-02	9.106390E-02	6.252940E-02	8.207220E-01	2.569880E-01
	Co-CLPSO	0.000000E+00	5.986100E-03	1.331500E-02	0.000000E+00	5.240300E-16	4.672200E-16
	SAMO-GA	0.000000E+00	1.403400E-03	4.880000E-03	0.000000E+00	3.737000E-03	6.284000E-03
	ECHT-ARMOR-DE	0.000000E+00	2.847800E-02	5.000000E-02	0.000000E+00	0.000000E+00	0.000000E+00
c17	MEA-LS-R2S	0.0000000E+00	3.4512665E-33	3.1223041E-33	5.5901650E-25	1.3289496E-17	3.9772317E-17
	MJADE-R2S	0.0000000E+00	1.5608000E-01	3.7892000E-01	0.0000000E+00	2.5243000E-01	2.9517000E-01
	MLS-MODE	3.3716223E-27	4.4542095E-24	1.1601496E-23	3.3560550E-22	2.3915346E-14	1.1228612E-13
	DEbavDBmax	1.346610E-29	7.053446E-19	1.079196E-18	1.061228E-18	1.811932E-12	5.281940E-12
	SAMODE	0.000000E+00	1.265501E-23	3.217991E-23	0.000000E+00	1.022630E-10	1.455150E-10
	ϵDEag	1.463180E-17	1.249561E-01	1.937197E-01	2.165719E-01	6.326487E+00	4.986691E+00
	DE-DBmax	3.098066E-28	1.154546E-05	4.146688E-05	7.114167E-20	6.849019E-12	1.639527E-11
	eABC	2.533850E-02	3.242900E+00	6.831360E+00	3.267520E+00	2.680100E+01	1.634550E+01
	Co-CLPSO	7.667700E-17	3.798600E-01	4.528400E-01	1.578700E-01	1.391900E+00	4.262100E+00
	SAMO-GA	0.000000E+00	1.270700E-02	1.331029E-02	0.000000E+00	1.343600E-02	1.616200E-02
	ECHT-ARMOR-DE	0.000000E+00	3.697800E-33	3.100000E-33	3.356400E-16	4.033600E-01	3.510000E-01
c18	MEA-LS-R2S	0.0000000E+00	8.4432769E-33	1.8409150E-32	3.9443045E-31	2.4252039E-08	8.4589209E-08
	MJADE-R2S	0.0000000E+00	5.4925000E+02	1.2888000E+03	0.0000000E+00	2.4315000E+03	4.7555000E+03
	MLS-MODE	0.0000000E+00	3.2170734E-31	4.1468535E-31	1.4791142E-31	8.9491903E-05	1.1722656E-04
	DEbavDBmax	4.239528E-25	3.885478E-24	4.336668E-24	1.030097E-18	2.829429E-01	1.356367E+00
	SAMODE	2.236640E-15	1.617890E-14	3.820340E-14	1.375370E-14	4.738410E-14	6.573500E-14
	ϵDEag	3.731440E-20	9.678765E-19	1.811234E-18	1.226054E+00	8.754569E+01	1.664753E+02
	DE-DBmax	8.320589E-25	5.257827E-24	9.154472E-24	1.631766E-17	2.901024E-04	9.250217E-04
	eABC	3.098470E-03	3.470230E+02	3.710760E+02	1.960850E-01	2.933360E+02	3.528430E+02
	Co-CLPSO	7.780400E-21	2.319200E-01	9.955900E-01	6.004700E-02	1.087700E+01	3.716100E+01
	SAMO-GA	4.359000E-17	1.053095E-02	1.540100E-02	2.818290E-01	7.535728E+00	1.051703E+01
	ECHT-ARMOR-DE	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00

Table D.18: Function values obtained by MEA-LS-R2S, MJADE-R2S, MLS-MODE, ADE , EPSDE, SAMODE, DE-Acr, CDELS, CDASA and ISAMODE-CMA for CEC2011 for r01-r04

problem	Alg.	Best	Mean	Std.
r01	MEA-LS-R2S	1.5444180E+04	1.5444180E+04	5.9094187E-06
	MJADE-R2S	1.5444180E+04	1.5444180E+04	1.0855000E-06
	MLS-MODE	1.5444180E+04	1.5444180E+04	6.1846980E-07
	ADE	1.5445380E+04	1.5482070E+04	3.0473660E+01
	EPSDE	1.5400000E+04	1.5500000E+04	1.5500000E+01
	SAMODE	1.5444190E+04	1.5444230E+04	3.7333110E-02
	DE-Acr	1.5445000E+04	1.5446000E+04	8.4385000E-01
	CDELS	1.5446550E+04	1.5446999E+04	1.5449634E+04
	CDASA	1.5461000E+04	1.5511000E+04	4.2337000E+01
	ISAMODE-CMA	1.5444190E+04	1.5444190E+04	1.1438000E-05
r02	MEA-LS-R2S	1.8022057E+04	1.8099598E+04	4.1904473E+01
	MJADE-R2S	1.8028000E+04	1.8120000E+04	5.0666000E+01
	MLS-MODE	1.8050910E+04	1.8104920E+04	3.7112340E+01
	ADE	1.8224720E+04	1.8550260E+04	1.4151470E+02
	EPSDE	1.8100000E+04	1.8100000E+04	4.3900000E+01
	SAMODE	1.8046840E+04	1.8185590E+04	1.0714080E+02
	DE-Acr	1.8028000E+04	1.8096000E+04	3.9913000E+01
	CDELS	1.9007720E+04	1.9120010E+04	1.9185830E+04
	CDASA	1.8942000E+04	1.9323000E+04	2.1267000E+02
	ISAMODE-CMA	1.8026760E+04	1.8104590E+04	3.7545240E+01
r03	MEA-LS-R2S	3.2692510E+04	3.2692963E+04	2.5587740E-01
	MJADE-R2S	3.2692000E+04	3.2694000E+04	9.5705000E+00
	MLS-MODE	3.2797750E+04	3.2890930E+04	4.7772960E+01
	ADE	3.2744200E+04	3.2859400E+04	6.8413650E+01
	EPSDE	3.2600000E+04	3.2700000E+04	3.5900000E+01
	SAMODE	3.2721950E+04	3.2737580E+04	7.7779080E+00
	DE-Acr	3.2730000E+04	3.2790000E+04	3.0447000E+01
	CDELS	3.2928030E+04	3.2958290E+04	3.3031150E+04
	CDASA	3.2966000E+04	3.3181000E+04	1.0457000E+02
	ISAMODE-CMA	3.2694660E+04	3.2699270E+04	9.4083270E+00
r04	MEA-LS-R2S	1.2174208E+05	1.2228089E+05	3.2780024E+02
	MJADE-R2S	1.2145000E+05	1.2157000E+05	8.5076000E+01
	MLS-MODE	1.2281680E+05	1.2448750E+05	7.2852280E+02
	ADE	1.2406900E+05	1.2553230E+05	8.3625570E+02
	EPSDE	1.2800000E+05	1.3100000E+05	2.4700000E+03
	SAMODE	1.2278753E+05	1.2409967E+05	8.0725815E+02
	DE-Acr	1.2313000E+05	1.2389000E+05	4.5430000E+02
	CDELS	1.3590694E+05	1.3771068E+05	1.4118462E+05
	CDASA	1.3242000E+05	1.4666000E+05	6.3435000E+03
	ISAMODE-CMA	1.2161578E+05	1.2222620E+05	3.2839232E+02

