DESIGN AND APPLICATIONS OF ANTLION OPTIMIZER

Ph. D. THESIS

by

SHAIL KUMAR DINKAR

DEPARTMENT OF MATHEMATICS
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE - 247 667 (INDIA)
FEBRURY, 2019

DESIGN AND APPLICATIONS OF ANTLION OPTIMIZER

A THESIS

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

of

DOCTOR OF PHILOSOPHY

in

MATHEMATICS

by

SHAIL KUMAR DINKAR

DEPARTMENT OF MATHEMATICS
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE - 247 667 (INDIA)
FEBRUARY, 2019

INDIAN INSTITUTE OF TECHNOLOGY ROORKEE

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the thesis entitled "**DESIGN AND APPLICATIONS OF ANTLION OPTIMIZER**" in partial fulfilment of the requirements for the award of the Degree of Doctor of Philosophy and submitted in the Department of Mathematics of the Indian Institute of Technology Roorkee is an authentic record of my own work carried out during a period from December, 2015 to February, 2019 under the supervision of Dr. Kusum Deep, Professor, Department of Mathematics, Indian Institute of Technology Roorkee.

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

Signature of the Candidate

This is to certify that the above statement made by student is correct to the best of my (our) knowledge.

Signature of Supervisor(s)

The Ph.D Viva-Voce Examination of **Mr. Shail Kumar Dinkar**,Research Scholar, has been held on **21-06-2019.**

Chairperson,SRC **Signature of External Examiner**

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

Signature of Supervisor(s) **Head of the Department**

Dated:________________

Abstract

Antlion Optimizer (ALO) is a metaheuristic for global optimization problems based on life cycle and unique hunting behaviour of antlion. Though this algorithm is developed recently, yet its efficiency and effectiveness can be demonstrated to solve various real life complex applications in many areas such as feature selection problems, optimal power and load dispatch in electrical engineering, wireless sensors networks, clustering and classification problems, unit commitment problems are few of many.

The objective of this Ph.D Thesis is to improve the efficiency, reliability and stability of Antlion Optimizer (ALO). To achieve this goal, **Chapter 2** proposes a novel variant of ALO namely Opposition Based Laplacian Antlion Optimizer (OB-*L*-ALO) for unconstraint global optimization. This version addresses the drawback of premature convergence and inability to avoid entrapment into local optima of ALO. For this purpose, the random walk of classical ALO is improved by applying Laplace distribution in place of uniform distribution as a first strategy to enhance exploration of search region in early iterations. The second strategy is to apply opposition based learning (OBL) model to the best (elite) candidate solution. The performance of proposed OB-*L*-ALO is verified over a set of 31 benchmark problems of varying difficulties containing 23 state-of-the-art problems (a set of unimodal, multimodal and fixed dimension multimodal functions) and 8 IEEE CEC 2014 composition functions. This set is produced in Appendix I of this thesis. A wide analysis is performed to validate the performance of the proposed OB-*L*-ALO such as convergence behaviour, trajectory analysis of best candidate solution, average distance analysis of search agent before and after applying the updating strategies and elite convergence curve. Statistical significance of OB-*L*-ALO is tested using Wilcoxon ranksum test. The obtained numerical results establish that the proposed OB-*L*-ALO outperforms classical ALO for most of the problems.

Chapter 3 introduces two variants namely OB-ac-ALO and OB-SAC-ALO to accelerate the convergence to opposition based ALO as given in Chapter 2. The OBL model is hybridized with varying acceleration parameter (ac) to propose OB-ac-ALO which is useful to control the abrupt behaviour of the solutions at later stages of the generation and accelerate the convergence speed. The second modification is accomplished by hybridizing the OBL mechanism applied to best (elite) candidate solution with sine acceleration coefficient (SAC) to propose OB-SAC-ALO. The performance of both the proposed variants is investigated over same set of benchmark functions as applied in chapter 2 and reproduced in Appendix I of this thesis. The similar analysis

metrics are performed as utilized in chapter 2. The obtained results and analysis prove that OB-ac-ALO and OB-SAC-ALO perform better than ALO in majority of the problems.

Chapter 4 proposes another extended variant of opposition based ALO with inspiration to improve the random walk by taking the advantage of random jump based on step length of lévy flight distribution and utilizing the hybridization of OBL mechanism with acceleration parameter to propose opposition based lévy flight antlion optimizer (OB-LF-ALO). The performance of proposed OB-LF-ALO is tested on same set of benchmark problems as utilized in chapter 2 and 3 and compared with classical ALO which is reproduced in Appendix I of this thesis. Wilcoxon ranksum test is used to show the statistical significance with computational complexity. To keep the uniformity among all chapters, similar analysis metrics including search history behaviour are performed to investigate the performance of proposed OB-LF-ALO. The analysis of results prove that OB-LF-ALO outperforms the classical ALO.

Chapter 5 is divided into two parts. In first part, the modification utilizes a different distribution namely Cauchy distribution to generate the random number in place of uniform distribution to implement the random walk as a first strategy. As a second strategy, OBL mechanism is applied around the best (elite) candidate solution and hybridized with acceleration parameter to keep proper balance between early exploration and later exploitation. The modification is proposed as opposition based antlion optimizer using Cauchy distribution (OB-C-ALO) .OB-C-ALO is verified using the same set of benchmark problems as used in previous chapters and compared with classical ALO. Similar analysis metrics are used in previous chapters for analysis and Wilcoxon ranksum test is used to show the statistical significance with computational complexity. The second part of the chapter presents the performance comparison in terms of obtained results and analysis among all the five proposed variants of classical ALO i.e. OB-L-ALO, OB-ac-ALO, OB-SAC-ALO, OB-LF-ALO and OB-C-ALO.

In **Chapter 6**, the performance of proposed variants of classical ALO is investigated over a real world complex application of model order reduction of linear time invariant system in the field of control system. The performance of these algorithms are investigated by applying on three single input single output (SISO) systems including two four and one eight order problem of different characteristics.

Chapter 7 attempts to determine optimal values of heat transfer coefficient ($\widetilde{H}$) and pressure drop(ΔP) parameters. This problem consist of two conflicting objective functions: first to maximize heat transfer coefficient and second to minimize pressure drop value. The problem is optimized using two approaches: First, single objective approach for both the objective functions. Secondly, multi-objective approach in which both the objective functions are optimized .This purpose is achieved using two different methods of multi-objective optimization: (i) Using weighted sum approach of multi-objective optimization using classical ALO and its proposed variants (ii) Pareto based multi-objective optimization using multi-objective antlion optimizer (MOALO).

Chapter 8 investigates the performance of designed variants of classical ALO for optimizing the production of biodiesel from renewable energy sources. In this work, the regression equation demonstrating the relationship among three independent variables namely temperature ,methanol to oil ratio and concentration of catalyst is successfully optimized for biodiesel production using Antlion Optimizer (ALO) and its proposed modified versions OB-*L*-ALO,OB-ac-ALO,OB-SAC-ALO,OB-LF-ALO and OB-C-ALO.

Chapter 9 concludes the Thesis by deriving overall observations and concluding remarks. It also outlines the limitations and scope of the proposed variants of ALO algorithms. Later on, some suggestion to future research are provided.

List of Publications

Published in International Journals

1. Dinkar, S. K., & Deep, K. (2017). Opposition based Laplacian Antlion Optimizer. Journal of Computational Science, 23, 71-90. **(SCIE, IF-1.925)**.
2. Dinkar, S. K., & Deep, K. (2018). An efficient opposition based Lévy Flight Antlion optimizer for optimization problems. Journal of Computational Science, 29, 119- 141. **(SCIE, IF-1.925).**
3. Dinkar, S. K., & Deep, K. (2019). Process optimization of biodiesel production using antlion optimizer. Journal of Information and Optimization Sciences, Taylor & Francis. DOI: 10.1080/02522667.2018.1491821. **(ESCI)**.
4. Dinkar, S. K., & Deep, K. (2019). Accelerated Opposition-Based Antlion Optimizer with Application to Order Reduction of Linear Time-Invariant Systems. Arabian Journal for Science and Engineering, 44(3), 2213-2241. **(SCIE, IF-1.092).**
5. Dinkar, S. K., & Deep, K. (2019). Opposition based Antlion Optimizer using Cauchy distribution and its application to data clustering problem. Neural Computing and Applications, Springer. https://doi.org/10.1007/s00521-019-04174-0. **(SCIE, IF-4.213).**

Communicated

1. Dinkar, S. K., & Deep, K. (2019). Single and Multi-objective optimization of Nanofluid flow in flat tube to enhance Heat Transfer using Antlion Optimizer Algorithms. Arabian Journal for Science and Engineering, Springer.
2. Dinkar, S. K., & Deep, K. (2019).Opposition based Antlion Optimizer using Sine Acceleration Coefficient with application to Order Reduction of Linear Time Invariant Systems. Applied Soft Computing, Elsevier.
3. Dinkar, S. K., & Deep, K. (2019).A survey of recent variants and applications of Ant Lion Optimizer. Neural Computing & Applications.

Presented in Conferences and Published in Proceedings of Conference

1. Dinkar, S. K., & Deep, K. (2019). A Novel CPU Scheduling Algorithm Based on Ant Lion Optimizer. Soft computing for Problem Solving. Advances in Intelligent Systems and Computing, vol. 816, pp. 339-353. Springer, Singapore.

Acknowledgements

First and foremost I would like to thank the divine source of inspiration within, the almighty God, who created all the visible and invisible sources of inspiration around.

On the verge of giving a final shape to my dream, it gives me immense pleasure in expressing deep sense of gratitude to my supervisor Prof. Kusum Deep for her invaluable and scrupulous guidance, motivation and enthusiastic interest throughout my research work. Her affectionate treatment, perspicaciousness and magnanimity made feasible to conclude this gigantic task. I am highly indebted for her gentle and benign behaviour, incredible broad vision and constructive criticism for this thesis work. I humbly acknowledge a lifetime‘s gratitude to her.

I express my regards to Prof. N. Sukavanam Head of the Department, Prof. V.K. Katiyar, former Head of the Department and Prof. S.P. Yadav, SRC Chairperson for their support. Special thanks to my SRC members: Dr. Shiv Kumar Gupta and Dr. Deepak Khare (Dept of WRDM) for spending their valuable time during the discussions over the seminars. I am indebted to the Department of Mathematics, IIT Roorkee, to all its faculty and staff for their academic support and encouragement. I am also thankful to the members of Institute Computer Centre for their cooperation and support. A big thanks to my seniors specially Dr. Assif Assad and my colleagues Shubham Gupta, Teekam Singh, Amit Kumar, Vishnu Singh for providing healthy and progressive research environment and much needed human contact at some of the busier points during my Ph.D. I consider myself truly blessed as I have always been in a good company of friends. Their love, inspiration, supports and cooperation is beyond the scope of any acknowledgement, yet I would like to express my heartfelt gratitude to my friends Mr. Prashant Gaidhane, Dr. Sahaj Saxena and Mr. Avadh Kishor.

Now, I would like to thank some remarkable people who had, are having and will continue to have a tremendous influence on my life. I feel a deep sense of gratitude for my grandparents, Late. Mr. Itwari Lal and Mrs Samri, my parents Mr. Rewati Prasad Dinkar and Mrs. Mohar Wati Devi who raised me with their unconditional love, mental support that I needed in every step and sphere of my life. My parents & grand parents have been the biggest source of inspiration in my life and have inculcated in me the values that really matter in life. My sincere and heartfelt gratitude to my siblings Mr. Hemant Kumar, Mr. Indraj Singh and Mrs. Trimila Singh for their love and support. Most importantly, I want to thank my wife, Madhu Dinkar for her unconditional and continuous support during my whole Ph.D as I had to sacrifice lot of my family time but she always supported me and took responsibility to take care of family. I am also thankful to my son Aksh S. Dinkar and daughter Aadya Dinkar to keep calm and to understand my duties during this tenure. No words can express my gratitude and appreciation to her, and my family, for the kind support and tremendous care.

My thank is also due to G. B. Pant Institute of Engineering and Technology,Pauri,Garhwal for providing me study leave to pursue my research work at IIT Roorkee. I owe my special thanks to QIP center of IIT Roorkee and AICTE for sponsoring my research. My sincere apologies if I could not mention someone but I am grateful for their support also. Finally, I would also like to thank all the readers of this work, since any piece of academia is useful if it is read and understood by others so that it can become bridge for further research.

(Shail Kumar Dinkar)
Roorkee, February 6, 2019

Contents

List of Tables

List of Figures

List of Abbreviations

dg	DEGREE
dB	DECIBEL
rad/s	RADIAN PER SECOND
mm	MILIMETER
kWm^{-2}	KILO WATT PER METER SQUARE
%	PERCENT
nm	NANOMETER
m^3h^{-1}	CUBICMETER PER HOUR
wt%	WEIGHT PERCENT

CHAPTER 1

Introduction

This introductory Chapter states the definitions and underlines the objectives and motivation behind this Thesis. It also reviews the available literature. The Chapter closes with a brief summary of the work presented in this Thesis.

1.1 Optimization

Optimization is the methodology of choosing "the best" alternatives(s) among a specified set of available options. This approach of determining "the largest" / "the smallest" possible value of a given mathematical expression can attain in its specified domain of definition, is called optimization. The mathematical expression that has to be optimized could be either linear, nonlinear, integer, geometric or fractional. In some situations, explicit mathematical formulation of the function is not readily defined or may not be available. Many times the mathematical function which needs to be optimized has restrictions in the form of inequality or equality constraints. Therefore, the process of optimization can be considered as a problem of finding those values of the independent variables which do not violate the inequality and equality constraints in such a way to provide an optimal value of the mathematical function to be improved. In other words, "the mathematical techniques for determining the optimal value or value(s) ("the greatest possible value" or "the least possible value") of a mathematical function are called "Optimization Techniques". Determining the solution of most realistic problems may not be possible in the absence of robust optimization techniques. In literature, numerous books and articles are available based on mathematical concepts of optimization [1-15].

Optimization problems arise in various fields of science, engineering, software industry, economics, manufacturing system, physical science, transportation etc. In view of their applicability, it is necessary to design and develop robust and efficient computational algorithms.

1.2 Definition of an Optimization Problem

Mathematically speaking, the most general form of single objective optimization problem is:

$$Minmize\ (or\ maximize)\ f(\vec{X});\ \vec{X} = (x_1, x_2, \ldots, x_D) \qquad (1.1)$$

$$Subject\ to\ \vec{X} \epsilon\ F,\ \ usually\ defined\ by$$

$$F = \begin{cases} \vec{X} \in R^D \ s.t. \quad h_i(\vec{X}) = 0; i = 1,2,\dots,m \qquad (Equality\ constraints) \qquad (1.2) \\ and \quad g_j(\vec{X}) \geq 0; j = m+1, m+2, \dots, p \quad (Inequality\ constraints) \qquad (1.3) \\ and \quad a_i \leq x_i \leq b_i, i = 1,2,3, \dots D \qquad (Box\ constraints) \qquad (1.4) \end{cases}$$

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

Function $f(\vec{X})$ is known as "objective function" which is to be maximized or minimized. Equation (1.2) showing $h_i(\vec{X}) = 0 \ for \ i = 1,2,\dots,m$ are termed as the equality constraints and equation (1.3) depicting $g_j(\vec{X}) \geq 0 \ for \ j = m+1, m+2, \dots, p$ are known as inequality constraints. (Inequality constraint $g_j(\vec{X}) \leq 0$ may be represented as $-g_j(\vec{X}) \geq 0$). The determined values of the variables $x_1, x_2, \dots, x_D$, should be find in such a way that the optimal value of objective function $f(\vec{X})$ should be optimized without violating any of the restrictions imposed in equation (1.2), (1.3) and (1.4). The variables $x_i's$ are known as 'decision variables'. a_i's are the lower bounds and b_i's are the upper bounds of the "decision variables". "A decision vector $\vec{X} = (x_1, x_{2,\dots,}x_D) \in R^D$which satisfies all the constraints is called a feasible solution". If a solution X is feasible and it optimizes the objective function $f(\vec{X})$ then it is termed as a feasible optimal solution.

On the basis of presence of Constraints, there are two types of optimization problems named unconstrained optimization problems and constrained optimization problems. Unconstrained optimization problems involve an objective function given by equation (1.1) or lower or upper bounds on variables given by equation (1.4). Constrained optimization problems involve an objective function given in equation (1.1), the box constraints given by equation (1.4) and linear or/and non-linear, equality constraints given by equation (1.2) or/and inequality constraints given by equation (1.3). Due to presence of equality and inequality constraints, constrained optimization problems are more difficult to solve.

1.3 Optimal Solutions: Local and Global

Let F denotes the feasible region of the solution vector X that satisfies the conditions of all constraints defined for an optimization problem. Consequently, in case of a minimization problem, "if for $\vec{\bar{X}} \in F$ there exists an ε-neighborhood $N_\varepsilon\left(\vec{\bar{X}}\right)$ around $\vec{\bar{X}}$ such that $f(\vec{X}) \geq f(\vec{\bar{X}})$ for each $\vec{X} \in F \cap N_\varepsilon(\vec{\bar{X}})$ then $\vec{\bar{X}}$ is known as a Local minimum solution". However, "if, $\vec{\bar{X}} \in F$ and $f(\vec{\bar{X}}) \geq f(\vec{\bar{X}})$ for all $\vec{\bar{X}} \in F$ then $\vec{\bar{X}}$ is known as a Global minimum solution" of the optimization problem at hand. Figure 1.1 shows local and global optimum solutions of a mathematical function.

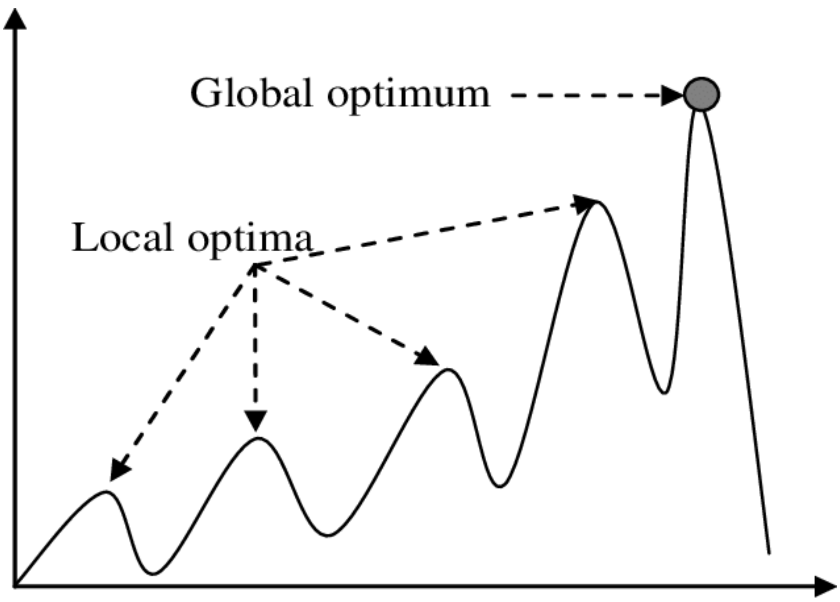


Figure 1.1: Demonstration of Local optima and Global optima

In general it may happen that there are either no optimal solutions, or a unique optimal solution or several optimal solutions, for a given nonlinear optimization problem. In case if a problem is unimodal i.e. it contains only one optimal solution then the same solution will also be the global solution. But in case of multimodal function, it contains many local optimal solution and one or more global optimal solution. If problem in hand is a linear programming problem, every local optima is also the global optima. But it is not true for nonlinear optimization problems. If the feasible region of nonlinear problem is convex for minimization problem, then only the local optima will also be the global optima.

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

For a mathematical function which is twice-differentiable, there exist conditions which may be used to determine a local optimal solution. If it is not so, then due to the property of "continuous differentiability of function", a solution with a lesser objective function value can be determined in its neighborhood. Thus, a sequence of solutions can be constructed which converge to the local optimal solution. However, in general, such tests are not sufficient. Sometimes, it is thought that finite number of steps may not be sufficient to solve a global optimization problem. Therefore any given solution cannot be guaranteed as a solution of global minima without evaluating the objective function at least at one solution of its neighborhood. But, the neighborhoods of a solution may be unbounded; therefore, an infinite numbers of steps are required to attain the global minima.

1.4 Methods for Global Optimization

The method of global optimization aims to determine the best (maximum or minimum) solution out of all the local optima. Designing an algorithm to find global solution is a critical a task as there are no fixed criteria whether the developed technique is able to find a global solution or not. However, many techniques are noticed in literature for solving nonlinear optimization problems. These methods to solve nonlinear optimization may be primarily categorized as deterministic and probabilistic methods.

Fixed guesses are tried in deterministic methods to obtain the neighborhood of global optima. These methods are free from the use of any stochastic approach however try to search the feasible region completely. But these techniques are applied to restricted class of functions. On the other hand, the probabilistic techniques are able to approximate the near to optimal solution. This is because it is assumed that the good solutions are near to global solution and to each other within the search region. This assumption is valid for most real life problems [16]. These methods utilizes the stochastic based approach. Though these methods may not provide an absolute global solution but promise to give good solutions where deterministic approach are failed due to their applicability over a wide range of functions.

1.5 Nature Inspired Optimization Algorithm

One of the most striking trend that emerged in the optimization field is the simulation of natural processes as efficient global search methods. The natural processes or phenomena are firstly analysed mathematically and then coded as computer programs for solving complex nonlinear real world problems. The resulting methods are called 'Nature Inspired Algorithms (NIA)' that can often outperform classical methods. The advantages of these methods are their ability to solve various standard or application based problems successfully without any prior knowledge of the problem space. Moreover, these algorithms are more likely to obtain the global optima of a given problem. They do not require any continuity and differentiability of the objective functions and / or constraints. Also, they work on a randomly generated population of solutions instead of one solution. They are easy to programme and can be easily implemented on a computer.

Nature Inspired optimization techniques can be categorized into various categories. First category is Evolutionary Algorithms which depend upon the evolution process of living organisms. They depend on the Darwin's theory of Evolution which has two features i) Inheritance, and ii) survival of the fittest. The popular techniques under this category are: Genetic Algorithm (GA) [17]. The decision parameters are encoded into a encoded space (Binary / Real / Octal, etc) and crossover, mutation and elitism is performed over a number of generations until a pre specified stopping criteria is attained.

Another popular algorithm under this category called Differential Evolution is proposed in middle of 90's which uses only the mutation operator on a target vector [18]. Then it is extended for constrained optimization using different techniques and applied on applications of various fields [19-22].

Second category is the Swarm Intelligence based Algorithms. These algorithms mimics the foraging behaviour of swarm of various species like ants, birds, wolves, bees etc. In this category of algorithms, the search agents moves in a search space by updating their positions. An important development is the introduction of Particle Swarm Optimization [23]. It mimics the behaviour of a flock of birds or school of fishes. All the solutions or particles of the swarm fly through the search space using their personal best position in history as well as the global best position of the entire swarm. Improved PSO are proposed for global optimization by Ali et al. [24] and Sabat et al.[25-26]. In [27-29], another improvements in PSO are presented and applied to various real world applications.

Glow Worm Swarm Optimization mimics the behaviour of glow worms which emit light in order to attract the others in the group for mating [30]. It is particularly designed to capture multiple local and global optima. Artificial Bee colony optimization is based on Self-organization and division of labor [31]. It is based on inspecting the behaviours of bees on finding nectar and sharing the information of food sources to the bees in the hive, by the employed bees, onlooker bees and scouts. Another Swarm Intelligence based algorithm is the Spider Monkey Algorithm given by Bansal et al. [32]. It is based on the foraging behaviour and fission-fusion social structures of spider monkeys.

Ant Colony Optimization is proposed wherein the pheromone left behind ants and their ability to change their path as and when an obstacle is encountered on their path, is mimicked into the design of Ant colony optimization [33]. The behaviour of the growth of bacteria forms a basis of Bacterial Foraging Optimization Algorithm given by Passino [34]. A modified version of this algorithm is also proposed for engineering design problem [35]. Grey Wolf Optimizer is a relatively new nature inspired optimization technique. This algorithm imitates the exclusive hunting behaviour and leadership hierarchy of grey wolves [36]. Four types of grey wolves such as alpha, beta, delta and omega are employed for simulating the leadership hierarchy by incorporating the three steps of hunting namely searching for prey, encircling the prey and attacking the prey.

The third category draw their inspiration from the physical laws of nature. For example Gravitational Search Algorithm is based on gravitational interaction between masses [37]. It artificially simulates the Newton's Theory, Newtonian laws of gravitation and motion. Similarly, Central Force Optimization [38], is based on gravitational kinematics.

Harmony Search Algorithm is another nature inspired optimization which is inspired from music and applied to engineering applications [39-41].

These days many new nature inspired optimization techniques are being proposed by researchers. Some of them are: Biogeography based Optimization[42-43],Water drop Algorithm [44], Firework Algorithms [45], Teaching Learning Based Optimization [46], Water Weed Optimization [47], Kidney Inspired Optimization [48], Moth-flame Optimization Algorithm [49], Whale Optimization Algorithm [50] and Sine Cosine Algorithm [51].

An excellent review of Nature Inspired Optimization Techniques is presented in: Ali and Törn [52],Yang [53], Kar [54] and Yang et al.[55].

Many of these algorithms suffer with stagnation in local optima and premature convergence. It leads researchers to improve existing algorithm or to develop new algorithm. In the same line, a new swarm intelligence based efficient algorithm namely Antlion Optimizer (ALO) [56] is developed for global optimization problems. This algorithm consists good combination of exploration and exploitation mechanisms.

The scope of this Thesis is Antlion Optimizer (ALO) algorithm.

1.6 The No Free Lunch Theorem

A major and interesting result in optimization theory was the presentation of the "No Free Lunch (NFL) theorem" given by Wolpert and Macready [57-58]. NFL states "that the performance of all optimization (search) algorithms, amortized over the set of all possible functions, is equivalent." The theorem has far reaching implications, because it implies that "no algorithm can be designed so that it will be superior to a linear enumeration of the search space, or even a purely random search". Although, finite search space if used for theorem, yet, it is not proved if the result is applicable to infinite search spaces, e.g. R^D. The implementation of search algorithms in computers is done using finite search spaces; therefore it is applied to all the available algorithms. This theorem says that the performance of search algorithms over all functions is average which implies that “it does not necessarily hold for all subsets of this set. The set of all functions over a finite domain includes the set of all the permutations of this domain”.

1.7 Antlion Optimizer: Background

Antlion optimizer (ALO) is a newly developed non-deterministic, population based metaheuristic technique which relies on the life cycle of antlion. The life cycle of antlion is divided in two main phases: larvae and adult. The antlion hunts in larvae stage and reproduction is done in adult phase. The larvae performs hunting in a sole technique by tunneling a cone shaped pit in the land by throwing the sand outside using its strong jaw as shown in Figure 1.2(a). The antlion’s larvae sits at the bottom and waits for a prey or ant to fall and consumes it(Figure 1.2(b)). Once the antlion larvae catches the ant then it rebuilds the trap again by digging another pit and waits for new ant. The ants are assumed to walk randomly around the antlion’s trap. Another trap is rebuilt again to catch some other ant. The random walk of ant around the antlion mimics the exploration of the search space. Once the ant is caught by the antlion then the fitness of ant becomes better than the corresponding antlion and the position of the ant becomes new position of antlion.

(a)Cone-shaped traps (b)Hunting behaviour of antlions

Figure 1.2: Antlion behaviour

Taken from: Mirjalili, S. (2015). The ant lion optimizer. Advances in Engineering Software, 83, 80-98,Elseveier.[Reprinted with License No. 4578561079159,29-04-2019]

This conceptual description draws the following assumption and facts:

(1) The ants (prey) are assumed to walk randomly around the antlions.

(2) The antlion builds the trap (cone shaped pit) to hunt the prey or ant.

(3) The depth of the pit is proportional to the fitness of antlion i.e. higher fitness means more depth of pit with higher probability to catch the ant.

(4) The boundaries of ant's random walk around antlion is decreased adaptively to mimic the movement of ant towards antlion.

(5) Once the ant is caught and consumed by the antlion then it implies that the fitness of ant becomes better than the antlion.

(6) The position of antlion is updated with the most recent ant (prey) caught by the corresponding antlion and rebuilds the trap to catch another ant.

1.7.1 Mathematical Model

Let the size of population (number of ants) be N within a search space of dimension D. The terminology used in mathematical modelling can be given as follows [59]:

- $S_{ant} = (S_{A,1}, S_{A,2}, \dots S_{A,n}, \dots S_{A,N})^{T}$: Population of ants
- $S_{A,n} = (S_{A,n}^{1}, \dots S_{A,n}^{d}, \dots S_{A,n}^{D})$: nth ant's position and $S_{A,n}^{d}$ is the position of dth variable of the nth ant
- $T_{ant} = (T_{A,1}, T_{A,2} \dots T_{A,n}, \dots T_{A,N})^{T}$: Matrix of Objective(fitness) function values of ants and $T_{A,n}=f(S_{A,n}^{1}, \dots, S_{A,n}^{d}, \dots, S_{A,n}^{D})$ is the fitness of nth ant.
- $S_{antlion} = (S_{AL,1}, S_{AL,2}, \dots, S_{AL,n}, \dots, S_{AL,N})^{T}$: Population of antlion
- $S_{AL,n} = (S_{AL,n}^{1}, \dots S_{AL,n}^{d}, \dots S_{AL,n}^{D})$: nth antlion position and $S_{AL,n}^{d}$ is the position of dth variable of the nth antlion
- $T_{antlion} = (T_{AL,1}, T_{AL,2} \dots, T_{AL,n}, \dots, T_{AL,N})^{T}$: Matrix of Objective(fitness) function values of antlions
- $T_{AL,n}=f(S_{AL,n}^{1}, \dots S_{AL,n}^{d}, \dots S_{AL,n}^{D})$: Fitness of nth antlion

The ant moves randomly around the antlion which is termed as random walk and can be defined as:

$$r_w(S_{A,n}^{d}) = [cumsum(2r(it_1)\text{-}1), cumsum(2r(it_2)\text{-}1) \dots cumsum(2r(it_{max})\text{-}1)] \quad (1.5)$$

The dth variable at itth iteration of nth ant is given as $S_{A,n}^{d}(it)$.

where $cumsum$ states cumulative sum of uniformly generated random numbers and $r(it)$ is a function defined as:

$$r(it)=\begin{cases}1 \; if \; rand > 0.5 \\ 0 \; if \; rand \leq 0.5\end{cases} \qquad (1.6)$$

where $rand$ produces random numbers generated in uniform way between 0 and 1.

Random walk is kept within the search region by applying min-max normalization process as shown in equation(1.7)

$$S_{A,n}^{d}(it) = \frac{(S_{A,n}^{d}(it) - \min r_w(S_{A,n}^{d}))(U^d(it) - L^d(it))}{\max r_w(S_{A,n}^{d}) - \min r_w(S_{A,n}^{d})} + L^d(it) \qquad (1.7)$$

where $\min r_w(S_{A,n}^{d})$ and $\max r_w(S_{A,n}^{d})$ denote min and max of random walk of nth ant for dth variable, $L^d(it)$ and $U^d(it)$ are lower and upper bounds at itth iteration of dth variable.

Two random walks around selected antlion and best antlion are performed. The selection of antlion is done using GA's roulette wheel's method. Then the average of these two random walk is taken to determine the new position of ant as shown in equation (1.8)

$$S_{A,n}^{d}(it)=\frac{r_{w_A}(it)+r_{w_E}(it)}{2} \qquad (1.8)$$

here random walk $r_{w_A}(it)$ is around S_{sel} (selected antlion) and random walk $r_{w_E}(it)$ is around S_{elite}(elite antlion).

After exploration using random walk, the exploitation mechanism is performed by shrinking boundaries adaptively of trap built by antlion. This mechanism is analogous to adaptively decreasing the lower and upper bounds as defined in equations (1.9) and (1.10):

$$L^d(it) = \frac{L^d(it)}{I} \qquad (1.9)$$

$$U^d(it)=\frac{U^d(it)}{I} \qquad (1.10)$$

Where " I is a ratio defined as $I = 10^v \frac{it_{curr}}{it_{max}}$ where it_{curr} is the current iteration and it_{max} is the maximum number of iteration,v is a constant based on the value on current iteration($v = 2$ when $it_{curr} > 0.1\, it_{max}$, $v = 3$ when $it_{curr} > 0.5 it_{max}$, $v = 4$ when $it_{curr} > 0.75\, it_{max}$, $v = 5$ when $it_{curr} > 0.9\, it_{max}$, and $v = 6$ when $it_{curr} > 0.95\, it_{max}$). Here the value of v is used to control the exploitation process".

Bounds of the trap built by antlion are changed around the elite (best) and selected antlion after adaptive shrinking as mentioned in equations (1.9) and (1.10). This phenomena can be modelled as shown in equations (1.11) and (1.12).

$$U_{it}^{d}(it)=\begin{cases} S_{AL}^{d}(it)+U_{it}^{d} \ if \ rand>0.5 \\ S_{AL}^{d}(it)-U_{it}^{d} \quad otherwise \end{cases} \tag{1.11}$$

$$L_{t}^{d}(it)=\begin{cases} S_{AL}^{d}(it)+L_{it}^{d}(it) \ if \ rand>0.5 \\ S_{AL}^{d}(it)-L_{it}^{d}(it) \quad otherwise \end{cases} \tag{1.12}$$

Here the dth variable is S_{AL}^{d} of the antlion S_{sel} or S_{elite}.

Once the objective(fitness) function value becomes better as compared to objective(fitness) function value of antlion then conceptually ant is caught by antlion and the position of antlion is updated with ant's position as shown in equations (1.13):

$$S_{AL,j}(it) = S_{A,i}(it) \ if \ f(S_{A,i}(it)) < f(S_{AL,j}(it)) \tag{1.13}$$

here $S_{AL,j}(it)$ is the position of jth antlion at itth iteration, $S_{A,i}(it)$ is the ith ant's position at itth iteration. "This process is obtained by concatenating all T_{ant} and $T_{antlion}$ and sort them from smallest to largest. Then first N rows are updated as $T_{antlion}$ to the corresponding position of $T_{antlion}$".

The computing steps of classical ALO algorithm can be depicted in Table 1.1 as follows:

Table 1.1: Pseudo code of ALO algorithm [56]

Input: *Population Size N, Maximum iteration* it_{max}*,Lower bound L, Upper bound U and dimension D*

Output: *The best candidate solution* S_{elite}

Randomly initialize the initial population N of ants and antlions
Determine the objective(fitness) function value of antlions
Find out the best(with min fitness) antlion as the elite S_{elite}
Initialize iteration no. $it_{curr}=2$
while ($it_{curr} \le it_{max}$)
for every ant($i = 1, 2, 3, \ldots, N$)
Find an ant lion S_{sel} *using Roulette wheel selection method*
Modify lower L and upper U boundaries with Eqs. $L^{d}(it) = \frac{L^{d}(it)}{I}$ *and* $U^{d}(it)=\frac{U^{d}(it)}{I}$)
for every dimension($j = 1, 2, 3, \ldots, D$)
Perform random walk $rw_{A}(it)$ *around* S_{sel}
and $rw_{E}(it)$ *around* S_{elite} *using uniform distribution with Eq..* $r_{w}(S_{A,n}^{d})$ = [$cumsum(2r(it_{1})$-1), $cumsum(2r(it_{2})$-1)… $cumsum(2r(it_{max})$-1)]
Normalize random walk using Eqs $r_{w}(S_{A,n}^{d})$ = [$cumsum(2r(it_{1})$-1), $cumsum(2r(it_{2})$-1)… $cumsum(2r(it_{max})$-1)] *and*
$S_{A,n}^{d}(it) = \frac{(S_{A,n}^{d}(it)-\min r_{w}(S_{A,n}^{d}))(U^{d}(it)-L^{d}(it))}{\max r_{w}(S_{A,n}^{d})-\min r_{w}(S_{A,n}^{d})}+L^{d}(it)$
end for
Modify the position of ant using Eq. $S_{A,n}^{d}(it)=\frac{r_{w_A}(it)+r_{w_E}(it)}{2}$
end for
for every ant($i = 1, 2, 3, \ldots, N$)
Determine the fitness of all ants
end for
Substitute an antlion with its respective ant if it becomes fitter using Eq. $S_{AL,j}(it) = S_{A,i}(it) \ if \ f(S_{A,i}(it)) < f(S_{AL,j}(it))$
Modify S_{elite} *if an ant lion becomes fitter than the elite*
Increment iteration i.e. $it_{curr}=it_{curr}+1$
end while
Return elite S_{elite}

1.8 Literature Review: Antlion Optimizer (ALO)

ALO has been recognized as an efficient metaheuristic in recent time due to its effective explorative and exploitative behaviour for solving complex nonlinear optimization problems as compared to other available algorithms. ALO is attracting the researchers and developers from optimization field enormously. Numerous advancements have been noted as far as the enhancement in the efficiency and employability of ALO in various field of studies are concerned. Some of them are as follows:

1.8.1 Variants of Antlion Optimizer

The existing algorithms are sometimes not capable of finding global optima due to certain limitations equipped in algorithms. It motivates the researchers to develop modified versions to overcome these limitations and to solve the complicated real life applications. The modifications may be proposed to apprise the basic functional mechanism of the original algorithms or to update the operators to avoid local optima entrapment in early generations by proper balancing the exploration and exploitation mechanisms. Few modification are proposed by incorporating the operators for achieving adequacy of diversification at early generations and intensification at later generations. In this section, various modifications in ALO proposed so far in literature are discussed in different subsections.

Improved Operators

Saha et al. [60] proposed a modified variant of ALO for better balancing between exploration and exploitation. In this work, the initial random population is generated by implementing quasi-opposition based learning (QOBL) mechanism. The searching mechanism in proposed work utilizes QOBL based generation jumping to enhance convergence speed. Chaotic local search (CLS) mechanism is incorporated with QOBL to perform local searching for better exploitation at later stages of generations. The superiority of proposed algorithm is verified by using state of the art benchmark functions available in literature. It is also employed to solve real world applications from the field of electrical engineering.

Employing New Operators

This subsection elaborates the performance enhancement of classical ALO algorithm by employing new local search or cross over operators. Few works have been noticed in this direction using ALO:

In [61], chaotic antlion optimizer (CALO) is proposed for feature selection problem. In classical ALO, the parameter I exhibits trade-off between exploration and exploitation. This parameter decreases linearly and shows exploration at early stages of generations and exploitation at later stages. The authors adapts parameter $\frac{1}{I}$ for obtaining exploration and exploitation and use chaotic system due to its ergodicity and intrinsic stochasticity for adapting this parameter [61]. Various chaotic maps such as logistic map, sinusoidal maps, tent maps etc. are utilized in this work..

A binary version of ALO is also proposed by Emary et al. [62] for feature selection problems. In this work authors proposed two approaches: In first approach, crossover operator is applied on two binary solutions acquired from random walks around elite antlion and around selected antlion using roulette wheel selection method as described in classical ALO. It is determined as:

$$X_i^{t+1}(rw_1, rw_2) \tag{1.16}$$

Here, the crossover $(x,\ y)$ depicts solutions x and y whereas rw_1 and rw_2 are binary vectors representing two random walks around best solution and selected solution respectively.

In second approach of binary ALO, the random walks rw_1 and rw_2 which are determined as continuous functions, are converted into binary solutions by squashing each continuous value in every dimension using S-shaped or V-shaped transfer functions [63]. These transfer functions are also utilized in [64] to improve the performance of binary ALO for feature selection problems. The performance of the proposed binary ALO is verified over data taken from UCI machine learning repository.

Multi-objective version of ALO (MOALO) is proposed by Mirjalili et al. [65] to solve multi-objective problems. Conceptually, multi-objective version should follow the same mechanism of parent optimization algorithm. Here, MOALO utilizes similar mechanism to obtain the updated positions as depicted in parent ALO. All the operators used in ALO are utilized in MOALO.

1.8.2 Applications of ALO

The appropriate balancing between exploration and exploitation operators involved in ALO has ample benefits over other optimization algorithms and establish ALO as an efficient and impressive algorithm to solve real life complex optimization algorithms. The applications tackled by ALO may be categorized as engineering applications, computational intelligence based applications, wireless sensor networking applications etc.