Table D.19: Function values obtained by MEA-LS-R2S, MJADE-R2S, MLS-MODE, ADE , EPSDE, SAMODE, DE-Acr, CDELS, CDASA and ISAMODE-CMA for CEC2011 for r05-r08

problem	Alg.	Best	Mean	Std.
r05	MEA-LS-R2S	1.6816543E+06	1.6884982E+06	4.5464334E+03
	MJADE-R2S	1.6649000E+06	1.6685000E+06	3.4583000E+03
	MLS-MODE	1.6769140E+06	1.7812770E+06	4.9848520E+04
	ADE	1.8906710E+06	1.9250980E+06	2.3429040E+04
	EPSDE	1.9100000E+06	1.9200000E+06	1.1800000E+04
	SAMODE	1.8891308E+06	1.9876716E+06	1.2556507E+05
	DE-Acr	1.7124000E+06	1.8386000E+06	4.5074000E+04
	CDELS	1.9209047E+06	1.9464404E+06	1.9573575E+06
	CDASA	1.8828000E+06	2.0375000E+06	2.4110000E+05
	ISAMODE-CMA	1.6932933E+06	1.7047279E+06	7.3628216E+03
r06	MEA-LS-R2S	4.2463168E+04	4.3584413E+04	6.4356329E+02
	MJADE-R2S	4.5730000E+04	4.6586000E+04	5.5149000E+02
	MLS-MODE	4.8705940E+04	5.0643040E+04	1.2705080E+03
	ADE	5.0170600E+04	5.4181520E+04	4.8669640E+03
	EPSDE	5.1100000E+04	5.2200000E+04	7.2400000E+02
	SAMODE	5.0981400E+04	5.3278380E+04	2.0317090E+03
	DE-Acr	4.4847000E+04	4.6321000E+04	1.3962000E+03
	CDELS	6.8146400E+04	7.3806300E+04	8.0638850E+04
	CDASA	5.1210000E+04	5.2017000E+04	3.9028000E+02
	ISAMODE-CMA	4.7110730E+04	4.8765770E+04	7.8542710E+02
r07	MEA-LS-R2S	1.0709458E+06	1.0741581E+06	1.9933780E+03
	MJADE-R2S	9.2271000E+05	9.4068000E+05	8.0356000E+03
	MLS-MODE	1.0544620E+06	1.0684960E+06	4.9241360E+03
	ADE	1.0785260E+06	1.0866840E+06	4.4556550E+03
	EPSDE	1.0600000E+06	1.0700000E+06	2.1300000E+03
	SAMODE	1.0675010E+06	1.0707990E+06	1.2727460E+03
	DE-Acr	1.0354000E+06	1.0490000E+06	2.9044000E+05
	CDELS	2.4228390E+06	2.4461757E+06	2.5662654E+06
	CDASA	1.2683000E+06	1.2717000E+06	1.8724000E+03
	ISAMODE-CMA	1.0597810E+06	1.0698800E+06	8.7739890E+03
r08	MEA-LS-R2S	9.1582841E+05	9.1639120E+05	3.7769095E+02
	MJADE-R2S	9.1692000E+05	9.1733000E+05	3.8198000E+02
	MLS-MODE	9.1578410E+05	9.1618010E+05	2.6227390E+02
	ADE	9.2482150E+05	9.3060260E+05	3.5027700E+03
	EPSDE	9.3900000E+05	9.4300000E+05	2.6300000E+03
	SAMODE	9.2435370E+05	9.2580000E+05	6.7159130E+02
	DE-Acr	9.2280000E+05	9.2393000E+05	7.4822000E+02
	CDELS	9.8301754E+05	1.0188345E+06	1.2755968E+06
	CDASA	9.4142000E+05	9.4569000E+05	3.6946000E+03
	ISAMODE-CMA	9.2665670E+05	9.2820880E+05	7.3186610E+02

Table D.20: Function values obtained by MEA-LS-R2S, MJADE-R2S, MLS-MODE, ADE , EPSDE, SAMODE, DE-Acr, CDELS, CDASA and ISAMODE-CMA for CEC2011 for r09-r10

problem	Alg.	Best	Mean	Std.
r09	MEA-LS-R2S	9.1665633E+05	9.1804031E+05	8.2400159E+02
	MJADE-R2S	9.1749000E+05	9.1837000E+05	5.2993000E+02
	MLS-MODE	9.1970990E+05	9.2427600E+05	1.6122740E+03
	ADE	9.2832270E+05	9.9296430E+05	1.9642450E+05
	EPSDE	9.4000000E+05	9.9000000E+05	4.1400000E+04
	SAMODE	9.2432330E+05	9.2602220E+05	7.2979820E+02
	DE-Acr	9.2717000E+05	9.3001000E+05	1.6287000E+03
	CDELS	1.4593034E+06	1.5374957E+06	1.8722742E+06
	CDASA	1.1044000E+06	1.4012000E+06	1.7944000E+05
	ISAMODE-CMA	9.2867730E+05	9.3071350E+05	1.3829670E+03
r10	MEA-LS-R2S	9.2611291E+05	9.2748192E+05	7.4658811E+02
	MJADE-R2S	9.2737000E+05	9.2926000E+05	9.6017000E+02
	MLS-MODE	9.2689750E+05	9.3485490E+05	4.4514900E+03
	ADE	9.2714540E+05	9.3052880E+05	3.0638130E+03
	EPSDE	9.3900000E+05	9.4300000E+05	2.6300000E+03
	SAMODE	9.2437627E+05	9.2584595E+05	7.6154770E+02
	DE-Acr	9.2303000E+05	9.2382000E+05	5.1463000E+02
	CDELS	1.0026198E+06	1.0295312E+06	1.2755968E+06
	CDASA	9.3967000E+05	9.4887000E+05	1.8056000E+04
	ISAMODE-CMA	9.2614541E+05	9.2795464E+05	9.1952510E+02

Appendix E

Two-phase Differential Evolution Framework for Solving Optimization Problems

Karam M. Sallam, Saber M. Elsayed, Ruhul A. Sarker and Daryl L. Essam

School of Engineering and Information Technology
University of New South Wales at Canberra
Australia
Emails: karam.sallam@student.adfa.edu.au;{s.elsayed, r.sarker, d.essam}@adfa.edu.au

***Abstract*—Over the last two decades, different differential evolution (DE) variants have been successfully used to solve different optimization problems. However, no single DE algorithm has consistently been the best for solving a wide range of them. In the literature, this drawback has been tackled by using multiple DE operators in a single framework. However, utilizing a problem's landscape in the design of an efficient selection mechanism to emphasize the best-performing DE variant has not yet been thoroughly explored. Motivated by this fact, in this paper, a new two-phase (exploration and exploitation) multi-operator DE algorithm is proposed. It starts with the exploration phase, dynamically placing emphasis on the best-performing DE based on two landscape indicators and its performance history, and then repeats this process during the exploitation phase. To judge the performance of this algorithm, a variety of real-world optimization problems taken from different disciplines are solved. According to the results obtained, this algorithm shows superior performance to those of state-of-the-art algorithms.**

***Index Terms*—differential evolution, multi-operator, landscape indicators, real-world problems**

I. Introduction

Many real-world problems in different fields, such as operations research, biology, data mining and engineering design, can be formulated as optimization problems with one or more objective functions subject to some constraints.

Due to the importance of solving these problems, researchers and practitioners have proposed exact methods. However, as they usually encounter many challenges, such as (1) requiring specific mathematical properties, such as convexity, continuity and differentiability, to be used; and/or (2) maybe needing the problem to be simplified by making various assumptions for the convenience of mathematical modeling [1], their use in the majority of real-world cases is unverified. For these reasons, many evolutionary algorithms (EAs) have been proposed and implemented in order to tackle real-world optimization problems. However, although they do not require specific mathematical properties to be satisfied, are flexible to dynamic changes, can handle evaluating each solution in parallel, have the capability for self-organization and are more broadly applicable in practice, there is no guarantee that they can reach an optimal solution and the quality of their solutions depends on their parameter settings. Indeed, selecting an appropriate algorithm for solving an optimization problem is not an easy task.

The family of EAs contains various algorithms, such as DE [2], genetic algorithms (GAs) [3] and evolution strategy (ES) [4], with the major difference among them the ways in which they produce new solutions. Of EAs, DE has gained popularity for solving continuous optimization problems [5][6]. However, there is no guarantee that a DE algorithm, which performs well for one or a certain class of problems will work well for another or a range of problems. One reason for this is the variability of the underlying mathematical properties of optimization problems.