1.8.2.1 Engineering Applications

Every field of real world is related with the design and applications of wide domains of engineering area. The impact of these applications has direct influence on the life of a normal human beings. The outcomes of these applications may require to be improved in terms of efficiency and robustness which initiates the need of the use of good optimization algorithm. The recently proposed ALO is also one such algorithm which is extensively employed to optimize the various engineering applications. In this section, various engineering applications are discussed such as:

Tuning and Designing of Controller

This subsection focused on related literature to optimize the proportional integral derivative (PID) controller. In [66], the performance of PID controller of automobile cruise control system (ACCS) is optimized. The cruise control system is implemented using linearized model in closed loop system and the problem is redesigned as an optimization problem which is solved using ALO.

The ACCS system is utilized to control the speed of vehicle as per the reference command issued by driver. Then the PID controller in control system of ACCS is optimized in terms of time response measured from various performance indexes.

The problem of Load frequency control (LFC) by estimating and tuning the PI controller parameters is resolved using antlion optimizer (ALO) and determines very exciting results [67]. The test system of three area interconnected power system is considered for simulation. Various parameter conditions of loading are chosen to investigate the effectiveness of ALO. The obtained results are then compared with other popular metaheuristic algorithms such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Bat Algorithm (BA) and using standard PI controller. ALO outperforms other techniques as only one parameter is required for fine tuning. The obtained results using ALO exhibits the effectiveness of the controller as far as various indices and settling time are concern.

Control and Power System

The huge amount of power transmission may cause unimportant signal oscillations due to long distance power lines [68]. Now a days the automatic generation control of interconnected systems is in high demand as per the usage requirement of electricity throughout. In [69], the thermal generators are successfully employed for generation control automatically of inter connected power systems of two areas. ALO is utilized to find the parameters of loop governing the thermal generators. To measure the performance of the system, classical fitness function namely Integral Square Error (ISE) and Integral Time Absolute Error (ITAE) are implemented successfully. The authors determined speed regulation and frequency sensitivity coefficient along with integral gain for both the areas possessing interconnected power system.

In [70], the most general problem of power system for optimal power flow is dealt using ALO. A standard 30-bus IEEE system is used to solve this problem for different cases such as active power loss, fuel cost, reactive power loss, and voltage stability index and voltage deviation. The results determined using ALO are also compared with PSO and FA to investigate and analyze the comparative study. This study shows that the obtained results are better than the other algorithms.

ALO is also utilized to deal with the optimal reactive power dispatch (ORPD) problem for a large scale complex power system [71]. The formulation of this problem is done as a complex combinatorial optimization problem having nonlinear characteristics. Three varying test system of different complexity i.e. IEEE 30-bus, larger IEEE 118-bus and IEEE 300 bus system are utilized for simulation to exhibit the performance of ALO in terms of achieving global solution. The superiority of proposed technique using ALO is shown using various simulation and compared with other techniques also. ALO is also utilized in [72] to optimally schedule the various thermal units for electrical economic power dispatch problem (EEPDP) and verified over six unit test systems. The results show minimized operating cost, high speed of convergence and optimum generation of power.

Another version of ALO is developed to solve optimal power dispatch problem in [73]. In this work, elitism property is used with weights and known as weighted elitism ALO. The exploration property is enhanced by redesigning and controlling the elitism phase of original ALO. A parameter "w" termed as elitism weight is used to control the probability of random walk of ant around elite antlion. Using this approach, the modified equation for updating the position of ant as shown in equation (1.8) can be represented as:

$$S_{A,n}^{d}(it)=\frac{(2-w)\times r_{w_A}(it)+w\times r_{w_E}(it)}{2} \tag{1.17}$$

Where $0 \leq w \leq 2$

When $w = 0$, it signifies random walk around selected antlion using roulette wheel method.

When $w = 2$, it shows the random walk around elite antlion.

When $w = 1$ then it behaves like original ALO.

This modified ALO using weighted elitism technique is then applied to optimal power dispatch problem while tested on standard IEEE-30 and IEEE-57 power bus systems and compared with the original ALO.

Fuzzy based ALO is utilized in [74] to solve the dynamic P-Q dispatch problem of power system. The proposed algorithm is tested and verified using multi-objective environment including fuel cost and emission for IEEE-50 and JEAS-118 bus systems. The various important aspects of optimal power flow like VP effects, POZ, Spinning reserves, RR and other limitation s of fuels are also considered in this paper.

Renewable and Grid Distributed Generations

Now a days, renewable sources of energy are in high demand. In case of Distributed Generations, Photovoltaic system (PVS) and Wind Turbine (WT) are identified as great source of distributed generation (DG) [75-76]. These sources produce smaller amount of carbon dioxide as compared to other natural gas and coal sources [77]. The sizing and allocation of distributed generation is performed using antlion optimizer (ALO) in various distribution networks. The loss sensitivity factor (LSF) index in radial distribution network is also performed. Multi-objective functions is used to minimize loss of power, enhancing voltage profiles and voltage stability index(VSI) distribution system. ALO proves its superiority over other techniques while applying on wide variety of systems.

The growing exploitation of fossil fuels indicating the need of renewable energy sources. It can be seen as an alternate energy resource in unit commitment operation [78]. Effectiveness of this operation can be maximized using smart grid system. This system is capable of generating intelligence based decision to maintain environment stability while generating electricity. ALO is found to be good technique to solve unit commitment problem in smart grid as well as in conventional UC in terms of minimum time and operating cost. It is also observed by analyzing the results that using solar energy sources with unit commitment scheduling algorithm are able to optimize the operational cost proficiently.

1.8.2.2 Machine Learning and Computational Intelligence based Applications

The use of machine learning techniques is highly in demand due to its extensive capability of solving very complex and large problems very efficiently. These applications can be categorized in various categories such as clustering applications, features selection problems, training neural networks and fuzzy systems.

Clustering Applications

The use of clustering techniques in wireless sensor networks for harmonizing energy consumption of sensor nodes and increasing network lifetime are of great interests for researchers and developers these days [79]. The use of sensors networks is enormously increased in every field of engineering where energy consumption and handling of energy related constraints are concern. The routing algorithms need to be optimized properly keeping less energy consumption in mind. Clustering based routing algorithm are catching the attention and becoming popular for optimal path. In [80], antlion optimizer(ALO) is utilized by applying clustering based routing algorithm for determining optimum path. In this work, the selection of cluster head is modelled and termed as objective function to improve the performance of network. The discrete version of ALO is applied to determine the optimal data gathering tour for a sink node having minimum tour length of data collection. After applying the improved clustering approach, the results in terms of life time of network, throughput and reduction in number of nodes as compared to other existing algorithms are determined. Thus ALO produces an optimal path for a mobile sink to collect data form the cluster head having tour distance with minimum data collection.

In [81], ALO is utilized as a clustering technique to perform segmentation in MRI images. It is used to segment the liver image while combining with the statistical image. Morphological operations are used to improve the region of segmented liver called as region of interest (ROI). For this purpose mean shift clustering technique is used to determine the number of ROIs using a statistical image of liver. A collection of 70 MRI images is used to segment the liver while validating the proposed approach. An useful metric called structural similarity index (SSIM) is used to measure the accuracy and efficiency of obtained results.

Feature Selection Problems

A recent variant of ALO namely chaotic antlion optimizer [61] is proposed to solve feature selection problems. The parameter of ALO responsible for adaptive shrinking of search space boundaries is being improved using chaos based approach. Then the improved version of ALO is implemented to solve feature selection problems in wrapper mode. The fitness function to be evaluated reflects the classification performance and also to reduce the number of features.18 different data sets are evaluated to perform the simulation and compared against original ALO, PSO and GA in terms of selected features quality. The binary version of ALO are also proposed to solve feature selection problems [61] as discussed in section 3.

Data Communication and Networking Applications

The revolution of digital communication over internet increases lot of data to be transferred through transmission channels. It incurred a number of security issues while transmitting data from source to destination which motivates researchers to optimize and invent new security techniques [82-83]. Community detection in social network is very interesting topic of researchers involved in analyzing these networks [84]. This problem can be treated as an optimization problem as to separate various interconnected network groups having strong connectivity within same group than other group [85]. ALO is used efficiently by determining community fitness and modularity in the networks automatically. Locus based adjacency encoding scheme is utilized relying on GA [86-87]. Normalized mutual information (NMI) is used to measure the similarity index within the same group of elements [88].

ALO is also applied to electromagnetics and antenna community to improve the communication efficiency of the network. In [89], antlion optimizer (ALO) is applied to antenna current and antenna position for pattern synthesizer in linear antenna. The ALO is effectively implemented to attain an array pattern while minimizing side lobe level (SLL) as well as placement of null in specified locations. Close-in side lobe level is also minimized for optimal pattern synthesis. The obtained results are compared with state of the art metaheuristic algorithms.

In [90], a new variant of ALO called enhanced ALO (e-ALO) is proposed to replace uniform distribution of original ALO with four different distribution namely (a) Beta (b) Pareto (c) burr and (d) Cauchy. The updated position of ant in e-ALO can be represented as following equation:

$$S_{A,n}^{d}(it) = \gamma_A \times r_{w_A} + \gamma_E \times r_{w_E} \quad (1.18)$$

Here γ_A and γ_E weighting factors for selected and elite antlion respectively. These weighting factors can be determined using above four distribution to update the position of ant.

This improved e-ALO algorithm is applied in antenna array synthesis for two configuration namely Linear Antenna Array (LAA) and Circular Antenna Array (CAA) [90]. The purpose of applying e-ALO is to determine inter spacing distance between the elements of antenna and their excitation amplitudes. The parameter optimized is side lobe level with the constraints such as beam widths, null at specified locations etc.

Training Neural Network and Fuzzy

In [91], ALO is proposed as stochastic approach to train multi-layer perceptron (MLP). The ALO determines the weights and biases while training MLP to minimize the error and to achieve high classification rate. This model is verified by using four classification datasets and obtained results are compared with trainers using PSO, ACO and GA. The comparison shows that the proposed model achieves the best MSE values and high classification rate. The proposed model using ALO also exhibits better exploration than the other trainers.

1.8.2.3 Biomedical Applications

The use of evolutionary techniques is increased enormously in the field of biomedical recently. The identification and feature extraction of gene groups having similar properties and pattern are very useful for analyzing biomedical problems [92]. ALO is applied for exchanging kidneys in a given pool. Kidney exchange is modeled as an optimization problem [93]. In this work, bio-inspired ALO is implemented to the kidney exchange space and maximizing the number of feasible cycles and chains in the pool of pairs. ALO is proved to be efficient technique to identify comparable kidney exchange results in comparison to other deterministic like integer programming and stochastic approaches such as the genetic algorithm. The program is implemented in Matlab software.

Nowadays, the hepatitis C virus is influencing the humans at large throughout the world. The chemical effect on the human body can be approximated using the quantitative structure-activity relationship (QSAR) models in various applications. Adaptive neuro-fuzzy inference system (ANFIS) is much famous regression technique used to create QSAR models. The use of descriptors and their selection using ANFIS is of much concerned as it shows slow convergence and high complexity. To address this problem, in [94] antlion optimizer (ALO) is used to select appropriate descriptors prior to construct a nonlinear QSAR model and descriptors using ANFIS.

1.8.3 Open Source Software of ALO

In [95], a toolkit is developed to implement the antlion optimizer (ALO) including all the steps within the algorithm in a LabVIEW^{TM} environment. Prior to this, a toolkit of grey wolf optimizer is also developed in LabVIEW^{TM} environment [96]. However, only differential evolution (DE) toolkit is available in LabVIEW^{TM} as a standard optimizer toolkit. The experiments using the developed toolkit is performed on benchmark problems and also performed on DE toolkit. The obtained results using ALO toolkit are superior.

1.9 Summary of Literature Review

The literature review summary of antlion optimizer is depicted in Table 1.2, Table 1.3 and Table 1.4.

Table 1.2: Various variants of ALO

Updating techniques	Saha et al.[60]
New operators	Zawabaa et al.[61],Emary et al.[62]
Multi-objective ALO	Mirjalili et al. [65]

Table 1.3: Applications of ALO

Area	Subarea	Studies
Engineering	Tuning and Designing of Controller	Pradhan et al.[66], Satheesh et al.[67]
	Control and Power Systems	Gupta et al.[69],Trivedi et al.[70], Mouassa et al.[71],Tung et al.[72], Rajan et al.[73], Radha et al.[74]
	Renewable and Grid distributed generations	Ali et al.[75], Ali et al.[76], Sam'on[78]
Machine learning and computational intelligence	Clustering	Yogarajan et al.[80],Mostafa et al.[81]
	Feature selection	Zawabaa et al.[61],Emary et al.[62]
	Communucation and Networking	Babers[85],Pizzuti[86],Pizzuti[87], L. Danon et al.[88],Saxena et al.[89], Subhashini et al.[90]
	Training neural network and fuzzy	Yamany et al.[91]
Biomedical	Biomedical	Hamouda et al.[93], Elaziz et al.[94]

Table 1.4: Developed Open source software of ALO

Technology	Studies
Matlab	Mirjalili et al. [56]
LabVIEW	Gupta et al.[95]

1.10 Motivation and Objective of the Thesis

The performance of metaheuristic algorithms has proven to be superior than the deterministic approach based optimization techniques where gradient derivative free searching optimization is required. In these case stochastic operators utilized in optimization algorithms play vital roles and decisive in the obtained performance of algorithm. Overall, this performance of metaheuristic algorithms heavily depend on exploration and exploitation capability during the whole evolution process. An efficient metaheuristic algorithm must exhibit enhanced exploration at staring phase of the generation and improved exploitation at the later stage of generation so as to be more conducive while converging.

Most of the available nature inspired optimization techniques suffer with stagnation to local optima and premature convergence due to inappropriate balance between two contradictory operators to implement exploration and exploitation process. Though the performance of this recently developed Antlion Optimizer is comparable with other available state of the art optimization algorithms, yet it also suffers with problems as discussed above. This motivates author to design new variants for achieving adequacy between exploration and exploitation so that the performance of classical Antlion Optimizer (ALO) may be improved in terms of efficiency, reliability and robustness.

The work done in this thesis is interdisciplinary in nature and computationally dominant. The major objectives of the thesis may be identified as:

1. Designing efficient and robust Antlion Optimizer based algorithms.
2. Verifying the designed algorithms over well-known benchmark problems available in literature.
3. Applying classical Antlion Optimizer as well as new designed algorithms over wide variety of real life optimization application from the field of Science and Engineering.

1.11 Organization of Thesis

The chapter wise summary of the Thesis is given below:

Chapter 1 is introduction. It states the relevant concepts and elaborates the introduction of Antlion Optimizer (ALO), its existing literature including related modifications and applications.

Chapter 2 proposes a novel opposition based Laplacian antlion optimizer (OB-L-ALO) as a modified variant of classical ALO to address the drawback of premature convergence and inability to avoid entrapment into local optima. The performance of proposed algorithm is analyzed and compared with classical ALO over a set of 31 benchmark problem having variety of difficulties including 23 state-of-the art problems containing unimodal, multimodal, fixed dimension multimodal functions and 8 IEEE CEC 2014 composition benchmark functions. The obtained numerical results for proposed algorithm demonstrates its superiority over classical ALO.

Chapter 3 proposes two variants namely OB-ac-ALO and OB-SAC-ALO to accelerate the convergence to opposition based ALO as given in Chapter 2. The OBL mechanism is applied to the best candidate solution and then hybridized with the acceleration parameter which is useful to control the abrupt behaviour of the solutions at later stages of the generation and accelerate the convergence speed. The second modification is accomplished by hybridizing the OBL mechanism applied to best (elite) candidate solution with sine acceleration coefficient (SAC) to propose OB-SAC-ALO. The performance of both the proposed hybrid algorithm is analyzed and compared with classical ALO using the same set of benchmark problems as utilized in chapter 2. The similar analysis metrics are performed as utilized in chapter 2 including computational complexity. The obtained results and analysis prove that OB-ac-ALO and OB-SAC-ALO perform better than ALO in majority of the problems.

Chapter 4 strikes to improve the balance between exploration and exploitation by proposing a novel Lévy Flight Distributed Opposition based Antlion optimizer (OB-LF-ALO) with Acceleration Coefficient. The performance of proposed version is compared and analyzed with classical ALO using the same set of benchmark problems as utilized in chapter 2 and chapter 3. The quantitative and qualitative results authorizes superiority of proposed algorithm over classical ALO. The analysis over various metric of wide characteristics and obtained numerical results establish the superiority of proposed algorithms.

Chapter 5 introduces another modified version to improve the performance of classical ALO namely Cauchy Distributed Hybrid Opposition based Antlion Optimizer with Acceleration Coefficient (OB-C-ALO). It improves the balance between early exploration and later exploitation during evolutions process. The same set of benchmark problems as used in Chapter 2, 3 and 4 is used to validate the performance of modified algorithm. The obtained results and wide analysis prove the superiority of proposed algorithm over classical ALO. The second part of this chapter represents the comparison among all the modified variants including classical ALO.

Chapter 6 represents the performance of modified variants of classical ALO over a real world complex application of model order reduction of linear time invariant system from the field of control system. The performance of these algorithms are verified by applying on three single input single output (SISO) systems including two four and one eight order problem. The obtained output model of reduced order using the proposed algorithms are compared and analyzed to devise the performance order.

Chapter 7 demonstrates single and multi-objective optimization to determine values of five independent design variables to find out the optimal values of heat transfer coefficient ($\widetilde{H}$) and pressure drop(ΔP) parameters by utilizing two conflicting objective functions: first to maximize heat transfer coefficient and second to minimize pressure drop value. For single objective function optimization all the modified variants of classical ALO are applied while multi-objective antlion optimizer (MOALO) is utilized to find out pareto optimal front while using both the objective function simultaneously.

Chapter 8 focuses to determine optimal values of three independent variables namely temperature, methanol to oil ratio and concentration of catalyst to optimize biodiesel production using classical ALO and its modified variants in this thesis.

Chapter 9 concludes the thesis. It depicts the overall conclusions of this Thesis and outlines the limitations and scope of the proposed algorithms. It also suggests some future scope and new research direction in this area of research.

CHAPTER 2

Opposition based Laplacian Antlion Optimizer (OB-*L*-ALO) for Unconstrained Continuous Optimization Problems

2.1 Introduction

The adequate balance between operators of exploration and exploitation help in improving the performance of metaheuristic algorithms in solving optimization problems. Absence of their proper combination leads an optimization algorithm towards stagnation to local optima and premature convergence. The classical antlion optimizer (ALO) suffers with these limitations. To overcome these shortcomings in ALO, two strategies are applied in this chapter to propose a modified variant of classical ALO. Initially, random walk is modified using Laplace distribution in place of employing uniform distribution to enhance the diversity of search region. At later part of generation, opposition based learning (OBL) model is applied around the best candidate solution to approximate the opposite solution along with the initially randomly generated solution during the process of evolution. The OBL learning is capable of enhancing exploration and acceleration in convergence. This technique is proposed for continuous optimization problems and named as Opposition based Laplacian Antlion Optimizer (OB-*L*-ALO).

A benchmark test suit containing 31 problems of diversified characteristic including 23 state of art problems and 8 IEEE CEC composition functions are taken to show the performance of newly developed variant for varying problem size. A wide variety of analysis metrics are used to investigate the impact of employed strategies such as: convergence curve, best candidate solution's trajectories, best candidate's (elite) convergence curve, average distance between the search agents before and after employing the new strategies and computational complexity. The statistical significance of proposed technique is also analyzed by performing non-parametric Wilcoxon ranksum test.

The organization of this chapter is as follows:

Section 2.2 defines motivation and literature. Represents related literature. Section 2.3 depicts proposed variant OB-*L*-ALO including related concepts about the proposed techniques and pseudo code. Section 2.4 represents the description of benchmark test functions, experimental

---

Partial contents of this chapter has been published as:

- **Dinkar, S. K., & Deep, K. (2017). Opposition based Laplacian Ant Lion Optimizer. Journal of Computational Science, 23, 71-90. (SCIE,IF-1.906)**

Setup, results and discussion. Section 2.5 exhibits convergence and other analysis using various metrics to authenticate proposed algorithm. Section 2.6 presents concluding remarks.

2.2 Motivation

Classical antlion optimizer (ALO) is recently developed metaheuristic algorithm by Mirjalili [56] to solve unconstraint continuous optimization problems. Though ALO is quite comparative with other optimization algorithms available in literature. However, the random walk of ALO ensures diversification in early phase and adaptive shrinking of boundaries guarantees intensification at later part yet it suffers with entrapment in local optima and premature convergence.

The primary goal of this chapter is to enhance the exploration at the early phase of generation to probe all the possible candidate solutions and then quickening the exploitation process to speed up the convergence. In classical ALO, average of two random walks is utilized to determine the new position. The first random walk is performed around the best solution and the second one is performed around the antlion determined using “roulette wheel selection method”. Random walk is implemented using random number generated using uniform distribution. In developed OB-*L*-ALO technique, uniform distribution is replaced with Laplace distribution to enhance the exploration as established in [97]. Then OBL mechanism introduced by Tizoosh et al. [98] is employed to determine opposite solution at later phase of generation to accelerate the convergence towards determining the optimal solution. This concept is mathematically supported with the fact that the opposite numbers come out to be nearer to global optima as compared to original random numbers [99]. Later on, this concept was also conceptualized in the form of space transformation search by Wang et al. [100].

2.3 Proposed Opposition based Laplacian Antlion Optimizer (OB-*L*-ALO)

The proposed modified version of classical ALO is described in this section. In case of continuous distribution, infinite solutions may be approximated within the given range of search region. Keeping this fact in mind, first strategy is applied to enhance the exploration as much as possible by replacing uniform distribution with Laplace distribution to implement random walk. The second strategy, Opposition based learning (OBL) is utilized at later part of the generation to accelerate the convergence. The detailed description of the applied strategies are as follows:

2.3.1 Laplace Distribution

The standard Laplace probability distribution, denotes as $L(\mu,\alpha)$ can be defined in terms of probability density function as:

$$f(x;\mu,\alpha)=\frac{1}{2\alpha}\exp(-\frac{|x-\mu|}{\alpha}),\ -\infty<x<\infty \tag{2.1}$$

In same manner, "distribution function of Laplace distribution" is defined as

$$F(x)=\begin{cases}\frac{1}{2}\exp\left(\frac{|x-\mu|}{\alpha}\right), & x\le\mu,\\ 1-\frac{1}{2}\exp\left(-\frac{|x-\mu|}{\alpha}\right), & x>\mu,\end{cases} \tag{2.2}$$

Here, μ is termed as location parameter in $\mu\ \in\ (-\infty,\infty)$ and α is a scale parameter such as $\alpha>0$. The density function " f is a symmetric function about location parameter μ, increasing in the range $[0,\mu]$ and decreasing in the interval $[\mu,\infty]$ with mode $x=\alpha$".

After performing experiments, the values chosen for location μ and scale α parameters are 1 and 2 respectively.

Figure 2.1 depicts the comparative random number generation utilizing uniform and Laplace distribution over 200 generations. The figure clearly demonstrates that the more search area is spanned using Laplace distribution as compared to uniform distribution. Thus the use of Laplace distribution guarantees to enhance exploration. The random walk shown in Eq.(1.5) can be modified using Laplace distribution as:

$rw(S_{A,n}^{d})$=[$cumsum$(2$*$ $L(it_1)$-1), $umsum$(2$*$ $L(it_2)$-1),…,$cumsum$(2*$L(it_{max})$-1) (2.3)

Here $L(\mu,\alpha)$ denotes Laplace distribution.

2.3.2 Opposition Based Learning (OBL) Model

Metaheuristic algorithms initiate searching process of the region by generating initial random population without any priori information about objective function or other limitations [101]. After enhancing exploration at earlier stage of generation, the exploitation should be improved at later part to accelerate the convergence. It is being proved that the time required to converge candidate solutions is related to their distance from global optima [101]. Mathematically, it is also proved that the opposite numbers are more likely to be closer to the optimal solution than the initial numbers [102]. This mechanism is successfully applied in [103-107].Thus the OBL approach is useful to accelerate the convergence at later stage. Definition may be described as follows:

Definition 1: Opposition Based Learning Rahnamayan et al.[102]. The concept of opposite numbers transforms the region to a new region. Let s be a randomly generated number/ solution in the search region s.t. $s \in [L, U]$, then the opposite number/solution s^* in a transformed space can be defined as

$$s^* = L + U - s \quad (2.4)$$

This definition of OBL may be extended to higher dimensions as well.

Definition 2: Opposition Based Learning in High Dimension Rahnamayan et.al.[102]**.** Let "$\vec{S}$ =$(s_1, s_2, \ldots, s_D)$ be a point in N dimensional search space where $s_1, s_2, \ldots, s_N \in R$, where R is real Euclidean space and $s_i \in [L_i, U_i] \ \forall \ i \in (1,2, \ldots, N)$ then the new transformed feasible solution $\overrightarrow{S^*} = (s_1^*, s_2^*, \ldots, s_N^*)$" is defined by

$$s_i^* = L_i + U_i - s_i \quad (2.5)$$

Let $f(.)$ denotes the cost function or objective function of a given optimization problem and $f(\vec{S})$ objective (fitness) function value at position $\vec{S}$ where $\vec{S} = (s_1, s_2, \ldots, s_N)$ be a candidate solution in an N-dimensional space. The opposite candidate solution of the existing solution $\vec{S} = (s_1, s_2, \ldots, s_N)$ in transformed search space can be defined as $\overrightarrow{S^*} = (s_1^*, s_2^*, \ldots, s_N^*)$ and its objective (fitness) function value can be determined as $f(\overrightarrow{S^*})$. For a problem to be minimized

$$Min \ \ f$$
$$if \ f(\overrightarrow{S^*}) \leq \ f(\vec{S})$$
$$then$$
$$\vec{S} = \overrightarrow{S^*}$$
$$else$$
$$continue \ with \ \vec{S} \ for \ next \ generation$$

In this way, the opposite point (candidate solutions) $\overrightarrow{S^*}$ of initial point $\vec{S}$ generated randomly are determined simultaneously and fitter one is passed to the next generation. It has concluded that the fitness of the candidate solutions is directly proportional to the distance between these solutions to global optima. Hence, the candidates nearer to the optima will be fitter as compared to the other candidates.

In improved version of ALO, the generation of random population is done initially using uniform distribution. The objective function (fitness) value is determined for each solution. The obtained fitness value are sorted in ascending order to determine the minimum fitness value termed as best (elite) candidate solution. After applying improved random walk using Laplace distribution, the updated positions of the solutions are retrieved. For current generation, from the definition of the opposite number the elite candidate solution must be nearer to the optima as compared to other solutions. Using the fact that there are 50% chance for an opposite number to be nearer than the original number, the opposite candidate solution to the elite is determined. Now, a new population containing the opposite candidate solution to elite is generated. The obtained opposite position around elite candidate solution is inserted in place of the candidate of first dimension of the updated population obtained after applying random walk. The objective function (fitness) value is determined and process of evolution is continued by carrying the elite solution to the next generation until termination criteria is satisfied. Mathematically, this process can be modelled as:

$$S_{A,i}(it) = L_i + U_i - S_{elite}(it) \tag{2.6}$$

here $S_{A,i}$ signifies the “updated(opposite) position” of candidate after applying OBL. S_{elite} is the best position obtained in generation it.

2.3.3 Pseudocode of Proposed OB-L-ALO

Two strategies discussed above are applied to classical ALO. The pseudo code of proposed OB-L-ALO is described in Table 2.1. The flow chart is depicted in Figure 2.2.

2.4 Experimental Setup

2.4.1 Benchmark Test Problems

Benchmark test functions of different characteristics including unimodal, multimodal, fixed dimension multimodal and composition functions are tested to investigate the performance of newly developed OB-L-ALO in terms of diversification and intensification. The functions containing single optima are unimodal functions and useful to investigate the exploitation and convergence speed of proposed algorithm. Similarly, the multimodal functions contain many local optima and one or more global optima, hence these function testify the avoidance ability form local (exploration)of proposed algorithm. Though the fixed dimension multimodal functions also contain many optima but these functions behave differently due to their non-scalable property. The composition functions are combination of scaled, transformed and rotated unimodal and multimodal functions and capable of verifying both exploration and exploitation capability of algorithm.

A set of 31 benchmark problems out of which 23 well-known state-of-the-art functions defined by Yao et al. [108] and 8 composition functions taken from IEEE CEC-2014[109] benchmark test suit are employed in this work. The details of the benchmark functions are shown in Appendix I. The first seven functions $F1$-$F7$ are unimodal functions and next six functions $F8 - F13$ are multimodal functions followed by ten functions $F14$-$F23$ which represent low and fixed dimensional multimodal functions. The last eight functions $F24 - F31$ are composition test functions.

Experimental and Parameter Setup

The balance between the applied operators in evolutionary algorithms can be established by execution of continuous experiments by approximating the "population size, stopping criteria and independent runs" etc. Analysis of experiments suggest that the probability of reaching global optima enhances by increasing iterations with appropriate size of population. In this work, the population size is taken as 30.

The performance of OB-L-ALO is validated by comparing with classical ALO. For scalable problems (unimodal and multimodal functions), the experiments are performed for 10 and 30 dimensions and for composition functions, experiments are performed for 10 dimensions. The dimensions of these problems are chosen as 10 and 30 as taken in original work of classical ALO and to keep them similar to real world problems. The performance of proposed algorithm is measured "in terms of average, standard deviation, maximum and minimum of objective function values". Next, thirty independent runs are utilized to record the results by keeping initial random population size as 30. For 10 dimension, 500 iterations and for 30 dimensions and fixed dimensional multimodal functions, 1000 iterations are fixed as stopping criteria. For composition functions, 1000 iterations are chosen. "All the experiments have been performed on MATLAB R2010a on Intel(R) Core(TM) i5-7200 CPU @ 2.50GHz-2.71 GHz with 8GB RAM".

2.4.2 Experimental Results and Discussion

Results on various benchmark test function of newly developed OB-L-ALO algorithm are shown within tables. Since the two strategies are applied to expand the performance of classical ALO thus it is worthy to compare the newly developed algorithm with the classical ALO to authorize superiority of proposed algorithm over the original one. Table 2.2, Table 2.3, Table 2.4 and Table 2.5 exhibit results on 10 and 30 dimensions(scalable unimodal and multimodal functions) whereas Table 2.6 shows the results on fixed dimensions multimodal functions. Table 2.7 presents the results on composition benchmark test functions.

Performance Evaluation of Exploitation Capability (Function $F1 - F7$)

These functions efficiently verify the exploitation capability of proposed algorithm due to the presence of only single optima. The obtained results for these functions are exhibited in Table 2.2 for 10 dimension and Table 2.3 for 30 dimensions respectively. The results in terms of average of objective function values show significant improvement as compared to classical ALO except function $F6$. These obtained results are statistically analyzed using non parametric Wilcoxon ranksum test and signify the superiority of developed OB-L-ALO as depicted in tables of results. The results indicate that the applied strategies using Laplace distribution at early phase and OBL mechanism at later phase of generations efficiently balance the exploration and exploitation process. The obtained near to optimal results show that the opposite candidate solutions around the best candidate solution approximate fitter (closer) solutions than the original solutions. This mechanism helps to accelerate the convergence and speed up the exploitation.

Performance Evaluation of Exploration capability (Function $F8 - F13$)

The multimodal functions are known to have many local and one or many global optimas. These functions are capable of evaluating the local optima avoidance ability of optimization algorithm. The stagnation to local optima can be avoided if algorithm searches the region efficiently by exploring the unvisited regions of search space.

The results for multimodal functions $F8 - F13$ are depicted in Table 2.4 and Table 2.5 for 10 and 30 dimensions respectively. It is observed from the tables that the performance of developed OB-L-ALO is far better in terms of average objective function values for functions $F9 - F13$ for 10 dimension and for functions $F8 - F13$ for 30 dimensions as compared to classical ALO. The analysis of results clearly demonstrates the enhanced capability of exploring the search region using OB-L-ALO by employing Laplace distributed random walk. Also the results of test functions indicate that OBL mechanism is able to accelerate the convergence at later part of evolution. The average standard deviation also authorizes the superiority and stability of proposed algorithm over classical ALO.

Performance Evaluation of Fixed Dimension Multimodal Functions ($F14 - F23$)

Though these functions are multimodal yet the behaviour of these functions are contrary to the scalable multimodal function. Hence it is always interesting to analyze these fixed dimensional functions. The results are exhibited in Table 2.6 which are taken when the stopping criteria is to achieve a maximum of 1000 iterations. It is evident from the table that the proposed OB-L-ALO performs slightly better in function $F14$ and $F15$ than classical ALO in terms of average objective function values. Proposed algorithm also performs better in terms of average objective function values for functions $F19$,$F20$ and $F21$ as compared to classical ALO. The obtained results for functions $F16$,$F17$, $F18$,$F22$ and $F23$ are similar for proposed OB-L-ALO and classical ALO.

Performance Evaluation on Composition Functions

These problems are taken from IEEE CEC 2014 benchmark test suit to evaluate the performance of proposed OB-L-ALO algorithm over complex composite test problems. As these problems are very complex in nature hence it is quite challenging to avoid stagnation to local optima while solving such problems. The results are depicted in Table 2.7. It is evident from the table that average objective function value of function $F24$ is better than classical ALO. However, there is no improvement in average value of function $F25$ and $F26$, still the standard deviation is steady throughout and reflects the stable distribution. The functions $F29$ and $F31$ show strong improvement and remaining $F27$ and $F28$ show slight improvement as compared to classical ALO.

2.5 Analysis of Results

The following analysis are performed to verify the obtained results:

- 2.5.1 Convergence Behaviour
- 2.5.2 Statistical Analysis- Wilcoxon Ranksum Test
- 2.5.3 Proposed Algorithm Analysis
  - Trajectory analysis
  - Trajectory behaviour of average distance between search agents
  - Elite convergence curve
- 2.5.4 Computational Complexity

2.5.1 Convergence Behaviour

To verify superiority of newly developed OB-L-ALO algorithm, graphical interpretation of convergence behaviour is analyzed and compared with classical ALO. Various test functions of different characteristics are taken to authorize the performance of OB-L-ALO. The number of generations and the best average value of objective functions for 30 independent runs have been plotted on horizontal-axis and log scale of vertical-axis respectively.

In Figure 2.3, three curves showing convergence behaviour over fixed number of generations for unimodal functions $F1$,$F3$ and $F7$ are depicted. The curves clearly indicate that the proposed OB-L-ALO starts converging at early generations than the classical ALO. The convergence curve for functions $F1$,$F3$ depicts significant convergence speed and better convergence behaviour for function $F7$ as compared to classical ALO. These functions in Figure 2.3 exhibit that inclusion of OBL followed by enhanced exploration using Laplace distribution is highly impactful for convergence acceleration and capable of finding global optima at early stage of generations. This conduct of proposed algorithm promises to converge to global optima [110].

In Figure 2.4, curves of three scalable multimodal function ($F11, F12, F13$) are drawn and compared with classical ALO. The proposed algorithm shows sharp convergence starting from the initial generations. The convergence curves of proposed OB-L-ALO clearly exhibit their superiority over classical ALO for all the functions. It shows the local optima avoidance capability of proposed algorithm. These curves also authorizes that the inclusion of Laplace distribution enhances exploration in early generation and OBL mechanism accelerates exploitation at later generation. Figure 2.5 depicts convergence curves for fixed or non-scalable multimodal functions $F15$ and $F21$. Proposed OB-L-ALO exhibits improvement in convergence as compared to classical ALO. Overall, it can be concluded that inclusion of OBL model promises to enhance diversification by exploring the opposite candidate solution simultaneously with the original solutions and able to avoid local optima entrapment.

Figure 2.6 depicts the convergence graph of three composition function ($F27$, $F29$ and $F31$) . These benchmark functions are combinations of compound functions which make them very similar to real life problems. These problems require very efficient and tuned algorithm containing balanced combination of exploitation and exploration parameters. Convergence analysis of Function $F27$ shows slightly better convergence but functions $F29$ and $F31$ exhibits better convergence rate for proposed OB-L-ALO over the classical ALO.

2.5.2 Statistical analysis- Wilcoxon Ranksum test

The success of an algorithm can be measured by analyzing its statistical significance. This analysis tests the data samples and its distribution pairing with the data sample drawn from other algorithm and concludes if an algorithm is superior as compared to other algorithm in terms of statistical significance or not. This section demonstrates statistical analysis of proposed OB-*L*-ALO and compared with classical ALO. To verify statistical significance, Wilcoxon ranksum test is performed to reject null hypothesis.

"The null hypothesis states that the sample data drawn from algorithm1 and algorithm 2 from continuous distribution having equal median against the alternative that actually they are not". The sampled data should not necessarily of same length but it must be independent. The applied Wilcoxon ranksum test utilizes this null hypothesis and verifies the statistical significance of the algorithms.

The data for this test is drawn in pair from two algorithms to be tested. The dimensions of the data drawn is 30 for 23 state-of-the-art problems and 10 for CEC 2014 composition functions which is taken after simulating 30 independent runs. The confidence level is chosen as 0.95 to test the null hypothesis. The obtained p-values of the two pairs of ALO with proposed OB-*L*-ALO proposed in this chapter are shown in Table 2.8. In Table 2.8, it is shown "that '+' means significant statistical difference (rejection of null hypothesis) at 0.05 level of significance, '-' designates no significant difference and '=' shows that the sample drawn from both the algorithms are same and no comparison is possible".

It can be clearly observed from the table that OB-*L*-ALO proves to be statistical significant as compared to classical ALO for 19 problems and behaves statistically similar for 5 problems.

2.5.3 Analysis of Proposed Variant

The modified version OB-*L*-ALO of classical ALO is proposed to enhance exploration and acceleration in convergence to achieve proper balance between exploration and exploitation at early as well as later stages of generations. These improvements can be authorized by employing certain metrics such as (1) Trajectory analysis (2) Average distance between search agents before and after applying improvement strategies and (3) Elite convergence for analyzing the performance of proposed version. For this purpose some benchmark test functions of different characteristics have been used. All the experiments for analysis are performed by employing four search agents for 30 dimensions over 200 iterations and compared with classical ALO with same parameter setting. The analysis using various metric is as follows:

Analysis of trajectory

The trajectory behaviour of best (elite) candidate solution for proposed OB-L-ALO and classical ALO is depicted in Figure 2.7 to verify movement of candidate solutions towards the optima. These trajectories of first dimension of elite candidate are drawn up to 200 iterations. This trajectory behaviour analysis is capable of showing whether the proposed algorithm is able to approximate the nearby positions of global optima or not.

For this analysis, a set of benchmark test functions is chosen including unimodal ($F1$,$F3, F7$), multimodal ($F9, F10$,$F11$), two fixed dimension multimodal ($F14$,$F20$) and two composition ($F28$,$F29$) functions. It can be observed from the figures that the proposed OB-L-ALO shows enhanced exploration due to employment of Laplace distribution at earlier phase of evolution. It is clearly evident from the figure that the positions of elite in OB-L-ALO shows sharp acceleration in convergence for functions $F1$,$F3, F7$, $F9$ and $F11$ and approve the use of Laplace distribution for exploration and OBL mechanism for convergence acceleration. In same manner, exploration for function $F10$ is also enhanced with great extent at early generations and then converges to the nearby positions of global optima. The average fitness value of proposed OB-L-ALO is slightly better than classical ALO for function $F14$ and consequently trajectory of the proposed OB-L-ALO is able to search the closer positions as compared to classical ALO. Whereas, for function $F20$, the proposed OB-L-ALO and classical ALO approximate almost similar nearby positions of optima in spite of better average fitness value achieved by proposed OB-L-ALO. The trajectory of composition function $F28$ shows similar behaviour as of classical ALO but it approximates the closer positions to global optima at the later generations as compared to classical ALO. Similarly, the trajectory of function $F29$ is far than the ALO but approximates the nearer positions at later generations. It can be concluded that the performance of OB-L-ALO outperforms the classical ALO as far as the searching of closer positions to the global optima is concern.