To overcome this limitation, multi-methods and/or multi-operator algorithms, in which more than one algorithm and/or search operator are used in a single framework with a mechanism for emphasizing the best-performing one during the search process [7], [8], have been developed. In the literature, there are different categories of these methods, such as ensemble-based [9][10], hyper-heuristics [11], mutli-methods [12][13], multi-operators [7] and heterogeneous [14] approaches. They all consist of a pool of different algorithms and/or operators, and use a selection mechanism to emphasize the best-performing algorithm and/or operator during the search process based on different criteria, such as a re-enforcement learning mechanism [15][16], improvement in the solution quality and/or constraint violations and/or feasibility rate [8], and convergence differences and progress ratios [17].

However, determining a way of combining these operators and/or methods is still challenging. Also, utilizing a problem's landscape to design an efficient selection mechanism has not been comprehensively explored. Although it has been noted that the pool of operators is always fixed during the evolutionary process, this may lead to consuming too much computational effort because of poorly performing DE operators. Therefore, in this paper, we investigate using landscape metrics to obtain information about the characteristics of problems, such as modality, separability, etc., and place emphasis on better-performing DE operators, and changing the pool of DE operators based on the optimization phase.

Consequently, a novel two-phase multi-operator DE with a landscape selection mechanism, referred to as TP-MODE, is proposed in this paper. In TP-MODE the whole evolutionary process is divided into two phases based on the number of fitness evaluations. In the first, a group of DE mutation strategies which have the capability to maintain diversity are chosen to generate trial vectors and, in the latter, other DE mutation strategies characterized by their ability to converge fast are chosen to produce offspring. Simultaneously, a landscape metric is used to place more emphasis on the best-performing operator in each phase. TP-MODE is tested on 22 benchmark test functions taken from the IEEE CEC2011 competition for solving real-world applications [18]. The experimental results imply that the performance of TPMODE is better than those of four other state-of-the-art DE variants.

The rest of this paper is organized as follows: in Section II, a review of DE algorithms and operators, and some landscape measures is presented. Section III describes the proposed framework; the simulation results for the benchmark problems and values of the parameters are provided in Section IV; and finally, Section V discusses the conclusions drawn from this study and possible future research directions.

II. DE AND THE RELATED WORK

In this section, a literature review of DE and the concept of landscape analysis are discussed.

A. DE algorithm

Generally speaking, DE algorithm has shown good performance for solving optimization problems. It has three main operations (mutation, crossover, and selection) to evolve a population of individuals during the search process. DE starts with a uniform random population, such that

$$\overrightarrow{X}_{i,G} = x_{i,G}^{min} + (x_{i,G}^{max} - x_{i,G}^{min}) \times rand(1, D) \qquad (1)$$

where $i \in NP$, $j = 1, 2, ..., D$, NP is the number of individuals, D the dimension of the problem and $rand$ a random real number uniformly generated between 0 and 1, and G the generation number.

1) mutation: Before the mutation operator is applied, every vector $\overrightarrow{X}_{i,G}$ in the population is considered a parent vector (target vector), and corresponding to each parent vector, a mutant vector $\overrightarrow{V}_{i,G} = (v_{i,G}^1, v_{i,G}^2, ..., v_{i,G}^D)$ is produced by the weighted difference between two random vectors to a third vector (the base vector) from the current population (D is the number of decision variables), such that

$$\overrightarrow{V}_{i,G} = \overrightarrow{X}_{r_1^i,G} + F \times (\overrightarrow{X}_{r_2^i,G} - \overrightarrow{X}_{r_3^i,G}) \qquad (2)$$

where $r_1^i \neq r_2^i \neq r_3^i$ are random integer numbers selected from the range$[1, NP]$, which are all different from the index i, the scaling factor F is the control parameter for scaling the difference vectors and G the generation number.

2) crossover: Following the mutation operation, a trial vector $\overrightarrow{U}_{i,G} = \{u_{i,G}^1, u_{i,G}^2, ..., u_{i,G}^D\}$ is produced by crossing a parent vector and its corresponding trial vecto, such that

$$u_{i,G}^j = \begin{cases} v_{i,G}^j & \text{if } (rand(0,1) \leq CR) \text{ or } (j = j_{rand}) \\ x_{i,G}^j & \text{otherwise} \end{cases} \qquad (3)$$

where $rand(0, 1)$ a random real number uniformly generated between 0 and 1, CR the crossover control parameter and $j_{rand} \in [1, 2, ..., D]$ a randomly chosen index which ensures that the trail vector, $\overrightarrow{U}_{i,G}$ obtains at least one element from the mutant vector, $\overrightarrow{V}_{i,G}$.

3) selection: To decide whether parent $\overrightarrow{X}_{i,G}$ or child $\overrightarrow{U}_{i,G}$ vectors survive to the next generation a selection operation is employed. This operation is performed using

$$\overrightarrow{X}_{i,G+1} = \begin{cases} \overrightarrow{U}_{i,G} & \text{if } f(\overrightarrow{U}_{i,G}) \leq f(\overrightarrow{X}_{i,G}) \\ \overrightarrow{X}_{i,G} & \text{otherwise} \end{cases} \qquad (4)$$

where $f(\overrightarrow{U}_{i,G})$ and $f(\overrightarrow{X}_{i,G})$ are the objective functions of the child and parent, respectively. Similar to other EAs algorithms, DE repeats the aforementioned steps, generation after generation, to evolve its population of solutions until some specific termination condition is met.

B. Improved DE Algorithms

In this section, commonly used improved variants of DE are discussed.

As mentioned earlier, in DE the quality of the obtained solutions depends on its parameter (population size NP, crossover rate CR, and scaling factor F) settings. Thus, in the literature there are some different adaptation schemes that have been proposed, and we will review some of them in this section.

A fuzzy adaptive DE algorithm (FADE), in which fuzzy logic controllers were used to adapt the parameters F and CR, was proposed by Liu and Lampinen [19]. An adaptive DE algorithm (ADE) was proposed by Zaharie and Daniela [20], in which F and CR were adapted with regard to population diversity. A self-adaptive (jDE) strategy to adapt the F and CR values of the DE algorithm, was proposed by Breast et al. [21]. In jDE, both F and CR are also encoded into the individuals and their values survive to the next generation with a particular probability, τ_1 and τ_2, and otherwise, F and CR values are randomly re-initialized to new values within the given range for the next generation [22]. An adaptive DE algorithm with an optional external memory (JADE) was proposed by Zhang et al. [23], in which the control parameters CR and F of each individual were automatically updated according to previously successful experiences. Also, it implements a "DE/current-to-$p$best" mutation strategy, with an external archive, which some of the good solutions are stored on it to avoid premature convergence and also diversify the population. Tanabe and Fukunaga [24] introduced success-history based parameter adaptation for differential evolution (SHADE), which is an improved version of JADE, uses a history based parameter adaptation method. The L-SHADE [25] algorithm is a SHADE algorithm that uses linear population size reduction (LPSR) to dynamically re-size the population during a run. LPSR reduces the population linearly as the number of fitness evaluations increases. L-SHADE showed good performance, in comparison with other algorithm over a set of unconstrained optimization problems. A new Sinusoidal DE (SinDE), in which new sinusoidal formulas was used to automatically adjust the values of F and CR, was proposed by Draa et al. [26]. It was claimed by the author that, SinDE especially gives good results for multi-modal and composition functions. To dynamically select the a best compromise of parameters (i.e. F, CR and NP) for solving a specific problem in each run, Sarker et al. [27] designed a new mechanism to do it. The experimental results demonstrated its superior performance over the state-of-the-art DE variants.

To follow our discussion about DE algorithms and as mentioned earlier that no single DE algorithm was able to solve all kind of optimization problems, a considerable number of DE algorithms, which uses more than one mutation strategy or incorporated with other algorithms, have been proposed to solve this drawback and here we will review some of them.