Trajectory behaviour of average distance between search agents

It is good idea to determine the "distance between search agents before and after applying the improvement strategies". The search agents with lesser distance apart disclose closer positions to global optima. This theory has been conceptualized by stating that there are half chances of a candidate solution determined using OBL mechanism to be closer than the randomly generated number [61]. So, it is better to use opposite number rather than generating additional random number during evolution process for convergence [61]. This analysis is useful to evaluate the importance of OBL mechanism. This can be determined experimentally by evaluating the "average of distance (Euclidean) between the search agent before and after applying the updating strategies".

Mathematically, the distance (Euclidean) between two points $P(p_1, p_2, \ldots, p_D)$ and $Q(q_1, q_2, \ldots, q_D)$ in a D-dimensional search space can be defined as [61]:

$$d(P,Q)=||P,Q||=\sqrt{\sum_{i=1}^{D}(p_i - q_i)^2} \tag{2.7}$$

It can be simplified for one dimension as follows:

$$d(P,Q)=\left||P,Q|\right| = |p - q| \tag{2.8}$$

The "absolute value of average distance between search agents before and after employing improvements strategies" is determined in each iteration. The obtained value is than compared with the average distance of classical ALO to establish the superiority of OB-L-ALO. The experiments are conducted for 200 iterations over first variable by choosing ten benchmark test functions including three unimodal$(F1, F3, F7)$,,three multimodal$(F9, F10, F11)$,two fixed dimension multimodal$(F14, F20)$ and two composition($F28$ and $F29$) functions"to validate the performance of OB-L-ALO.

The curves shown in Figure 2.8 depict the determined average distance using proposed OB-L-ALO. From the figure, it is shown that the average distance is fluctuating at early generations due to enhanced exploration caused by Laplace distribution. Then it gets smooth and steady once distance becomes lesser or closer to optimal solution at later generations. This behaviour is caused by using OBL integrated with exploitation adjustment parameter. For functions $F1, F3, F7, F9$, $F10$ and $F11$,the determined average distance shows steady behaviour around the optimal position as OB-L-ALO performs better than classical ALO as shown in result tables. This implies that the obtained distance for proposed OB-L-ALO is closer to optima as compared to classical ALO. The curve for function $F14$ shows that proposed algorithm determine closer distance as compared to classical ALO at later generations. The curve for function $F20$ shows abrupt behaviour in starting generations due to inclusion of Laplace distribution but become steady at later generations. The curves of composition functions $F28$ and $F29$ are also able to approximate the closer distance for OB-L-ALO while comparing with classical ALO. Overall, this analysis establishes that the obtained average distance using proposed OB-L-ALO promises to accelerate the convergence.

Elite convergence curve

The convergence curve verifies the convergence capability of best (elite) candidate solution. The similar set of test functions as utilized in previous section is used for this analysis. These curves are depicted in Figure 2.9.

The curves depicted in Figure 2.9 exhibit that the fitness of elite solution drops significantly starting from initial iterations which shows that the elite solution starts to converge even at initial iterations for proposed OB-L-ALO as compared to classical ALO. The classical ALO shows better convergence behaviour for function $F24$ in initial iterations but as compared to proposed OB-L-ALO. The curve of function $F25$ depicts better convergence steadily for proposed OB-L-ALO than classical ALO and further drops at later generation and exhibits the superiority of proposed OB-L-ALO over classical ALO. This analysis validates high acceleration in convergence with the increase in number of iterations and tries efficiently to search global optima with more precision. The employability of Laplace distribution ensures the enhanced exploration and avoidance of local optima stagnation. The opposition based learning (OB) is capable of approximating the nearer solution to the best ant lion (elite) and ensures the acceleration in convergence towards a point [110].

2.5.4 Computational Complexity

The computation complexity of proposed OB-L-ALO relies on the number of search agents, maximum number of iterations, for loops used throughout the algorithm and the number of function evaluation while determining the cost of objective function. The complexity of proposed OB-L-ALO and classical ALO can be defined as follows:

The parameters are initialized using constant input values for both algorithms with "time complexity $O(1)$".The search agents or population of "size N" for both the algorithms in step 1 are generated randomly with "time complexity $O(N)$". In step 2, the fitness or objective function value is determined with "time complexity $O(N) * O(f(\vec{X}))$" where " $f(\vec{X})$ denotes the objective function value" which is followed by sorting of fitness values as "linear search with time complexity $O(N)$".

The while loop up to maximum number of iteration $\boldsymbol{it_{max}}$ initiated at step 5. So every statement inside this loop will have time complexity multiplied by $\boldsymbol{it_{max}}$**.** The next step 6 executes *for* loop of size N and step 7 and 8 are executed for N times with complexities ($it_{max} * N$) respectively for both ALO and proposed OB-L-ALO. Then the random walk is determined and normalized for “each dimension D with time complexities $O(it_{max} * D * N)$”. In next step 12, the mean of the two random walks is determined to “update the positions of all N ants with time complexity $O\ (it_{max}\ * N)$”.

Before determining the fitness of all ants after position updates, the position of randomly generated population of ants are again updated using proposed opposition based learning(OBL) integrated with Laplace distribution having time complexity $O\ (it_{max}\ * 1)$. The random walks determined in earlier step are used to update the ants positions in step 14 having complexity O ($\boldsymbol{t_{max}}$ ***N**). Opposition based learning is applied around the elite antlion S_{elite} to again determine the updated ants positions in step 16 and 17 with complexity $O\ (\boldsymbol{t_{max}} * \boldsymbol{N})$ .This step defines as improvement strategy in proposed OB-L-ALO which is not present in the original ALO. The fitness of all ants is calculated in step 19 using updated position by executing the instruction for N times. Thus the fitness of updated ants are determined “with time complexity $O(it_{max}\ * N) * O(f(\vec{X}))$”. Hence the time complexity in worst case scenario can be visualized for both “classical ALO as well as proposed OB-L-ALO as $O(it_{max} * D * N)$* O $(f(\vec{X}))$”.

2.6 Conclusion

The goal of this chapter is to enhance the exploration to avoid the stagnation in local optima and then accelerating the convergence speed at later part of the generation. Two strategies namely Laplace distribution by substituting the uniform distribution in random walk and opposition based learning (OBL) about the elite(best) candidate solutions are applied to achieve the purpose.

The improved performance of developed OB-L-ALO is verified by using 31 test problems of wide and different behaviour having complexities matching with real world optimization problems. Extensive analysis is performed to establish the superiority of the proposed algorithm such as convergence behaviour, trajectory behaviour, average distance between search agents before and after applying improvement strategies and elite convergence analysis. A non-parametric statistical Wilcoxon ranksum test is performed to verify statistical significance of proposed OB-L-ALO and computational complexities of both proposed algorithm and classical ALO are also discussed.

The inclusion of Laplace distribution is clearly capable of enhancing the exploration ability of the search region as compared to classical ALO. The analysis reflects the influence of OBL mechanism at later stage of generation in accelerating the convergence speed. In majority of the problems, proposed algorithm exhibits consistent acceleration after sudden fluctuation shown in starting iterations in trajectory curves. The employed techniques reduces the probability of stagnation to local optima and tend to find global optima while enhancing the convergence speed.

Though the proposed OB-*L*-ALO improves the overall performance of classical ALO, yet it suffers with certain limitations. The Laplace distribution is a double exponential probability distribution which makes random walk more exhaustive and increases the computation time specifically for higher dimensional problems.

Table 2.1: Pseudo code of Proposed OB-L-ALO

Input: *Population Size N, Maximum iteration* it_{max}*,lower bound L, Upper bound U and dimension D*

Output: *The best candidate solution* S_{elite}

Randomly initialize the initial population N of ants and ant lions
Determine the objective(fitness) function value of antlions
Find out the best(with min fitness) antlion as the elite S_{elite}
Initialize iteration no. it_{curr}*=2*
    while $(\boldsymbol{it_{curr} \leq it_{max}})$
        for every ant $(\boldsymbol{i = 1, 2, 3, \ldots, N})$
            Find an ant lion S_{sel} *using Roulette wheel*
            Modify lower L and upper U boundaries with equations Eqs. $L^d(it) = \frac{L^d(it)}{I}$ *and* $U^d(it) = \frac{U^d(it)}{I}$*)*
            for every dimension $(\boldsymbol{j = 1, 2, 3, \ldots, D})$
                Perform random walk $rw_A(it)$ *around* S_{sel}
                and $rw_E(it)$ *around* S_{elite} *using Laplace distribution with Eq.* $rw(S^d_{A,n})$=[$cumsum$(2$*$ $L(it_1)$-1), $umsum$(2$*$ $L(it_2)$-1),…,$cumsum$(2*$L(it_{max})$-1)
                Normalize random walk using Eqs. $rw(S^d_{A,n})$=[$cumsum$(2$*$ $L(it_1)$-1), $umsum$(2$*$ $L(it_2)$-1),…,$cumsum$(2*$L(it_{max})$-1) *and*
$$S^d_{A,n}(it) = \frac{(S^d_{A,n}(it) - \min r_W(S^d_{A,n}))(U^d(it) - L^d(it))}{\max r_W(S^d_{A,n}) - \min r_W(S^d_{A,n})} + L^d(it)$$
            end for
            Modify the position of ant using Eq. $S^d_{A,n}(it) = \frac{r_{W_A}(it) + r_{W_E}(it)}{2}$
        end for
        Modify position of ant by applying opposition based learning model
        using Eq. $S_{A,i}(it) = L_i + U_i - S_{elite}(it)$
        for every ant $(\boldsymbol{i = 1, 2, 3, \ldots, N})$
            Determine the fitness of all ants
        end for
        Substitute an antlion with its respective ant if it becomes fitter using Eq.
        $S_{AL,j}(it) = S_{A,i}(it)$ *if* $f(S_{A,i}(it)) < f(S_{AL,j}(it))$
        Modify S_{elite} *if an ant lion becomes fitter than the elite*
        Increment iteration i.e. it_{curr}*=*it_{curr}*+1*
    end while
Return elite

Table 2.2: Average, Standard Deviation, Minimum, and Maximum of objective function values of unimodal test functions (10D)

Function	***Methods***	***Ave.***	***Stan. Dev.***	***Min.***	***Max.***
$F1$	**OB-L-ALO**	**1.090E-11**	**3.860E-11**	**0.000E+00**	2.050E-10
	ALO	7.870E-09	5.490E-09	2.390E-09	**2.580E-08**
$F2$	**OB-L-ALO**	**1.850E-18**	**1.010E-17**	**0.000E+00**	5.550E-17
	ALO	4.860E-01	8.850E-01	1.550E-05	**2.800E+00**
$F3$	**OB-L-ALO**	**1.730E-32**	**6.950E-32**	**0.000E+00**	3.450E-31
	ALO	8.310E-02	1.850E-01	1.360E-04	**8.750E-01**
$F4$	**OB-L-ALO**	**4.090E-06**	**9.620E-06**	**0.000E+00**	3.220E-05
	ALO	3.180E-03	6.160E-03	1.020E-04	**3.290E-02**
$F5$	**OB-L-ALO**	**5.460E-04**	**1.600E-03**	**1.250E-07**	8.630E-03
	ALO	6.810E+01	1.990E+02	1.020E-04	**1.070E+03**
$F6$	OB-L-ALO	3.320E-08	3.000E-08	7.360E-09	**1.160E-07**
	ALO	**8.390E-09**	**5.220E-09**	**2.150E-09**	2.220E-08
$F7$	**OB-L-ALO**	**2.850E-04**	**2.490E-04**	**3.890E-05**	1.330E-03
	ALO	2.210E-02	1.230E-02	1.810E-03	**5.630E-02**

Table 2.3: Average, Standard Deviation, Minimum, and Maximum of objective function values of unimodal test functions (30D)

Function	***Methods***	***Ave.***	***Stan. Dev.***	***Min.***	***Max.***
$F1$	**OB-L-ALO**	**1.010E-09**	**4.630E-09**	**0.000E+00**	2.530E-08
	ALO	9.450E-06	5.540E-06	1.720E-06	**2.110E-05**
$F2$	**OB-L-ALO**	**5.980E-05**	**4.260E-05**	**0.000E+00**	1.090E-04
	ALO	3.650E+01	5.170E+01	2.090E-02	**1.340E+02**
$F3$	**OB-L-ALO**	**1.160E-29**	**5.950E-29**	**0.000E+00**	3.260E-28
	ALO	9.550E+02	5.080E+02	2.470E+02	**2.160E+03**
$F4$	**OB-L-ALO**	**1.090E-05**	**2.290E-05**	**0.000E+00**	7.130E-05
	ALO	1.210E+01	3.650E+00	4.630E+00	**1.960E+01**
$F5$	**OB-L-ALO**	**5.460E-03**	**7.170E-03**	**5.540E-06**	3.220E-02
	ALO	1.040E+02	3.120E+02	1.640E+01	**1.740E+03**
$F6$	OB-L-ALO	3.230E-05	1.840E-05	7.980E-06	**7.620E-05**
	ALO	**9.490E-06**	**6.100E-06**	**1.320E-06**	2.920E-05
$F7$	**OB-L-ALO**	**2.780E-04**	**2.530E-04**	**9.410E-06**	1.240E-03
	ALO	1.100E-01	3.570E-02	5.700E-02	**2.250E-01**

Table 2.4: Average, Standard Deviation, Minimum, and Maximum of objective function values of multimodal test functions (10D)

Function	***Methods***	***Ave.***	***Stan. Dev.***	***Min.***	***Max.***
$F8$	OB-L-ALO	-2.450E+03	4.230E+02	**-3.320E+03**	-1.920E+03
	ALO	**-2.470E+03**	**4.260E+02**	-3.730E+03	**-1.920E+03**
$F9$	**OB-L-ALO**	**5.070E-10**	**6.450E-10**	**0.000E+00**	**2.240E-09**
	ALO	2.480E+01	1.080E+01	6.960E+00	**5.070E+01**
$F10$	**OB-L-ALO**	**1.540E-05**	**8.100E-06**	**8.880E-16**	**2.850E-05**
	ALO	3.240E-01	5.540E-01	2.070E-05	1.650E+00
$F11$	**OB-L-ALO**	**7.210E-10**	**2.110E-09**	**0.000E+00**	**9.600E-09**
	ALO	2.140E-01	8.720E-02	6.640E-02	4.060E-01
$F12$	**OB-L-ALO**	**1.630E-08**	**2.410E-08**	**4.710E-32**	**1.230E-07**
	ALO	1.790E+00	1.670E+00	8.230E-09	5.610E+00
$F13$	**OB-L-ALO**	**6.820E-08**	**9.610E-08**	**1.350E-32**	**3.800E-07**
	ALO	1.800E-03	4.940E-03	3.840E-09	2.100E-02

Table 2.5: Average, Standard Deviation, Minimum, and Maximum of objective function values of multimodal test functions (30D)

Function	***Methods***	***Ave.***	***Stan. Dev.***	***Min.***	***Max.***
$F8$	OB-L-ALO	**-5.98E+03**	**1.71E+03**	**-1.26E+04**	-5.42E+03
	ALO	-5.44E+03	4.19E+01	-5.54E+03	**-5.42E+03**
$F9$	OB-L-ALO	**3.01E-09**	**4.82E-09**	**0.00E+00**	1.57E-08
	ALO	1.97E+00	8.39E-01	5.89E-04	**3.52E+00**
$F10$	OB-L-ALO	**2.10E-05**	**1.46E-05**	**8.88E-16**	4.20E-05
	ALO	1.87E+00	6.90E-01	1.02E-03	**3.09E+00**
$F11$	OB-L-ALO	**6.32E-09**	**1.27E-08**	**0.00E+00**	5.22E-08
	ALO	1.19E-02	1.26E-02	3.47E-04	**3.86E-02**
$F12$	OB-L-ALO	**6.44E-07**	**5.81E-07**	**7.19E-08**	2.10E-06
	ALO	1.00E+01	4.34E+00	5.75E+00	**2.33E+01**
$F13$	OB-L-ALO	**6.98E-06**	**5.33E-06**	**1.35E-32**	2.05E-05
	ALO	1.32E+00	3.15E+00	1.14E-05	**1.05E+01**

Table 2.6: Average, Standard Deviation, Minimum, and Maximum of objective function values of fixed dimension multimodal functions

Function	***Methods***	***Ave.***	***Stan. Dev.***	***Min.***	***Max.***
$F14$	OB-L-ALO	1.690E+00	8.700E-01	**9.980E-01**	2.980E+00
	ALO	**1.790E+00**	**9.520E-01**	9.980E-01	**3.970E+00**
$F15$	**OB-L-ALO**	**7.390E-04**	**1.280E-04**	4.650E-04	1.220E-03
	ALO	3.380E-03	6.780E-03	**3.080E-04**	**2.040E-02**
$F16$	**OB-L-ALO**	**-1.030E+00**	**6.780E-16**	-1.030E+00	-1.030E+00
	ALO	-1.030E+00	6.780E-16	**-1.030E+00**	**-1.030E+00**
$F17$	**OB-L-ALO**	**3.980E-01**	**1.690E-16**	**3.980E-01**	3.980E-01
	ALO	3.980E-01	1.690E-16	3.980E-01	**3.980E-01**
$F18$	**OB-L-ALO**	**3.000E+00**	**0.000E+00**	**3.000E+00**	3.000E+00
	ALO	3.000E+00	0.000E+00	3.000E+00	**3.000E+00**
$F19$	**OB-L-ALO**	**-9.400E+00**	**2.300E+00**	**-1.020E+01**	**-2.630E+00**
	ALO	-6.780E+00	2.680E+00	-1.020E+01	-2.680E+00
$F20$	**OB-L-ALO**	**-9.640E+00**	**2.330E+00**	**-1.040E+01**	**-2.770E+00**
	ALO	-7.390E+00	3.150E+00	-1.040E+01	-2.750E+00
$F21$	**OB-L-ALO**	**-7.490E+00**	**3.670E+00**	**-1.050E+01**	-2.420E+00
	ALO	-6.590E+00	3.400E+00	-1.050E+01	**-2.420E+00**
$F22$	**OB-L-ALO**	**-3.863E+00**	**3.160E-15**	**-3.863E+00**	**-3.863E+00**
	ALO	-3.863E+00	3.160E-15	-3.863E+00	-3.863E+00
$F23$	**OB-L-ALO**	**-3.322E+00**	**1.810E-15**	**-3.322E+00**	**-3.322E+00**
	ALO	-3.322E+00	1.810E-15	-3.322E+00	-3.322E+00

Table 2.7: Average, Standard Deviation, Minimum, and Maximum of objective function values of composition functions

Function	***Methods***	***Ave.***	***Stan. Dev.***	***Min.***	***Max.***
$F24$	**OB-L-ALO**	**2.546E+03**	**6.248E+01**	**2.500E+03**	**2.630E+03**
	ALO	2.630E+03	1.275E-03	2.630E+03	2.630E+03
$F25$	OB-L-ALO	2.540E+03	2.163E+01	2.517E+03	2.600E+03
	ALO	2.538E+03	1.477E+01	2.512E+03	2.577E+03
$F26$	OB-L-ALO	2.690E+03	**1.470E+01**	2.650E+03	2.700E+03
	ALO	**2.680E+03**	2.430E+01	2.630E+03	2.700E+03
$F27$	**OB-L-ALO**	**2.700E+03**	**1.030E-01**	**2.700E+03**	2.700E+03
	ALO	2.700E+03	1.090E-01	2.700E+03	**2.700E+03**
$F28$	**OB-L-ALO**	**2.940E+03**	**1.640E+02**	**2.700E+03**	3.100E+03
	ALO	3.010E+03	1.440E+02	2.700E+03	**3.110E+03**
$F29$	**OB-L-ALO**	**3.220E+03**	**1.230E+02**	**3.000E+03**	3.710E+03
	ALO	3.490E+03	2.920E+02	3.170E+03	**3.930E+03**
$F30$	**OB-L-ALO**	**1.050E+05**	**4.100E+05**	**3.100E+03**	1.730E+06
	ALO	8.270E+05	1.520E+06	3.100E+03	**3.670E+06**
$F31$	**OB-L-ALO**	**3.790E+03**	**2.530E+02**	**3.500E+03**	4.470E+03
	ALO	4.420E+03	4.140E+02	3.880E+03	**6.170E+03**

Table 2.8: P-values of Wilcoxon ranksum test

Function	***P-values***	***Conclusion***
$F1$	7.880E-12	+
$F2$	2.920E-11	+
$F3$	3.160E-12	+
$F4$	1.610E-11	+
$F5$	3.020E-11	+
$F6$	3.350E-08	-
$F7$	3.020E-11	+
$F8$	3.350E-08	-
$F9$	2.380E-11	+
$F10$	2.900E-11	+
$F11$	1.620E-11	+
$F12$	3.020E-11	+
$F13$	2.260E-10	+
$F14$	6.870E-01	-
$F15$	1.170E-02	+
$F16$	NA	=
$F17$	NA	=
$F18$	NA	=
$F19$	2.870E-04	+
$F20$	2.700E-03	+
$F21$	5.170E-05	+
$F22$	NA	=
$F23$	NA	=
$F24$	2.33E-11	+
$F25$	7.74E-01	-
$F26$	4.40E-03	-
$F27$	4.70E-03	+
$F28$	3.01E-02	-
$F29$	5.56E-09	+
$F30$	6.49E-05	+
$F31$	7.68E-14	+
24/31(19 ‘+’ , 5 ‘=’)		

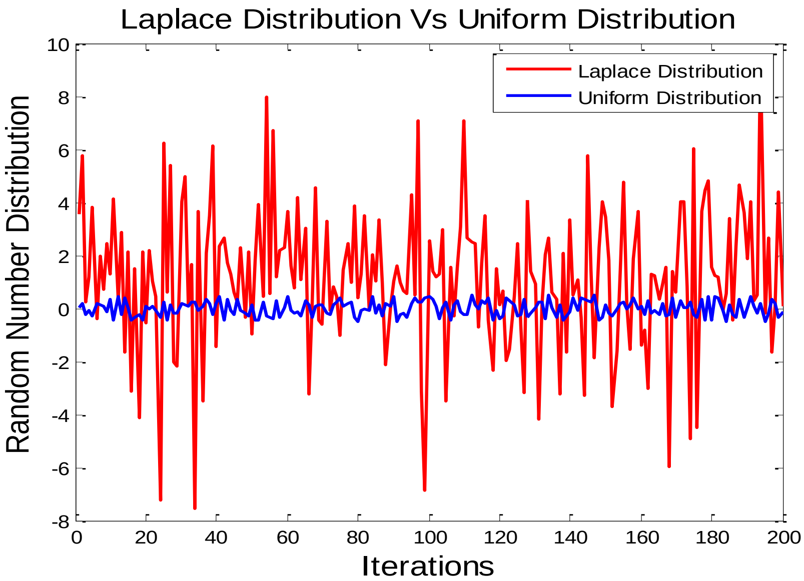


Figure 2.1: Generation of random numbers using Uniform and Laplace distribution

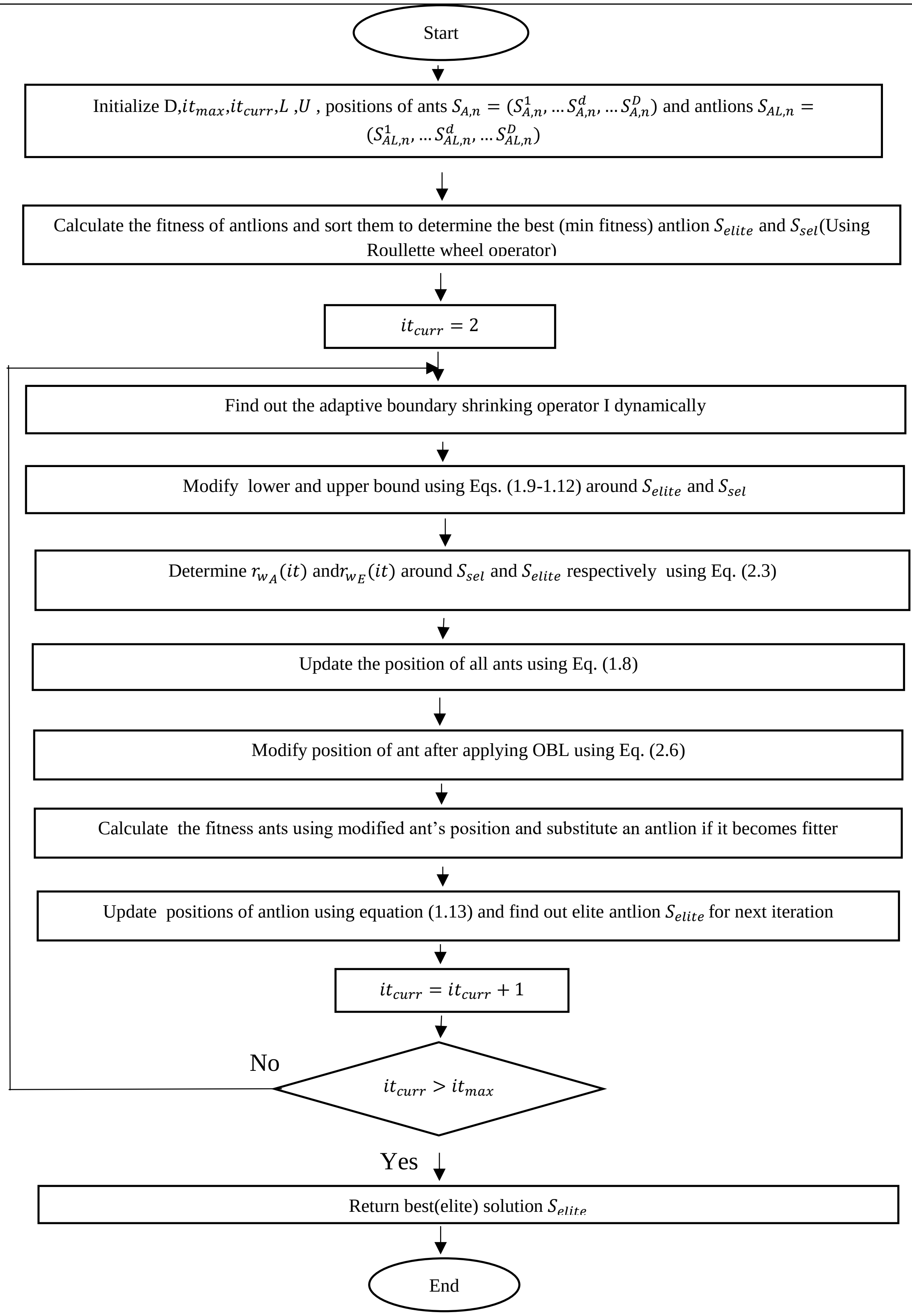


Figure 2.2: Flowchart of Proposed OB-*L*-ALO Algorithm

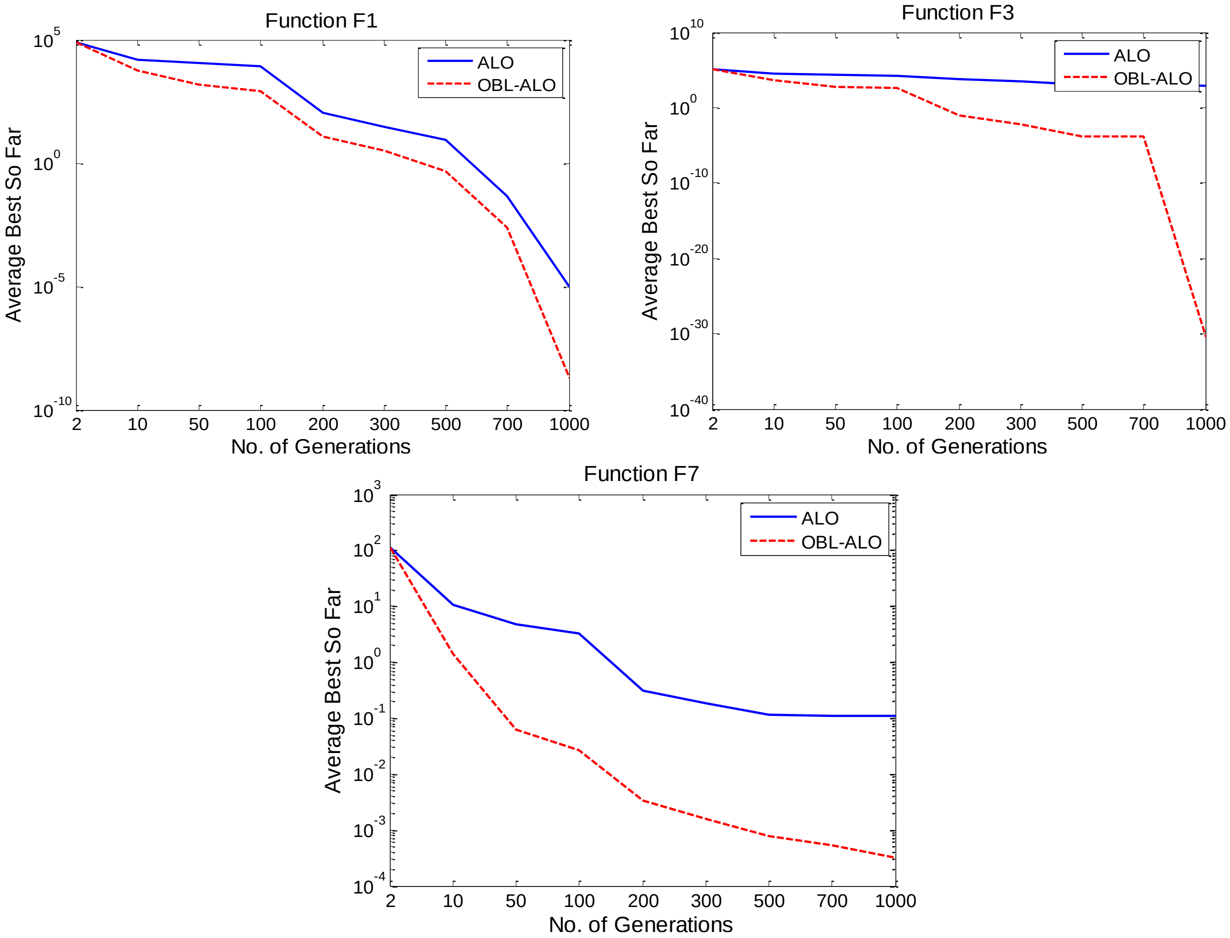


Figure 2.3: Convergence curves of three unimodal functions $F1$,$F3$ and $F7$

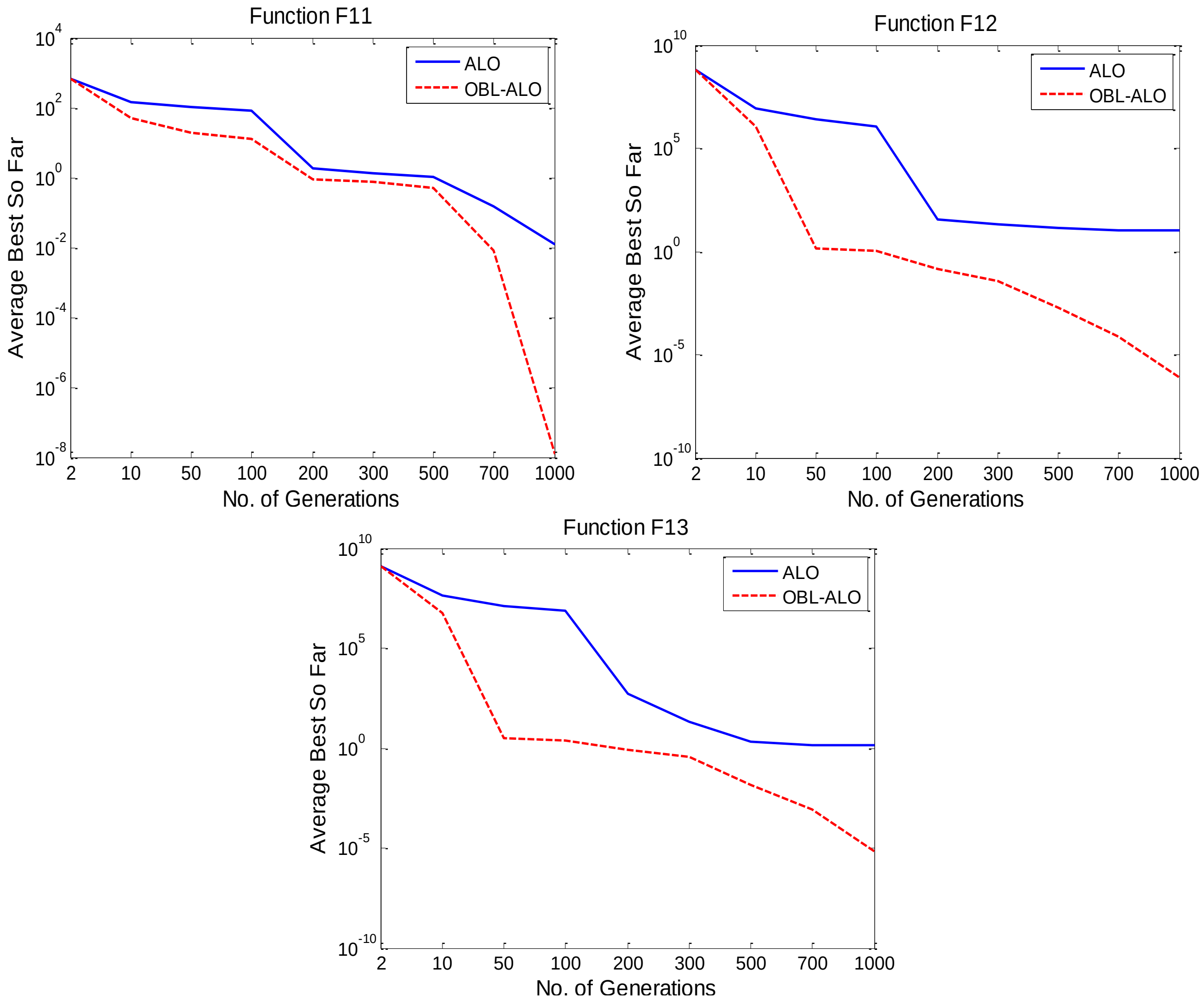


Figure 2.4: Convergence curves of three multimodal functions $F11$,$F12$ and $F13$

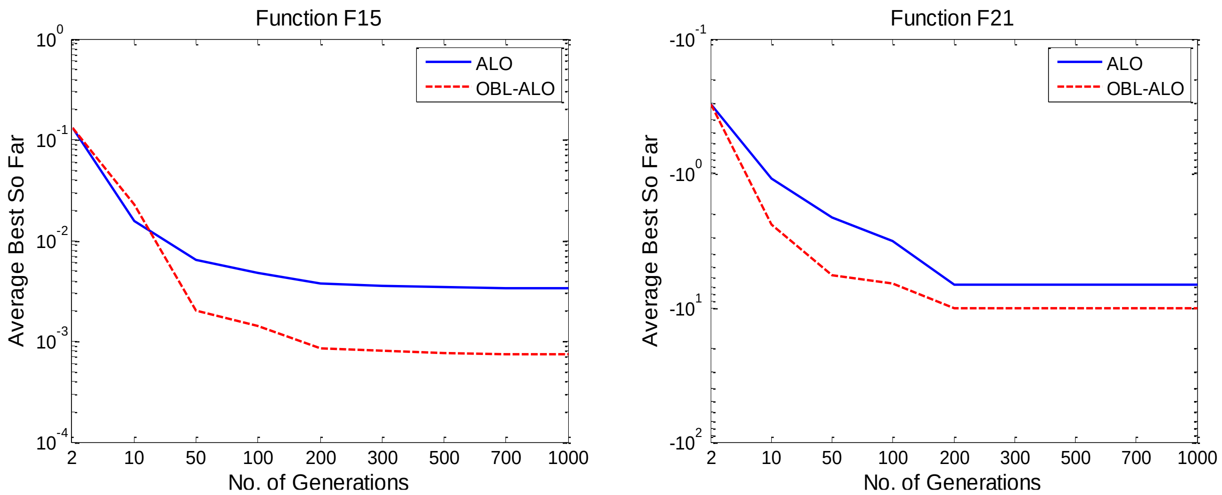


Figure 2.5: Convergence curves of two fixed dimension multimodal functions $F15$ and $F21$

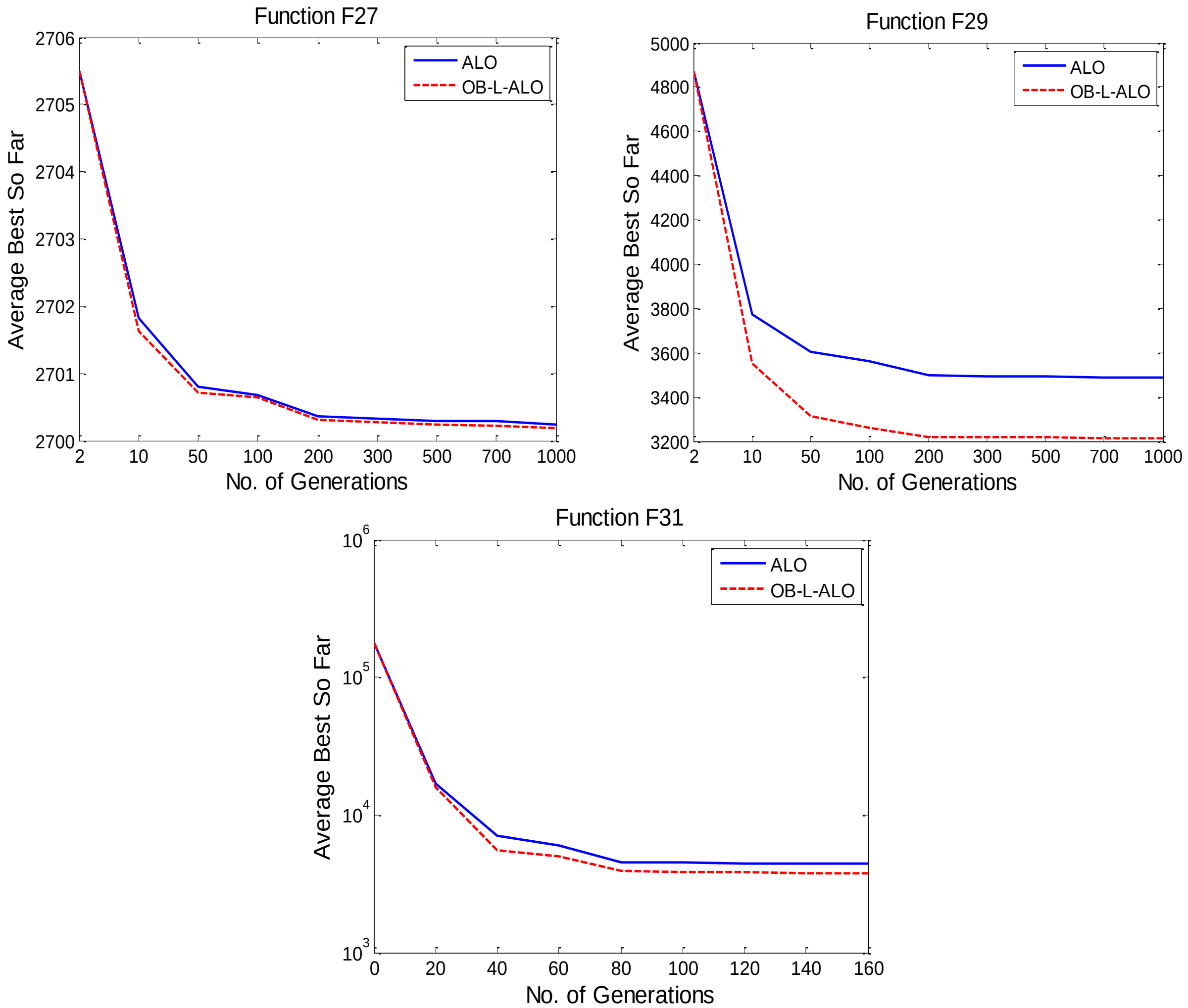


Figure 2.6: Convergence curves of three composition functions $F27$, $F29$ and $F31$