Sallam et. al [5] proposed a neuro-dynamic differential evolution algorithm for solving CEC2015 single objective optimization problems. An adaptive mechanism was proposed for the appropriate use of L-SHADE and neuro-dynamic during the search process. A self-adaptive DE (SaDE) was proposed by Qin et al. [28] for solving unconstrained real-parameter optimization. In SaDE, both the trial vector generation strategy and its associated control parameter values, were gradually self-adapted by learning from their previous search experience. A composite DE (CoDE) was proposed by Wang et al. [29] for solving optimization problems. It uses three mutation strategies randomly with three fixed control parameter settings for generating a new trial vector at each generation. To generate a new solution, three vectors were generated, then the best one among them was selected to enter the next generation. From the experimental results, it was concluded that CoDE is a promising optimization algorithm for solving optimization problems. A self adaptive multi-operator differential evolution (SAMO-DE), for solving constrained optimization problems, was proposed by Elsayed et al. [8]. In SAMO-DE, each operator had its own sub-population which was evolved by a different DE operator. Based on an improvement measure in which the solution quality, constraint violation and feasibility ratio were used to calculate the success of each operator, the number of individuals in each sub-population was adaptively updated and more emphasis was given to the operator with the highest success. The results showed that SAMO-DE performed better than other-state-of-the-art algorithms. Using a mix of four different DE mutation strategies, two crossover operators and two constraint handling techniques for constrained optimization problems was proposed by Elsayed et al. [30]. The experimental results show that the proposed algorithm is better than other state-of-the-art algorithms. Mallipeddi et al. [9] designed the ensemble differential evolution algorithms (EPSDE). In these algorithms, a pool of distinct DE mutation strategies, crossover strategies and a pool of associated values for the control parameters not only coexist throughout the evolutionary process, but also compete with each other to produce offspring. All of the above mentioned methods did not incorporate any landscape information in the selection phase. For more inforatiomation regarding DE algorithms, readers are refered to [31] [32].

In the recent past, researchers and practitioners have used fitness landscape to determine and select an appropriate algorithm or operator for solving optimization problems. In [33], the authors suggested using a cost sensitive learning model to select the best algorithm amongst a set of four algorithms. The four algorithms were selected manually from the algorithms solving Black-Box Optimization Benchmarking (BBOB) in 2009 and 2010 [34][35]. The functions in $10D$ were characterized using 19 measurements extracted by the exploratory landscape analysis (ELA) technique. In their work, low-level features [36] were obtained by systematic sampling of the functions on the search space. Then separability, modality, and global structures of the optimization problem were measured as the first step to characterize the landscape (this step was done offline). Next, a machine learning model was constructed to select the best algorithm from the portfolio. Based on two different cross validation schemes, the model was validated. However, the results may not be general for a knowledge base with problems of different dimensionality such as $2D$, $3D$, $5D$, and $20D$. Also, because the low level features were extracted in a different step, the authors did not add the computational cost for calculating the features to the function evaluations. Further, as the selection of the four algorithms was done manually, this weakened its validation on unobserved problems. In [37], decision trees were employed to predict the failure of seven different particle swarm optimization algorithms (PSO), by using a number of fitness landscape metrics. In [38], an adaptive operator selection mechanism, based on a set of four fitness landscape analysis techniques, was used to train an online regression learning model (dynamic weighted majority), which was used to predict the weight of each operator in each generation. Their proposed mechanism was used to determine the most suitable crossover operator, among four crossover operators, to solve a set of Capacitated Arc Routing Problem (CARP) instances. The authors used instantaneous reward, in which the reward was considered as the value computed at the last evaluation. In comparison with some of the-state-of-the-art algorithms, the algorithm did not show significant benefit.

III. Two-phase multi-operator DE

In this section, the two-phase multi-operator DE with a landscape selection mechanism is presented.

A. TP-MODE

In designing the proposed algorithm, two considerations taken into account are that, (1) as no single mutation strategy might be able to solve all kinds of problems, it is beneficial to use a multi-operator [8], [13], and (2) the new algorithm should exhibit an exploration capability in the early phase of the evolution and fast convergence in a later one.

The framework of TP-MODE is presented in Algorithm 1.

Generally speaking, the entire evolutionary process is divided into two phases, exploration and exploitation. In the first, TP-MODE focuses on improving its exploration by using two DE mutation strategies which maintain diversity and, in the second, two other DE strategies which can converge quickly are used. Initially, NP individuals are randomly generated by equation 15. Subsequently, based on the type of phase, two mutation strategies, in which each operator is randomly assigned the same number of individuals (n_{op}, $\forall op = \{1, 2\}$), are implemented. Then, a new solution is generated using its assigned DE variant and evaluated according to the fitness function value. If it is better than its parent, it survives to the next generation; otherwise its parent is retained for the next generation. At the same time, two landscape metrics (the information landscape negative searchability metric/ searchability indicator (SI), fitness distance correlation (FDC)) and one performance measure (success rate (SR)) are calculated for each operator, with the overall normalized measure computed from these values using equation (13) based on which, n_{op}, $\forall op = \{1, 2\}$ is updated. As this process may abandon certain operators which could be useful in later stages of the evolutionary process, we set a minimum number of individuals for each operator. The above process is repeated until the maximum number of fitness function evaluations (MAX_{FES}) is reached. In this paper, the algorithm switches from the first to second phase based on a predefined number of FEs ($\frac{MAX_{FES}}{3}$).

In the following subsections, TP-MODE's components are discussed in more detail.

1) Two-phase mutation strategies:: As, in the first phase, the main objective is to maintain exploration capability of TP-MODE to avoid premature convergence, DE/rand/1/bin is chosen to be placed in its pool while the second variant is DE/ϕbest/1, where ϕ is set to $[1, 0.6 * NP]$, as described in [39].

On the other hand as, in the second phase, the aim is to speed up convergence; information from the best solution is of great interest. Therefore, DE/current-to-ϕbest/1/bin/with archive and DE/ϕbest/1 are used, withϕ is equal to 0.1 for both operators.

2) Selection mechanism: As previously mentioned, in each phase, two operators are used, with TP-MODE placing emphasis on the best-performing one based on three indicators: (1) the SR; (2) SI; and (3) FDC.

- SR
  The SR of each operator (SR_{op}) is defined as the number of successful offspring generated by a search operator (op), divided by the number of individuals assigned to op as :

$$SR_{op} = \frac{\text{Number of successful offsprings}}{n_{op}}, \forall op = \{1,\ 2\} \tag{5}$$

Then, the normalized value for the SR is calculated using:

$$NM_{SR_{op}} = \frac{\overline{SR}_{op}}{\sum_{OP=1}^{m} \overline{SR}_{op}} \tag{6}$$

where the $\overline{SR}_{op}$ is the mean value of the SR of operator op.

- SI
  The searchability of the algorithm, which is the capability of its search operator to move to a region of the search space with a better fitness value, is measured by computing an information landscape metric based on the difference between the information landscape vector of the problem to be solved and a reference landscape vector. The reference landscape is the landscape of a function tthat can be easily optimized by any optimization algorithm in any dimension [40] and the reference function $f_{ref}(\overrightarrow{x})$ is determined using:

$$f_{ref}(\overrightarrow{x}) = \sum_{j=1}^{D} (x_j - x_j^*)^2 \tag{7}$$

where $\overrightarrow{x}_i^*$ is the best individual in the sample.
An information matrix for a minimization problem $M = [a_{i,j}]$ is constructed using:

$$a_{i,j} = \begin{cases} 1 & \text{if } f(x_i) < f(x_j) \\ 0.5 & \text{if } f(x_i) = f(x_j) \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

Not all the entries in the information landscape are required to define it [40] [41]. There are duplicates in the entries due to symmetry (the lower triangle should be omitted), the entries on the diagonal are always 0.5 (and should be omitted), and the row and column of the optimum solution should also be omitted. Therefore, the information matrix can be reduced to a vector $LS = (ls1, ls_2, ..., ls_{|LS|})$, where the number of elements in LS is: $|LS| = \frac{(NP-1)\times(NP-2)}{2}$.
After constructing the landscape vector of the problem to be optimized (LS_f) and the vector landscape of the reference function (LS_{ref}), the SI is computed as:

$$LD_{op} = \frac{1}{|LS_f|} \times \sum_{i=1}^{|LS_f|} |ls_{1i} - ls_{2i}| \tag{9}$$

When LD is close to 0, the problem is considered easy and difficult when LD is close to 1.
The normalized value for the LD is then calculated as:

$$NM_{LD_{op}} = \frac{(1 - M_{LD_{OP}})}{\sum_{OP=1}^{m}(1 - M_{LD_{OP}})} \tag{10}$$

where M_{LD} is the mean value of information landscape.