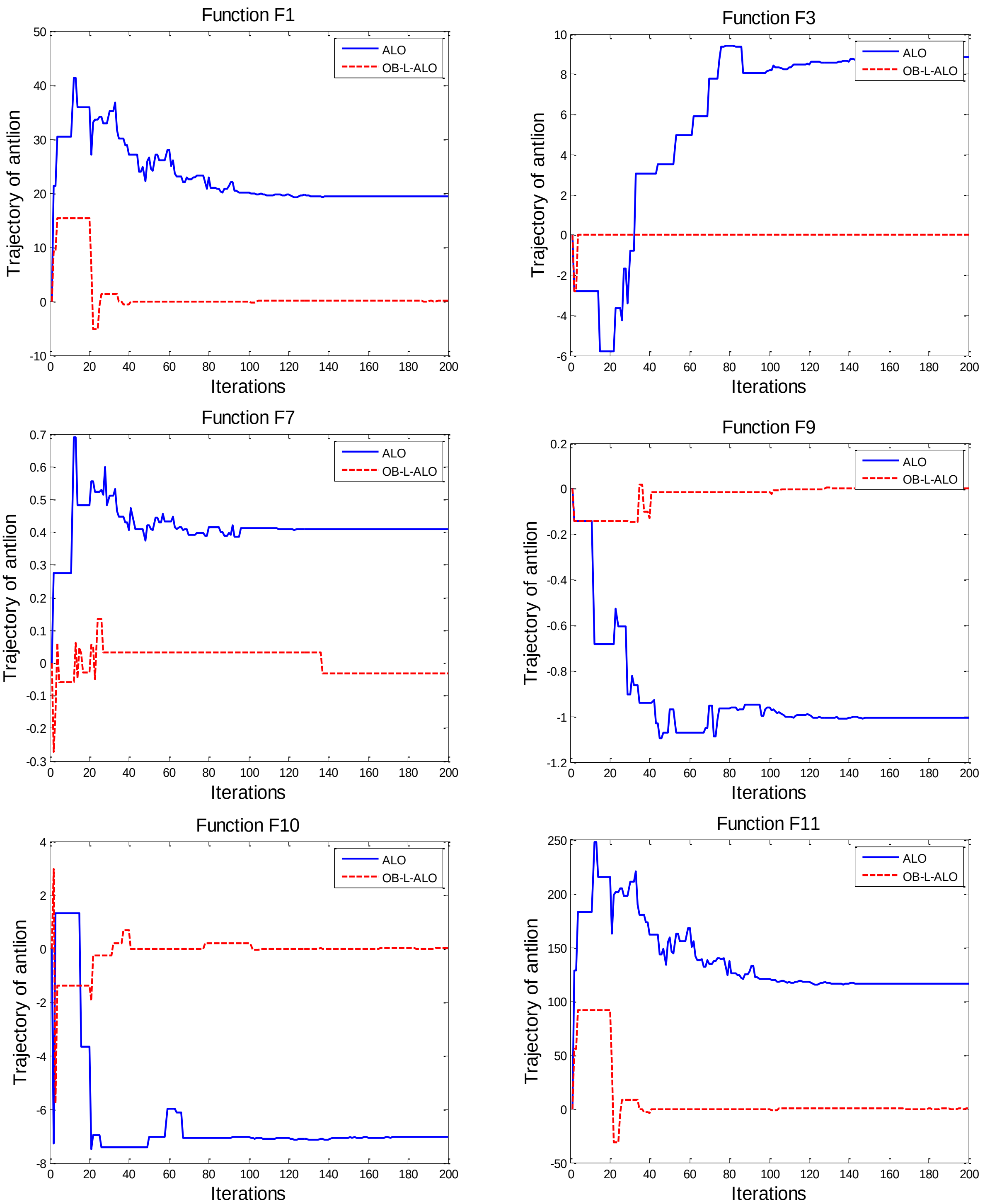


Figure 2.7: Trajectories of elite antlion

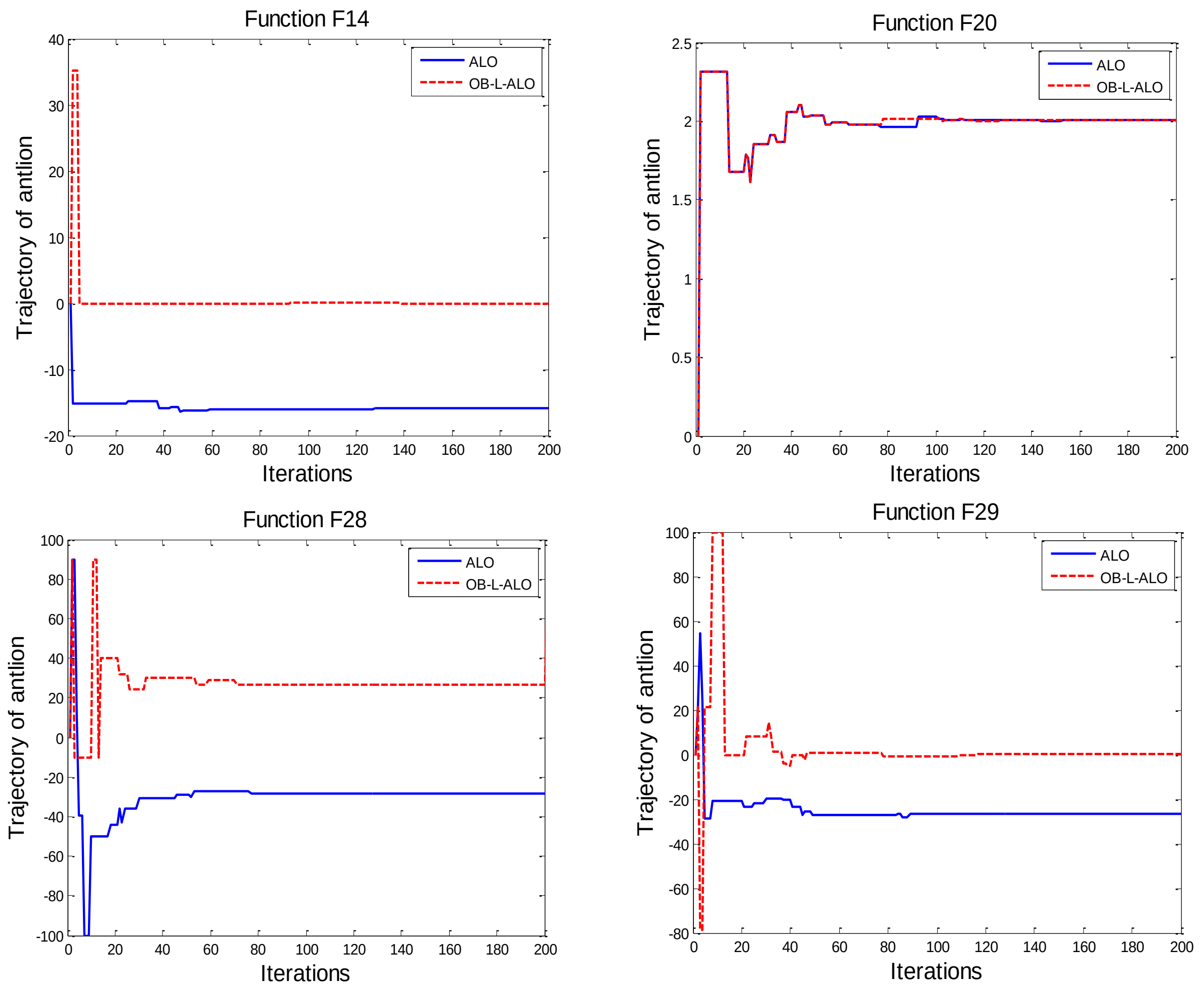


Figure 2.7: (Continued)

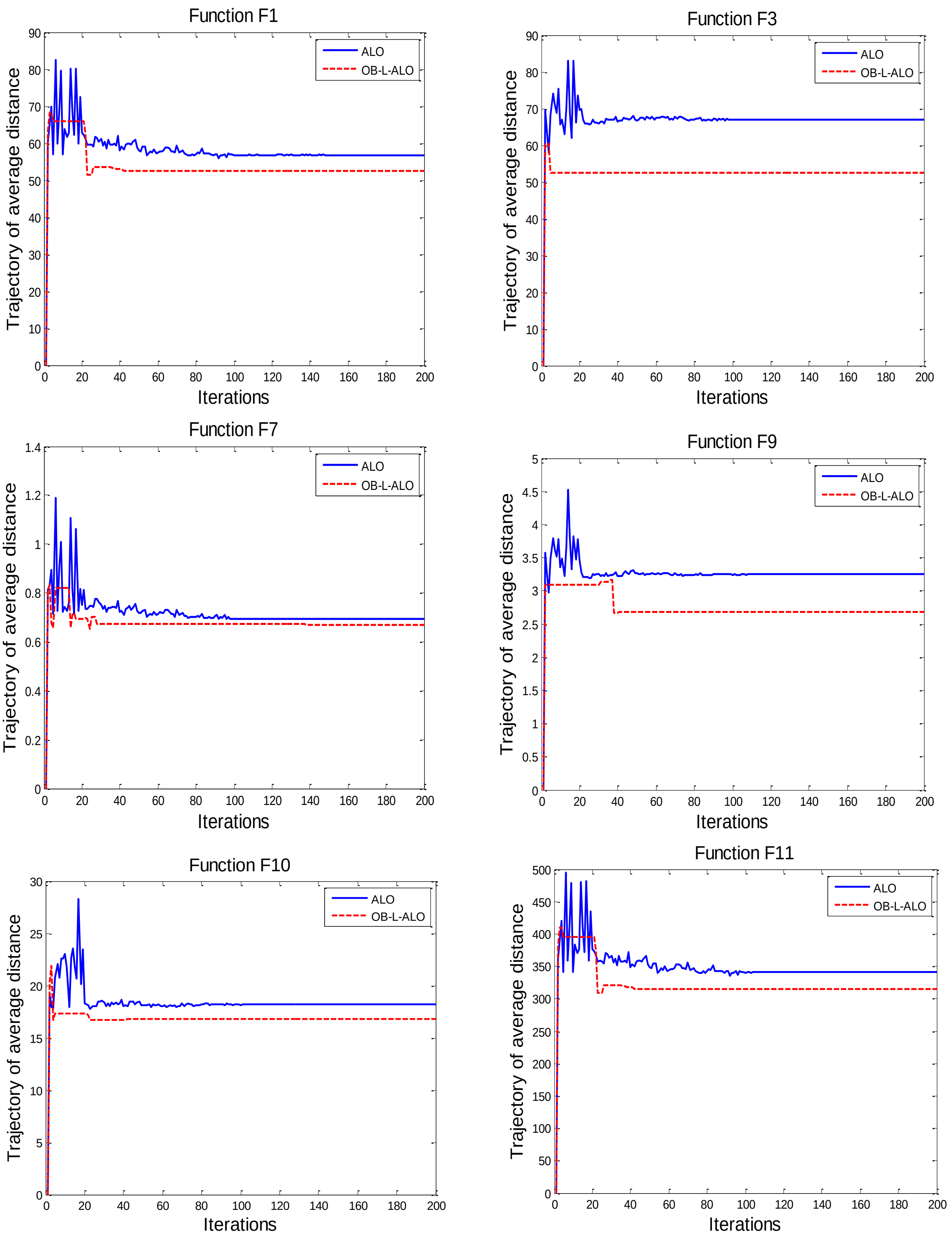


Figure 2.8: Trajectory behaviour of average distance between search agents

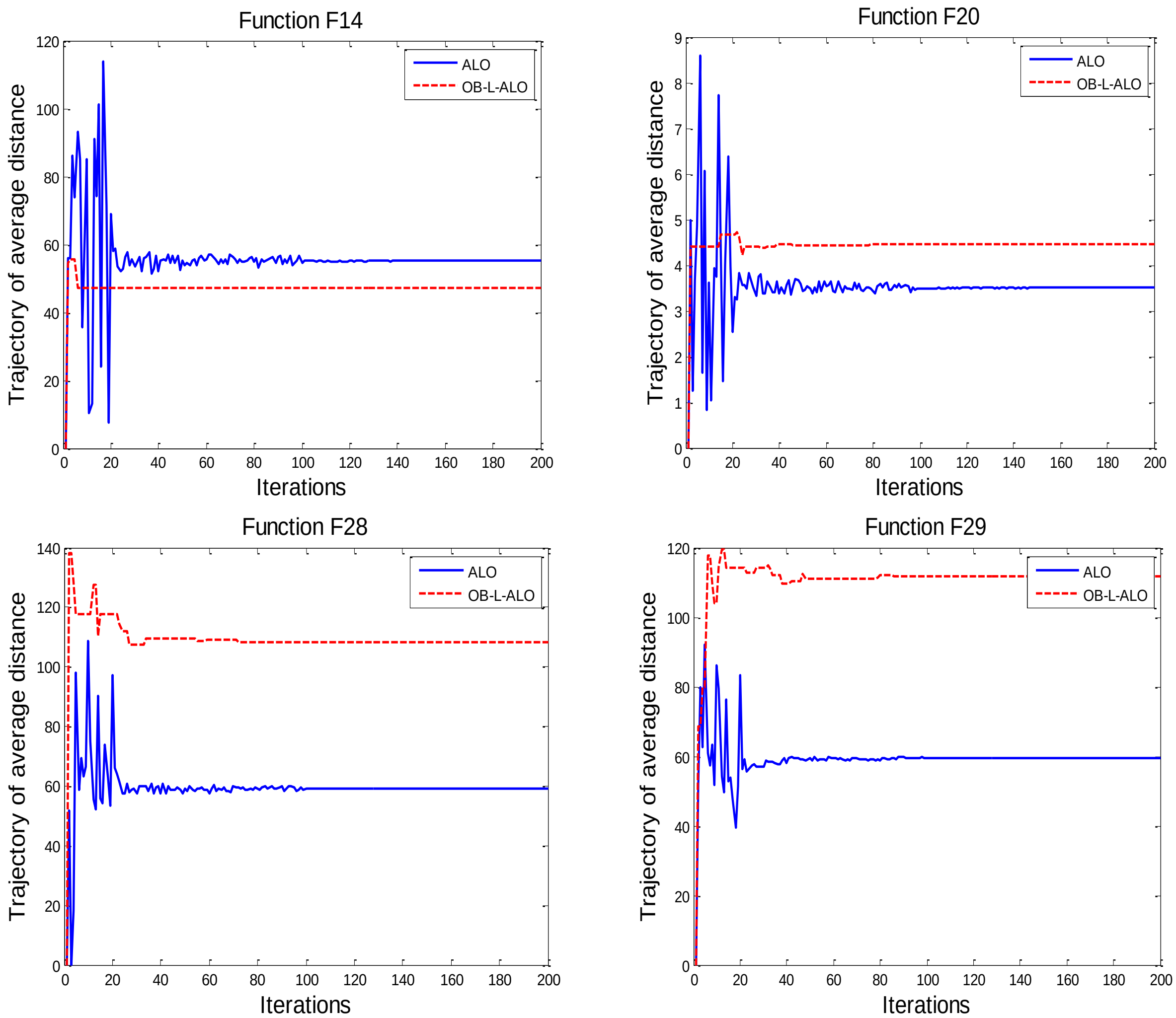


Figure 2.8: (Continued)

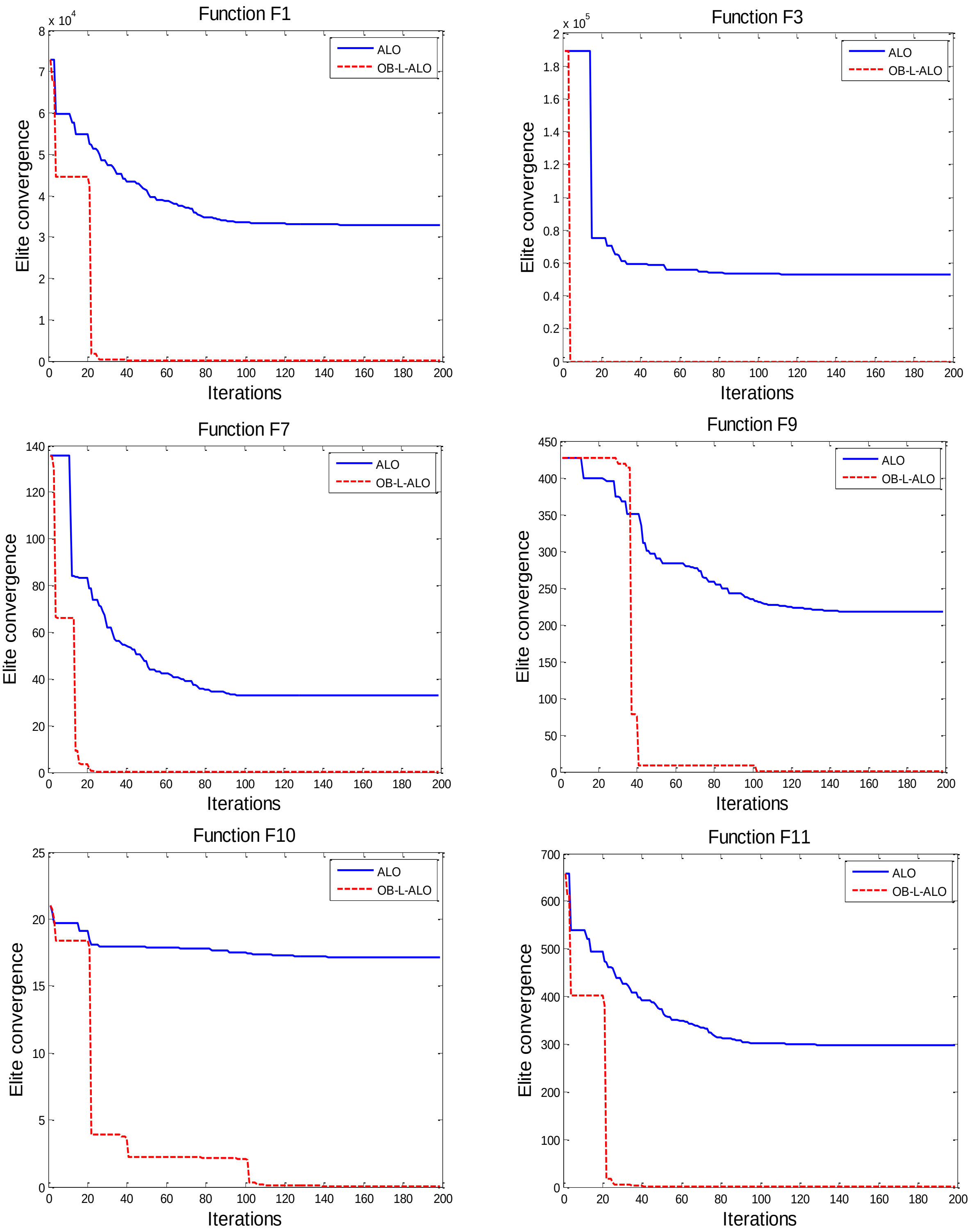


Figure 2.9: Elite convergence curves

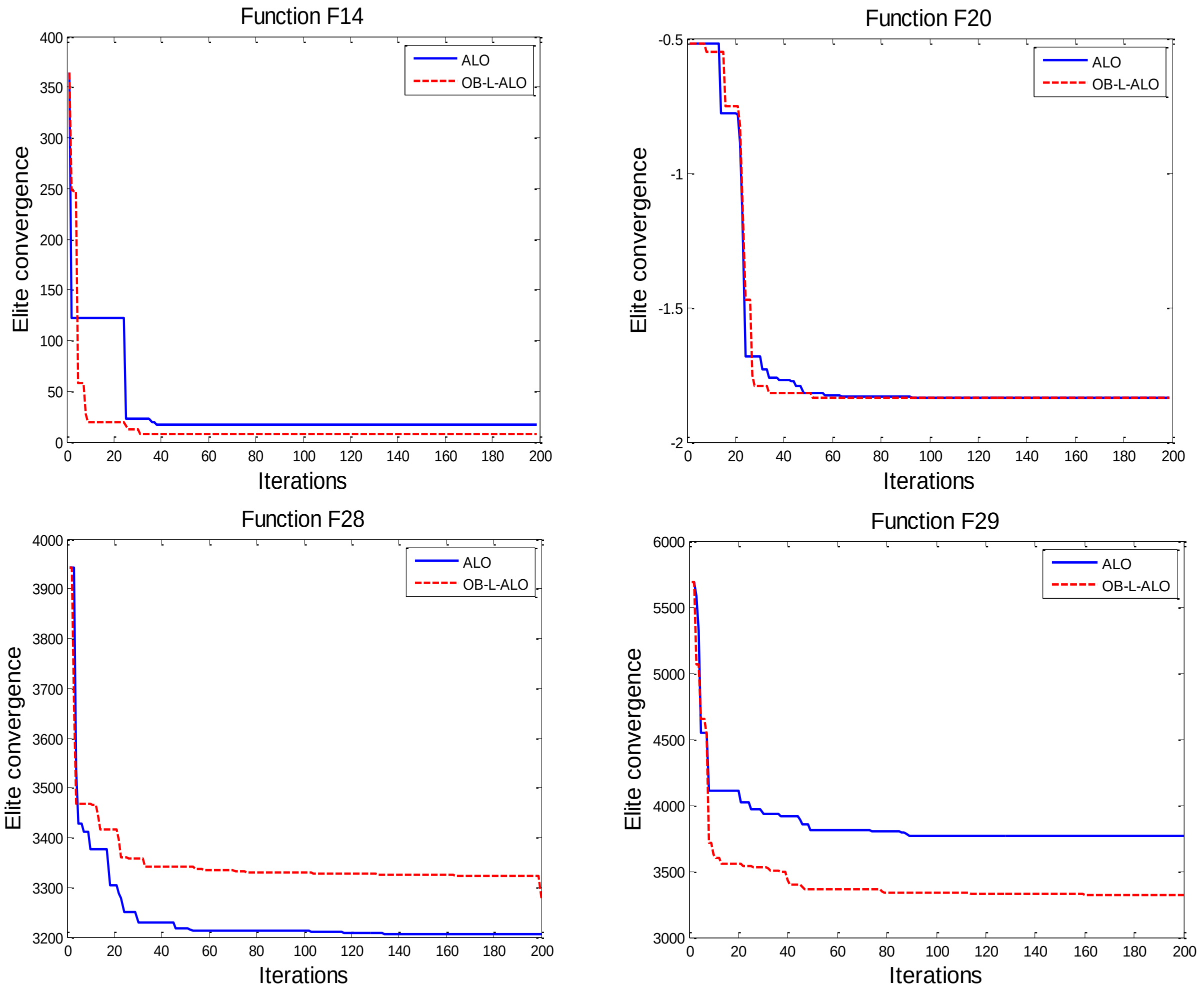


Figure 2.9: (Continued)