- FDC
  The FDC, which was proposed by Jones and Forrest [42] and measures the correlation between the objective value and distance to the nearest optimum in the search domain is another method to determine a problem's difficulty [43].
  Given a set of solutions $X = \{\overrightarrow{x}_1, \overrightarrow{x}_2, ..., \overrightarrow{x}_{NP}\}$ and their objective function values $F = \{f_1, f_2, ..., f_{NP}\}$, the FDC value is computed by:

$$FDC_{op} = \frac{\sum_{i=1}^{NP}(f_i - \bar{f})_{op} \times (d_i - \bar{d})_{op}}{n \times (\sigma_f \times \sigma_d)_{op}} \quad (11)$$

  where d_i is the distance between the i^{th} individual and the best solution in the population, $\bar{f}$, $\bar{d}$, σ_f, and σ_d are the mean and standard deviations of fitness function and distance, respectively.
  For a minimization problem, a value of FDC_{op} closes to 1 means that the problem is relatively easy, and one near 0 that it is difficult.
  The normalized FDC (NM_{FDC}) is calculated using equation by:

$$NM_{FDC_{op}} = \frac{\overline{FDC}_{op}}{\sum_{OP=1}^{m}\overline{FDC}_{op}} \quad (12)$$

  where the $\overline{FDC}_{op}$ is the mean value of FDC of operator op.

Subsequently, the overall normalized value for each operator is computed using:

$$ONM_{op,G} = \frac{NM_{SR_{op}} + NM_{LD_{op}} + NM_{FDC_{op}}}{\sum_{op=1}^{m}(NM_{SR_{op}} + NM_{LD_{op}} + NM_{FDC_{op}})}, \quad (13)$$

After determining the overall normalized value for each operator, the number of individuals for each operator in each generation, is calculated by:

$$n_{op,G} = MSS + \frac{ONM_{op,G}}{\sum_{op=1}^{m} ONM_{op,G}} \times (NP - MSS \times m) \quad (14)$$

where $n_{op,G}$ is the number of individuals operator op will evolve at generation G, MSS is the minimum number of individuals for operator op at generation G.

3) Population Initialization and Updating Method : In the initialization phase, the Latin Hypercube Design, which is a type of stratified sampling, is used to generate the initial population because of its capability to generate a sample of points thatmore efficiently covers the whole search region .

The initialization is performed using:

$$x_{i,j} = x_i^{min} + (x_i^{max} - x_i^{min}) \times lhd(1, NP) \\ i \in NP\, and\, j = 1,\, 2,\, ...,\, D \quad (15)$$

where lhd is a function that generates random numbers using the Latin Hypercube Design.

Moreover, a linear population size reduction schema is used to adaptively re-size the NP during the evolutionary process [25], as follows:

$$NP_{t+1} = round[(\frac{NP^{min} - NP^{init}}{MAX_{FES}}) \times FES + NP^{init}] \quad (16)$$

where NP^{min} is the smallest number of individuals that the proposed algorithm can use. FES is the current number of fitness evaluations, MAX_{FES} is the maximum number of fitness evaluations.

4) Managing F and CR: As it is a fact that the performance of the DE algorithm is affected by the values of the control parameters (F and CR) in each phase [44] [45], we use a different adaptation mechanism for their values. In the first, the mechanism proposed by Elsayed et al. [46] is adopted as it has shown its capability to maintain diversity.

Initially, for each individual in the population, two sets of control parameters, $F \in N(0.5,\, 15)$ and $CR \in (0.5,\, 0.15)$, are generated using a Gaussian distribution with mean and standard deviation values of 0.5 and 0.15, respectively. Then, as in [7], to generate new offspring, F and CR are calculated, respectively,:

$$F = F_{r1} + N(0,\, 0.5) \times (F_{r2} - F_{r3}) + N(0,\, 0.5) \\ (F_{r4} - F_{r5}) + N(0,\, 0.5) \times (F_{r6} - F_{r7}) \quad (17)$$

$$CR = CR_{r1} + N(0,\, 0.5) \times (CR_{r2} - CR_{r3}) + N(0,\, 0.5) \\ (CR_{r4} - CR_{r5}) + N(0,\, 0.5) \times (CR_{r6} - CR_{r7}) \quad (18)$$

where $r_1^i \neq r_2^i \neq r_3^i \neq r_4^i \neq r_5^i \neq r_6^i \neq r_7^i$ are random integer numbers selected from the range$[1, NP]$.

If the value of F is less than 0.1 or larger than 1, it is truncated to 0.1 and 1, respectively, while if the value of CR is less than 0.01 or larger than 1, it is truncated to 0.01 and 1, respectively.

In the second phase, inspired by [47], a scaling factor pool (i.e., F_{pool} = [0.6, 0.8, 1.0]) and a crossover control parameter pool (i.e., CR_{pool} = [0.1, 0.2, 1.0]) are established in DE. Then, at each generation, F and CR are randomly chosen from F_{pool} and CR_{pool}, respectively.

IV. Experimental results

In this section, the performances of the proposed algorithm for solving a benchmark test set of 22 problems taken from the CEC2011 competition considering real-world applications is discussed [18]. The proposed algorithm was run following the guidelines of the competition that required 25 independent runs for each test problem with up to $FES_{MAX} = 150,000$ fitness evaluations. It was coded using Matlab R2014a and run on a PC with a 3.4 GHz Core I7, 16 GB RAM, and windows 7.

Algorithm 1 Proposed framework

Generate an initial population (X) of size NP using Latin Hypercube Design; $FES \leftarrow 0$;
Evaluae X;
$FES \leftarrow FES + NP$;
n_{op_1}=n_{op_2}=$\frac{NP}{2}$;
while $FES \leq MAX_{FES}$ **do**
  if $FES \leq limit$ **then**
    {Phase1}
    Set operators $op_1 = DE/\phi best/1$; $op_2 = DE/rand/2$;
    Calculate F and CR using equations (17), and (18);
  else
    {Phase2}
    Set operators $op_1 = DE/current - to - \phi best/1/Archive$; $op_2 = DE/\phi best/1$;
    Calculate control parameters F and CR as in the last paragraph of Sec III-A4 ;
  end if
  Evolve individuals using its assigned operators op;
  Compute $NM_{SR_{op}}$, $NM_{LD_{op}}$, and $NM_{FDC_{op}}$ using equations (6), (10), and (11);
  Update n_{op_1} and n_{op_2} using equation (14);
  $FES \leftarrow FES + NP$;
  if $FES == limit$ **then**
    n_{op_1}=n_{op_2}=$\frac{NP}{2}$;
  end if
  Update the NP using equation (16);
end while

A. Algorithm Parameters and Operators

The default values of NP^{init} of 200 and NP^{min} of 10 were set based on our empirical analysis, with φ set at a value of 0.6 for DE/φbest/1 to maintain diversity, and 0.1 for the other two variants, to speed up the convergence rate. For DE/current-to-ϕbest/1/bin/with archive the archive rate (A) was set at a value of 1.4, $limit$ the maximum limit for running phase 1, after which phase two started, and set at a value of $\frac{FES_{MAX}}{3}$ was set based on our empirical analysis and this parameter was fixed for all problems.

B. Detailed Results of proposed algorithm

The detailed results (best, worst, median, average, and standard deviation (Std.)) are shown in Table I. The performance of TP-MODE was compared with those of three algorithms, GA-MPC [48] (the winner of the CEC2011 competition), SAMO-DE [7], an ensemble DE algorithm (EPSDE) [9], that uses different strategies, and SHADE [24] . All Algorithms used the same stopping criteria; i.e., $150,000$ fitness function evaluations.

Table II presents the best, mean and standard deviation values of these algorithms. Based on the best fitness values obtained, TP-MODE was better than GA-MPC, SAMO-DE and EPSDE for 7, 12 and 14 problems, respectively, and the same for 9, 7 and 3, respectively, while it was inferior for 6, 3 and 5, respectively. Regarding the average fitness values, TP-MODE was superior to GA-MPC, SAMO-DE and EPSDE for 10, 15 and 15 problems, respectively, obtained the same mean results for only 4, 3 and 2, respectively, and was inferior for 8, 4 and 5, respectively.

As it is also possible to study the statistical difference between any two algorithms using a non-parametric test, the Wilcoxon Signed Rank Test [49] was performed, with the results regarding the best and average fitness values presented in Table III. As a null hypothesis, it was assumed that there was no significant difference between the best and/or mean values of two samples while an/the alternative hypothesis was that there was a significant difference at a 10% significance level. Based on the test results, we assigned one of three signs ($+$, $-$, and $\approx$) for the comparison of any two algorithms (shown in the last column), where the "$+$" sign means the first algorithm was significantly better than the second, the "$-$" sign means that the first algorithm was significantly worse, and the "$\approx$" sign means that there was no significant difference between thim. It is clear that, from Table III that TP-MODE was superior to GA-MPC, SAMO-DE, EPSDE, and SHADE in terms of the average results and very competitive with them regarding the best.

Finally, the Friedman Test was conducted to rank all algorithms, with the results shown in Table IV in which it is clear that TP-MODE was superior to the other three algorithms regarding both the best and average fitness results, followed by GA-MPC, SHADE, EPSDE and SAMODE, respectively.

V. Conclusion and future work

In this study, a new multi-operator DE algorithm for solving real-world application problems was proposed. It focused on diversity without losing convergence through dividing the whole search process into two phases, with two DE mutation operators used in each phase to achieve the aim of the corresponding phase. Furthermore, a new measure based on the quality of solutions and characteristics of the landscape was used to emphasize the best-performing operator in each phase. The performance of the proposed algorithm was tested on 22 real-world application problems taken from the CEC2011 competition, with the results demonstrating its superiority over three other state-of-the-art algorithms.

Possible extensions of this work include obtaining a dynamic balance between convergence and diversity, and analyzing each component in the algorithm's design. Also, using more than one EA may be of great interest.

References

[1] R. Sarker, J. Kamruzzaman, and C. Newton, "Evolutionary optimization (evopt): a brief review and analysis," *International Journal of Computational Intelligence and Applications*, vol. 3, no. 04, pp. 311–330, 2003.

[2] R. Storn and K. Price, "Differential evolution a simple and efficient adaptive scheme for global optimization over continuous spaces, international computer science institute, berkeley," *Berkeley, CA*, 1995.

[3] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," *Machine learning*, vol. 3, no. 2, pp. 95–99, 1988.

[4] I. Rechenberg, "Evolution strategy," *Computational Intelligence: Imitating Life*, vol. 1, 1994.

Table I
THE FUNCTION VALUES OBTAINED BY TP-MODE OVER 25 RUNS AND 150000 FITNESS FUNCTION EVALUATIONS

Problems	Best	Worst	Median	Mean	Std.
P01	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
P02	-2.842253E+01	-2.163764E+01	-2.589359E+01	-2.573171E+01	1.594532E+00
P03	1.151489E-05	1.151489E-05	1.151489E-05	1.151489E-05	0.000000E+00
P04	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
P05.1	-3.684537E+01	-3.163659E+01	-3.479658E+01	-3.465468E+01	1.119970E+00
P05.2	-2.916612E+01	-2.300456E+01	-2.916612E+01	-2.822079E+01	1.385154E+00
P06	5.000000E-01	1.096765E+00	7.510147E-01	7.603013E-01	1.655763E-01
P07	2.200000E+02	2.200000E+02	2.200000E+02	2.200000E+02	0.000000E+00
P08	1.628873E+03	3.059153E+03	2.442968E+03	2.453615E+03	3.755708E+02
P09	-2.184235E+01	-2.164293E+01	-2.164414E+01	-2.165958E+01	5.411104E-02
P10.1	5.116130E+04	5.247562E+04	5.175826E+04	5.180184E+04	3.431911E+02
P10.2	1.068335E+06	1.076211E+06	1.072737E+06	1.072583E+06	1.572728E+03
P11.1	1.544419E+04	1.544419E+04	1.544419E+04	1.544419E+04	1.897504E-05
P11.2	1.812350E+04	1.867086E+04	1.833903E+04	1.835748E+04	1.507295E+02
P11.3	3.271559E+04	3.278829E+04	3.276597E+04	3.276269E+04	1.931411E+01
P11.4	1.259621E+05	1.295006E+05	1.279397E+05	1.278757E+05	1.111803E+03
P11.5	1.881031E+06	1.941075E+06	1.913161E+06	1.913606E+06	1.427120E+04
P12.1	9.362562E+05	9.420340E+05	9.392626E+05	9.392934E+05	1.544130E+03
P12.2	9.400879E+05	1.068943E+06	9.697023E+05	9.788815E+05	3.926205E+04
P12.3	9.362562E+05	9.420340E+05	9.392626E+05	9.392934E+05	1.544130E+03
P13	9.875190E+00	1.797719E+01	1.409832E+01	1.412068E+01	1.616015E+00
P14	8.593443E+00	1.877439E+01	9.827477E+00	1.147226E+01	3.270544E+00

Table II
THE FUNCTION VALUES ACHIVED BY TP-MODE, GA-MPC, SAMO-DE AND EPSDE OVER 25 RUNS AND 150000 FES

	TP-MODE			GA-MPC			SAMO-DE			EPSDE			SHADE		
Problems	Best	Mean	Std.	Best	Mean	Std.	Best	Mean	Std.	Best	Mean	Std.	Best	Mean	Std.
P01	0.00000E+00	0.000000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	1.21203E+00	3.37622E+00	0.00000E+00	1.78000E+00	4.18000E+00	2.73700E-02	7.59030E-01	2.29845E+00
P02	-2.84225E+01	-2.573171E+01	1.59453E+00	-2.84225E+01	-2.77007E+01	4.67305E-01	-2.84225E+01	-2.70698E+01	6.62481E-01	-2.14000E+01	-1.83000E+01	1.65000E+00	-2.44943E+01	-2.30316E+01	7.46050E-01
P03	1.15149E-05	1.151489E-05	0.00000E+00	1.15149E-05	1.15149E-05	0.00000E+00	1.15149E-05	1.15149E-05	0.00000E+00	1.15149E-05	1.15149E-05	1.52000E-19	1.15000E-05	1.15000E-05	5.19000E-21
P04	0.00000E+00	0.000000E+00	0.00000E+00	1.37708E+01	1.38154E+01	1.54600E-01	1.37708E+01	1.39407E+01	2.50222E-01	1.38000E+01	1.67000E+01	3.26000E+00	0.00000E+00	0.00000E+00	0.00000E+00
P05.1	-3.68454E+01	-3.465468E+01	1.11997E+00	-3.68454E+01	-3.50388E+01	8.32925E-01	-3.68439E+01	-3.35947E+01	1.57514E+00	-3.20000E+01	-2.90000E+01	1.84000E+00	-3.66566E+01	-3.60052E+01	6.38070E-01
P05.2	-2.91661E+01	-2.822079E+01	1.38515E+00	-2.91661E+01	-2.74881E+01	1.78214E+00	-2.91661E+01	-2.76347E+01	1.92353E+00	-1.99000E+01	-1.70000E+01	2.68000E+00	-2.91420E+01	-2.90486E+01	1.23890E-01
P06	5.00000E-01	7.603013E-01	1.65576E-01	5.00000E-01	7.48409E-01	1.24914E-01	5.00000E-01	8.16624E-01	1.19367E-01	1.28000E+00	1.42000E+00	7.38000E-02	9.06410E-01	1.12130E+00	8.34400E-02
P07	2.20000E+02	2.200000E+02	0.00000E+00	2.20000E+02	2.20000E+02	0.00000E+00	2.20000E+02	2.20000E+02	0.00000E+00	2.20000E+02	2.20000E+02	0.00000E+00	2.20000E+02	2.20000E+02	0.00000E+00
P08	1.62887E+03	2.453615E+03	3.75571E+02	4.66763E+02	1.22059E+03	3.61119E+02	9.44119E+02	2.26440E+03	8.54213E+02	7.84500E+02	2.52900E+03	1.32800E+03	1.14114E+03	2.22875E+03	7.14195E+02
P09	-2.18424E+01	-2.165958E+01	5.41110E-02	-2.18425E+01	-2.17022E+01	1.16347E-01	-2.18217E+01	-2.16589E+01	1.12958E-01	-2.18000E+01	-1.56000E+01	3.79000E+00	-2.18425E+01	-2.16289E+01	1.40740E-01
P10.1	5.11613E+04	5.180184E+04	3.43191E+02	5.09251E+04	5.20546E+04	4.49912E+02	5.12682E+04	5.23475E+04	5.78021E+02	5.11000E+04	5.22000E+04	7.24000E+02	5.15596E+04	5.32252E+04	3.80380E+03
P10.2	1.06834E+06	1.072583E+06	1.57273E+03	1.06955E+06	1.07338E+06	1.61759E+03	1.07021E+06	1.07313E+06	1.70325E+03	1.06000E+06	1.07000E+06	2.13000E+03	1.07000E+06	1.10000E+06	5.34206E+04
P11.1	1.54442E+04	1.544419E+04	1.89750E-05	1.54442E+04	1.54442E+04	1.75112E-07	1.54442E+04	1.54442E+04	7.24128E-04	1.54000E+04	1.55000E+04	1.55000E+01	1.54442E+04	1.54442E+04	5.57000E-12
P11.2	1.81235E+04	1.835748E+04	1.50730E+02	1.81006E+04	1.82610E+04	6.98163E+01	1.82843E+04	1.85249E+04	1.34481E+02	1.81000E+04	1.81000E+04	4.39000E+01	1.80286E+04	1.81307E+04	5.17471E+01
P11.3	3.27156E+04	3.276269E+04	1.93141E+01	3.27238E+04	3.27698E+04	2.68249E+01	3.27840E+04	3.28462E+04	3.45006E+01	3.26000E+04	3.27000E+04	3.59000E+01	3.27429E+04	3.27600E+04	2.80246E+01
P11.4	1.25962E+05	1.278757E+05	1.11180E+03	1.29213E+05	1.33230E+05	1.87880E+03	1.30037E+05	1.32707E+05	1.14873E+03	1.28000E+05	1.31000E+05	2.47000E+03	1.26951E+05	1.29231E+05	1.58563E+03
P11.5	1.88103E+06	1.913606E+06	1.42712E+04	1.92026E+06	1.95332E+06	1.40836E+04	1.91925E+06	1.97709E+06	3.53818E+04	1.91000E+06	1.92000E+06	1.18000E+04	1.88000E+08	1.91000E+06	1.40754E+04
P12.1	9.36256E+05	9.392934E+05	1.54413E+03	9.49500E+05	9.71289E+05	1.03874E+04	9.43215E+05	9.48921E+05	3.91430E+03	9.39000E+05	9.43000E+05	2.63000E+03	9.37267E+05	9.40475E+05	2.16143E+03
P12.2	9.40088E+05	9.788815E+05	3.92621E+04	9.72102E+05	1.05630E+06	5.70416E+04	1.00815E+06	1.20594E+06	9.95328E+04	9.40000E+05	9.90000E+05	4.14000E+04	9.39771E+05	9.52462E+05	1.21308E+04
P12.3	9.36256E+05	9.392934E+05	1.54413E+03	9.46598E+05	9.75109E+05	1.18489E+04	9.47654E+05	9.58792E+05	5.99265E+03	9.39000E+05	9.43000E+05	2.63000E+03	9.34264E+05	9.40667E+05	2.99161E+03
P13	9.87519E+00	1.412068E+01	1.61602E+00	7.09556E+00	1.28182E+01	3.24134E+00	6.94322E+00	1.10675E+01	2.65228E+00	1.66000E+01	1.88000E+01	1.67000E+00	1.41104E+01	1.75903E+01	1.42842E+00
P14	8.59344E+00	1.147226E+01	3.27054E+00	8.39869E+00	9.35934E+00	9.45433E-01	8.61063E+00	1.09952E+01	2.38898E+00	8.78000E+00	1.39000E+01	4.08000E+00	1.31146E+01	1.99439E+01	2.87591E+00

Table III
COMPARISON SUMMARY BETWEEN TP-MODE AND OTHER STATE-OF-THE-ART ALGORITHMS

Criteria	Algorithms	better	equal	worse	Dec.
Best	TP-MODE vs. GA-MPC	8	8	6	≈
	TP-MODE vs. SAMO-DE	13	7	2	+
	TP-MODE vs. EPSDE	12	3	7	≈
	SHADE	11	8	3	≈
Mean	TP-MODE vs. GA-MPC	10	4	8	+
	TP-MODE vs. SAMO-DE	15	3	4	+
	TP-MODE vs. EPSDE	17	2	3	+
	SHADE	11	6	5	≈

Table IV
FRIEDMAN'S TEST RESULTS

Algorithm	Best results rank	Mean results rank	Overall mean rank	Order
TP-MODE	**2.43**	**2.32**	**2.38**	**1**
GA-MPC	2.73	2.84	2.79	2
SAMO-DE	3.50	3.50	3.50	5
EPSDE	3.18	3.70	3.44	4
SHADE	3.16	2.64	2.90	3

[5] K. M. Sallam, R. A. Sarker, D. L. Essam, and S. M. Elsayed, "Neurodynamic differential evolution algorithm and solving cec2015 competition problems," in *Evolutionary Computation (CEC), 2015 IEEE Congress on*. IEEE, 2015, pp. 1033–1040.

[6] S. Das and P. N. Suganthan, "Differential evolution: a survey of the state-of-the-art," *Evolutionary Computation, IEEE Transactions on*, vol. 15, no. 1, pp. 4–31, 2011.

[7] S. M. Elsayed, R. A. Sarker, and D. L. Essam, "Differential evolution with multiple strategies for solving cec2011 real-world numerical optimization problems," in *2011 IEEE Congress of Evolutionary Computation (CEC)*. IEEE, 2011, pp. 1041–1048.

[8] S. M. Elsayed, R. A. Sarker, and D. L. Essam, "Multi-operator based evolutionary algorithms for solving constrained optimization problems," *Computers & operations research*, vol. 38, no. 12, pp. 1877–1896, 2011.

[9] R. Mallipeddi, P. N. Suganthan, Q. K. Pan, and M. F. Tasgetiren, "Differential evolution algorithm with ensemble of parameters and mutation strategies," *Applied Soft Computing*, vol. 11, pp. 1679–1696, 2011.

[10] G. Iacca, F. Caraffini, and F. Neri, "Multi-strategy coevolving aging particle optimization," *International journal of neural systems*, vol. 24, no. 01, p. 1450008, 2014.

[11] E. K. Burke, M. Gendreau, M. Hyde, G. Kendall, G. Ochoa, E. Özcan, and R. Qu, "Hyper-heuristics: A survey of the state of the art," *Journal of the Operational Research Society*, vol. 64, no. 12, pp. 1695–1724, 2013.

[12] J. A. Vrugt, B. A. Robinson, and J. M. Hyman, “Self-adaptive multi-method search for global optimization in real-parameter spaces,” *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 243–259, 2009.

[13] S. M. Elsayed, R. A. Sarker, D. L. Essam, and N. M. Hamza, “Testing united multi-operator evolutionary algorithms on the cec2014 real-parameter numerical optimization,” in *Evolutionary Computation (CEC), 2014 IEEE Congress on*. IEEE, 2014, pp. 1650–1657.

[14] F. V. Nepomuceno and A. P. Engelbrecht, “A self-adaptive heterogeneous pso for real-parameter optimization,” in *2013 IEEE Congress on Evolutionary Computation*. IEEE, 2013, pp. 361–368.

[15] P. Auer, N. Cesa-Bianchi, and P. Fischer, “Finite-time analysis of the multiarmed bandit problem,” *Machine learning*, vol. 47, no. 2-3, pp. 235–256, 2002.

[16] K. Li, A. Fialho, S. Kwong, and Q. Zhang, “Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition,” *Evolutionary Computation, IEEE Transactions on*, vol. 18, no. 1, pp. 114–130, 2014.

[17] L.-A. Gordián-Rivera and E. Mezura-Montes, “A combination of specialized differential evolution variants for constrained optimization,” in *Advances in Artificial Intelligence–IBERAMIA 2012*. Springer, 2012, pp. 261–270.

[18] S. Das and P. N. Suganthan, “Problem definitions and evaluation criteria for cec 2011 competition on testing evolutionary algorithms on real world optimization problems,” *Jadavpur University, Nanyang Technological University, Kolkata*, 2010.

[19] J. Liu and J. Lampinen, “A fuzzy adaptive differential evolution algorithm,” *Soft Computing*, vol. 9, no. 6, pp. 448–462, 2005.

[20] D. Zaharie, “Parameter adaptation in differential evolution by controlling the population diversity,” in *Proceedings of the international workshop on symbolic and numeric algorithms for scientific computing*, 2002, pp. 385–397.

[21] J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer, “Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems,” *IEEE transactions on evolutionary computation*, vol. 10, no. 6, pp. 646–657, 2006.

[22] A. Zamuda and J. Brest, “Self-adaptive control parameters randomization frequency and propagations in differential evolution,” *Swarm and Evolutionary Computation*, vol. 25, pp. 72–99, 2015.

[23] J. Zhang and A. C. Sanderson, “Jade: adaptive differential evolution with optional external archive,” *Evolutionary Computation, IEEE Transactions on*, vol. 13, no. 5, pp. 945–958, 2009.

[24] R. Tanabe and A. Fukunaga, “Evaluating the performance of shade on cec 2013 benchmark problems,” in *Evolutionary Computation (CEC), 2013 IEEE Congress on*. IEEE, 2013, pp. 1952–1959.

[25] R. Tanabe and A. S. Fukunaga, “Improving the search performance of shade using linear population size reduction,” in *Evolutionary Computation (CEC), 2014 IEEE Congress on*. IEEE, 2014, pp. 1658–1665.

[26] A. Draa, S. Bouzoubia, and I. Boukhalfa, “A sinusoidal differential evolution algorithm for numerical optimisation,” *Applied Soft Computing*, vol. 27, pp. 99–126, 2015.

[27] R. A. Sarker, S. M. Elsayed, and T. Ray, “Differential evolution with dynamic parameters selection for optimization problems,” *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 5, pp. 689–707, 2014.

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

[29] Y. Wang, Z. Cai, and Q. Zhang, “Differential evolution with composite trial vector generation strategies and control parameters,” *Evolutionary Computation, IEEE Transactions on*, vol. 15, no. 1, pp. 55–66, 2011.

[30] S. M. Elsayed, R. A. Sarker, and D. L. Essam, “A self-adaptive combined strategies algorithm for constrained optimization using differential evolution,” *Applied Mathematics and Computation*, vol. 241, pp. 267–282, 2014.

[31] F. Neri and V. Tirronen, “Recent advances in differential evolution: a survey and experimental analysis,” *Artificial Intelligence Review*, vol. 33, no. 1-2, pp. 61–106, 2010.

[32] S. Das, S. S. Mullick, and P. Suganthan, “Recent advances in differential evolution–an updated survey,” *Swarm and Evolutionary Computation*, vol. 27, pp. 1–30, 2016.

[33] B. Bischl, O. Mersmann, H. Trautmann, and M. Preuß, “Algorithm selection based on exploratory landscape analysis and cost-sensitive learning,” in *Proceedings of the 14th annual conference on Genetic and evolutionary computation*. ACM, 2012, pp. 313–320.

[34] N. Hansen, S. Finck, R. Ros, and A. Auger, “Real-parameter black-box optimization benchmarking 2009: Noiseless functions definitions,” 2009.

[35] N. Hansen, A. Auger, S. Finck, and R. Ros, “Real-parameter black-box optimization benchmarking 2010: Experimental setup,” 2010.

[36] O. Mersmann, B. Bischl, H. Trautmann, M. Preuss, C. Weihs, and G. Rudolph, “Exploratory landscape analysis,” in *Proceedings of the 13th annual conference on Genetic and evolutionary computation*. ACM, 2011, pp. 829–836.

[37] K. M. Malan and A. P. Engelbrecht, “Particle swarm optimisation failure prediction based on fitness landscape characteristics,” in *Swarm Intelligence (SIS), 2014 IEEE Symposium on*. IEEE, 2014, pp. 1–9.

[38] P. A. Consoli, L. L. Minku, and X. Yao, “Dynamic selection of evolutionary algorithm operators based on online learning and fitness landscape metrics,” in *Simulated Evolution and Learning*. Springer, 2014, pp. 359–370.

[39] S. Elsayed and R. Sarker, “Differential evolution framework for big data optimization,” *Memetic Computing*, vol. 8, no. 1, pp. 17–33, 2016.

[40] Y. Borenstein and R. Poli, “Information landscapes,” in *Proceedings of the 7th annual conference on Genetic and evolutionary computation*. ACM, 2005, pp. 1515–1522.

[41] Y. Borenstein and R. Poli, “Decomposition of fitness functions in random heuristic search,” in *Foundations of Genetic Algorithms*. Springer, 2007, pp. 123–137.

[42] T. Jones, S. Forrest *et al.*, “Fitness distance correlation as a measure of problem difficulty for genetic algorithms.” in *ICGA*, vol. 95, 1995, pp. 184–192.

[43] M. Tomassini, L. Vanneschi, P. Collard, and M. Clergue, “A study of fitness distance correlation as a difficulty measure in genetic programming,” *Evolutionary Computation*, vol. 13, no. 2, pp. 213–239, 2005.

[44] M. Weber, V. Tirronen, and F. Neri, “Scale factor inheritance mechanism in distributed differential evolution,” *Soft Computing*, vol. 14, no. 11, pp. 1187–1207, 2010.

[45] G. Karafotias, M. Hoogendoorn, and A. E. Eiben, “Parameter control in evolutionary algorithms: Trends and challenges.” *IEEE Trans. Evolutionary Computation*, vol. 19, no. 2, pp. 167–187, 2015.

[46] S. M. Elsayed, R. A. Sarker, and D. L. Essam, “Multi-operator based evolutionary algorithms for solving constrained optimization problems,” *Computers & operations research*, vol. 38, no. 12, pp. 1877–1896, 2011.

[47] Y. Wang, Z. Cai, and Q. Zhang, “Differential evolution with composite trial vector generation strategies and control parameters,” *Evolutionary Computation, IEEE Transactions on*, vol. 15, no. 1, pp. 55–66, 2011.

[48] S. M. Elsayed, R. A. Sarker, and D. L. Essam, “Ga with a new multi-parent crossover for solving ieee-cec2011 competition problems,” in *2011 IEEE Congress of Evolutionary Computation (CEC)*. IEEE, 2011, pp. 1034–1040.

[49] B. Rosner, R. J. Glynn, and M.-L. T. Lee, “The wilcoxon signed rank test for paired comparisons of clustered data,” *Biometrics*, vol. 62, no. 1, pp. 185–192, 2006.