CHAPTER 3

Modified Opposition based Antlion Optimizer (ALO) using Acceleration Mechanisms for Continuous Optimization Problems

3.1 Introduction

This chapter proposes two acceleration mechanisms to enhance the convergence speed of classic antlion optimizer (ALO) along with the opposition based learning(OBL) model. Though the improvement suggested in chapter 2 are quite useful to enhance the performance of ALO yet it exhibits slow convergence and vulnerable to be trapped in local optima for some problems. In this chapter, the acceleration parameters are incorporated with opposition based learning (OBL) model which is applied to the best candidate solution obtained in each generation to enhance the exploration of opposite candidate solution along with the originally generated solutions.

Two unified approaches are utilized using acceleration parameters to achieve this objective: (1) Varying acceleration coefficient (2) Sine acceleration coefficient (SAC) with OBL mechanism. The OBL mechanism helps to avoid local optima by exploring the original randomly generated as well as opposite candidate solution within the search space. Then it is combined with Varying acceleration coefficient (ac) and Sine acceleration coefficient (SAC) to accelerate the exploitation at later generations. These two proposed algorithms are named as OB-ac-ALO and OB-SAC-ALO. The performance of both these versions of ALO are experimentally tested using 31 unconstrained continuous benchmark problems including unimodal, multimodal, fixed dimensional multimodal and composition functions of CEC 2014 as utilized in chapter 2 and shown in Appendix I. A non-parametric Wilcoxon ranksum test is performed to evaluate the performance of the proposed variants over the existing one. The numerical as well as graphical analysis of results are performed using various metrics including convergence curves, trajectories of solutions, average distance between search agents before and after improving the algorithm and elite convergence curves. The computational complexity of proposed variants is also presented.

The organization of this chapter is as follows:

Partial contents of this chapter has been published as:

- **Dinkar, S. K., & Deep, K. (2018). Accelerated Opposition-Based Antlion Optimizer with Application to Order Reduction of Linear Time-Invariant Systems. Arabian Journal for Science and Engineering, 44(3), 2213-2241. (SCIE, IF-1.092).**

Section 3.2 represents related literature. Section 3.3 describes two proposed algorithms namely OB-ac-ALO and OB-SAC-ALO with some brief information about applied strategies. Section 3.4 depicts the description of benchmark test functions and experimental setting. Section 3.5 exhibits numerical results and discussion based on characteristics of benchmark functions. Section 3.6 shows convergence and other algorithm analysis using various metrics to authenticate proposed algorithms. Section 3.7 presents concluding remarks.

3.2 Related literature

The basic conceptual material of OBL was proposed by Tizoosh in 2005[98] which suggests to consider the currently determined solutions as well as their opposite solutions simultaneously to approximate new and closer solutions to optima efficiently. This strategy is proved to be quite efficient in past research while improving the efficiency of metaheuristic algorithms. In [103], the OBL mechanism is employed with recently proposed sine cosine algorithm (SCA) [51] to enhance its performance for global optimization. In [104], the lévy flight random walk is integrated with OBL in artificial bee colony (ABC). The efficiency of differential evolution is enhanced using OBL [105]. The shuffled differential evolution [101] and shuffled frog leaping (SFL)[106] are also improved in terms of convergence acceleration with the use of opposition based modelling.

To maintain the proper exploitation after exploration, varying acceleration coefficient (ac) is utilized to propose first variant to enhance convergence speed of opposition based ALO as a new idea. After applying OBL, the newly generated candidate solutions may exhibit unexpected behaviour which can be adjusted by balancing the exploration and exploitation process [111]. This coefficient is hybridized with OBL mechanism to accelerate the exploitation after exploration so as to accelerate the convergence speed.

The other parameter namely sine acceleration coefficient (SAC) is used as the second scheme for adjusting the balance between exploration and exploitation. Sine acceleration coefficient (SAC) is employed with opposition based learning (OBL) as second strategy. This parameter is useful for global search at early generation and local search at later generation [112].

3.3 Proposed Variants

3.3.1 Variant 1: Opposition based ALO with Varying Acceleration Coefficient (OB-ac-ALO)

This section demonstrates the newly developed variant as scheme 1 to improve the performance of classical ALO. ALO suffers from stagnation to local optima and slow convergence. In this technique, few candidate solutions are updated towards the best solution but some get stuck and move away from the best one. Due to this process, the algorithm is not capable of avoiding local optima and subsequently, it is enforced into premature and slow convergence. Therefore the proposed variant takes the candidate solutions in opposite direction while applying opposition based learning model (OBL). The OBL technique is combined with acceleration coefficient (ac) to improve the convergence speed in order to maintain proper balance between exploration and exploitation. Both of these mechanism are explained in the following subsections:

The Opposition based learning (OBL) model

Due to the fact that the fitter solutions are closer to the optimum than the rest of the solutions, it is always better to search the closer candidate solutions. The theory of opposite numbers establishes that these numbers are closer to the optimum than the original numbers [102]. So it is better to approximate the initial along with opposite candidate solutions simultaneously and retain the fitter one to the next generation as described in chapter 2. With this approach, the strategy of OBL is applied in this work to improve the convergence. The definition of opposite numbers is defined in Section 2.3.2 of Chapter 2.

In proposed work, uniformly distributed random numbers are employed to generate the initial random solutions. The fitness (objective function value) of each candidate solution (antlion) is determined. Then the best candidate solution is found after sorting the obtained fitness values. This candidate solution is termed as elite. The positions of the candidate solutions are updated as mentioned in the classical ALO. Now, a new population is generated around the elite candidate solution to explore the opposite search region along with the updated population. The obtained opposite position around elite candidate solution is interchanged with the candidate of first dimension of the initial population. Then the fitter (elite) solutions are carried to the next generation after determining the fitness using new population. The process of evolution is continued until the predefined criteria is not satisfied. Mathematically, this process can be modelled as:

$$S_{A,i}(it) = L_i + U_i - S_{elite}(it) \tag{3.1}$$

where $S_{A,i}$ signifies the updated(opposite) position of candidate after employing OBL. S_{elite} is the best position for current iteration it.

Varying acceleration coefficient

Though the employment of OBL strategy guarantees the convergence acceleration in ALO, yet the generated opposite numbers may exhibit unexpected explorative behaviour. However it is a good idea to have enhanced exploration at early stages of generation but the algorithm must exploit with time as the number of generation increases. Hence the strategy is to hybridize the OBL with acceleration coefficient which enforces parameter adjustment while balancing exploration and exploitation.

To achieve this, a varying acceleration coefficient e_{ac} is employed which is decreased adaptively as the number of iteration increases. This parameter e_{ac} can be calculated as [111]:

$$e_{ac} = emax - it_{curr} \frac{emax-emin}{it_{max}} \quad (3.2)$$

Here $emax = 1$ as maximum value and $emin = 0.00001$ as minimum value. it_{curr} Indicates current iteration and it_{max} indicates maximum iteration.

After combining, the eq.(3.1) can be formulated using eq.(3.2) as:

$$S_{A,i}(it) = e_{ac} \times (LB_i + UB_i - S_{elite}(it)) \quad (3.3)$$

here e_{ac} is given in eq.(3.2).

Proposed OB-ac-ALO Variant

The above described strategies are integrated to define proposed improved opposition based acceleration coefficient (OB-ac-ALO) as a new technique described in eq.(3.3). As first strategy, the OBL is applied around elite candidate solution to generate the opposite candidate solutions for the next generation. This strategy enhances the exploration by using the initial as well as opposite candidate solutions and improves the convergence acceleration.

The second strategy is acceleration coefficient which is applied to adjust the parameter for balancing between exploration and exploitation. This strategy ensures the exploration at early stages and then exploitation at later stages of generation. Hence it promises the acceleration in convergence as the number of iteration increased.

The pseudo code of proposed OB-ac-ALO is shown in Table 3.1.

3.3.2 Variant 2: Opposition based ALO with Sine Acceleration Coefficient (OB-SAC-ALO)

This section demonstrates scheme 2 to propose OB-SAC-ALO to improve the performance of ALO by first applying opposition based learning (OBL) model to update the positions of the candidate solution. This mechanism is combined with sine acceleration coefficient (SAC) as a new idea to accelerate the convergence speed. The OBL mechanism is described in chapter 2. The second strategy can be explained as follows:

Sine Acceleration Coefficient

During the evolution process, it is observed that an optimization algorithm should exhibit global search ability with enhanced exploration at early generations and good exploitation at later generations to achieve global convergence. So, it is good a idea to utilize a parameter adjusting coefficient to adjust the abrupt and unexpected explorative behaviour at initial phase and then exploit at advanced stages. The second strategy i.e. sine acceleration coefficient which is proposed by Chen et al. [112] is integrated with OBL to adjust steadiness during process of evolution and improving acceleration in convergence.

Sine acceleration coefficient (SAC) is capable of harmonizing between global search and global convergence . Mathematically, the SAC can be formulated as:

$$c_{SAC} = \mu \times \sin\left(\left(1 - \frac{it_{curr}}{it_{max}}\right)\right) + \rho \tag{3.4}$$

where μ and ρ are two constants $(\mu = 2, \rho = 0.5)$, it_{curr}= current generation and it_{max}= maximum number of generations[112].

Using the above two strategies, the position equation can be updated using eq.(3.1) and (3.4) as follows:

$$S_{A,i}(it) = c_{SAC} \times (L_i + U_i - S_{elite}(it)) \tag{3.5}$$

Proposed OB-SAC-ALO variant

The above mentioned techniques are integrated to propose an improved opposition based sine acceleration coefficient (OB-SAC-ALO) variant. The OBL technique is employed to the elite candidate solution at each iteration to generate the opposite candidates. The fitness is then evaluated and compared to determine the new elite solution which is carried forward to the next generation throughout the evolution process. The OBL mechanism efficiently explores the opposite candidate solutions to accelerate the convergence.

The other mechanism is a parameter to adjust between global and local search during early and later stages of generations respectively. The parameter sine acceleration coefficient (SAC) is hybridized with OBL mechanism as shown in eq.(3.5). After applying SAC in conjunction with OBL, the convergence accelerates as the number of generations increase. Hence it promises to enhance the convergence as the number of iteration increases.

The pseudo code the proposed OB-SAC-ALO is depicted in Table 3.2.

3.4 Experimental Setup

The performance of proposed OB-ac-ALO and OB-SAC-ALO are demonstrated using a set of benchmark problems depicting different characteristics as used in Chapter 2. The results are obtained in terms of average, standard deviation, maximum and minimum of fitness values taken over 30 independent runs.

Benchmark Test functions

The similar set of benchmark functions utilized in chapter 2 are used to evaluate the performance of proposed techniques. These benchmark functions are depicted in Appendix I.

Experimental and Parameter Setting

The performance of any metaheuristic algorithm majorly depends on the stochastic behaviour of the employed operators in algorithm. The adequacy of parameter-setting can be established by performing continuous experiments such as “the size of initial population, stopping criteria and number of independent runs” etc. The experimental analysis establishes that the chances to find global optima increases as the number of generations increase. Based on this analysis, the size of population is chosen as 30 in this work.

To demonstrate the performance of proposed OB-ac-ALO and OB-SAC-ALO, they are evaluated and compared with classical ALO. For scalable problems (unimodal and multimodal functions), the experiments are performed for 10 and 30 dimensions keeping as similar as utilized in chapter 2. The performance of proposed variants are measured in terms of “average, standard deviation, maximum and minimum of objective function values”. Further, 30 independent runs have been performed by generating an initial population size of 30. “Stopping criteria is fixed at 500 and 1000 iterations for 10 and 30 dimensions respectively. For composition function, stopping criteria is taken as 1000 iteration for 10 dimension over 51 independent runs as desired in evaluation criteria of CEC 2014. All the experiments have been performed on MATLAB R2014a on Intel(R) Core(TM) i5-7200 CPU @ 2.50GHz-2.71 GHz with 8GB RAM".

3.5 Results and Discussion

Table 3.3, Table 3.4, Table 3.5, Table 3.6, Table 3.7 and Table 3.8 show the results for different dimensions of various categories of problems.

Comparison on of results on Unimodal functions($F1$-$F7$)

Due to the presence of single optima, these are appropriate to evaluate the exploitation ability of an optimization algorithm. This analysis can be authorized by observing the results as shown in Table 3.3 and Table 3.4 for 10 and 30 dimensions respectively. It can be clearly observed from the tables that the proposed OB-ac-ALO and OB-SAC-ALO algorithms depict significantly better results as compared to classical ALO. For functions $F1 - F4$, the average fitness values for proposed OB-ac-ALO are significantly better and able to attain global optima in very less function evaluations. For function $F5$ and $F7$ the results are slightly better. The proposed OB-SAC-ALO shows significantly better results for functions $F1 - F4$ and attains the global optima for functions $F2 - F3$ in 10 dimension and for $F2 - F4$ in 30 dimensions respectively than classical ALO. It is evidently shown in the tables that both the variants OB-ac-ALO and OB-SAC-ALO clearly outperform the classical ALO except the case of function $F6$. Out of the proposed variants, OB-ac-ALO perform better than the other proposed OB-SAC-ALO algorithm. Also the standard deviations for proposed algorithm are steady throughout and establishes the superiority and stability over the classical ALO. The performance of proposed OB-SAC-ALO and OB-ac-ALO show that the applied OBL mechanism effectively explores the nearby opposite candidate solution with respect to global optima as compared to classical ALO. It is also evidently proven that the hybridization of sine acceleration coefficient (SAC) and varying acceleration coefficient (ac) promise improved acceleration in convergence.

Comparison of results on Multimodal functions($F8$-$F13$)

These functions contain numerous local and global optima. To perform over such kind of problems, an algorithm must be capable of avoiding stagnation to local optima. These benchmark problems authorizes the exploration ability of proposed OB-ac-ALO and OB-SAC-ALO by visiting the unexplored region of the search space while avoiding entrapment to local optima and approximating nearby positions to global optima.

The obtained results are observed in Table 3.5 and Table 3.6 for 10 and 30 dimensions respectively. The results clearly reflect that the proposed OB-SAC-ALO exhibits significant improvements for functions $F8 - F13$. Proposed OB-SAC-ALO attains significantly better results for functions $F8$ in 30 dimension than classical ALO. Also OB-SAC-ALO is able to attain the global optima for function $F11$ and near to optima for function $F9$.

The other proposed OB-ac-ALO shows better results than classical ALO for all six functions $F8 - F13$ and able to get global optima for function $F9$ and $F11$. OB-ac-ALO outperforms the other proposed OB-SAC-ALO for function $F8 - F11$ but OB-SAC-ALO performs better than OB-ac-ALO. It is clearly evident that both the OB-ac-ALO and OB-SAC-ALO outperform the classical ALO. The obtained outcome evidently shows that the applied OBL mechanism in proposed techniques is quite capable of improving diversification of the search space by exploring the original and opposite solutions simultaneously. The integration of sine acceleration coefficient (SAC) and varying acceleration coefficient (ac) guarantees to achieve balance between exploration and exploitation. So, the applied proposed techniques are able to improve the efficiency and determine better solutions in comparison to classical ALO.

Comparison of results on Fixed Dimension Multimodal Functions ($F14 - F23$)

This category of functions are multimodal but has fixed dimension and cannot be scaled. Thus the behaviour of these functions is contrary to scalable multimodal functions. It is interesting to analyze the performance of these function while having fixed dimensions. The results for these functions when the stopping criteria is to achieve a maximum of 1000 iterations are shown in Table 3.7. It can be observed from the table that the proposed OB-SAC-ALO is significantly improved for functions $F19$,$F20$ and slightly better for function $F14$ than the original ALO and other proposed OB-ac-ALO. The results obtained for proposed OB-SAC-ALO, OB-ac-ALO and ALO are same for functions $F16$, $F17$ $F18$, $F22$ and F23 . OB-SAC-ALO and OB-ac-ALO show improved performance than classical ALO for function $F19$,$F20$ and $F21$.

Comparison of results on Composition Functions ($F24 - F31$)

The results are depicted in Table 3.8. It is evident from the table that average value obtained for OB-ac-ALO is better in terms of average values than OB-SAC-ALO and classical ALO for function $F24$. But OB-SAC-ALO performs better than OB-ac-ALO and classical ALO for function $F25$ and $F26$. and OB-SAC-ALO achieve better standard deviation for function $F26$. It also outperforms the OB-ac-ALO in terms of better average value though the standard deviation is better for OB-ac-ALO which show stability of OB-ac-ALO through the independent runs. OB-ac-ALO performs better for function $F27, F28, F29$ and $F30$ than the classical ALO and OB-SAC-ALO. But both of these algorithms are not able to beat classical ALO for $F31$.

3.6 Analysis of Results

The following metrics are used to analyze the performance of proposed variants:

3.6.1 Convergence Behaviour

3.6.2 Statistical Analysis- Wilcoxon Ranksum Test

3.6.3 Proposed Algorithm Analysis

Trajectory analysis

Trajectory analysis of average distance between search agents

Elite convergence curve

3.6.4 Computational Complexity

3.6.1 Convergence Behaviour

Convergence behaviour of OB-ac-ALO and OB-SAC-ALO is analyzed and compared with classical ALO. To perform this analysis, functions of different characteristics have been chosen and convergence graph have been drawn to verify the performance of proposed variants in terms of convergence behaviour. The number of generations and the best average value of objective functions for 30 independent runs have been plotted on X-axis and log scale of Y-axis respectively.

Three curves in Figure 3.1 depicts the convergence behaviour of unimodal functions $F1$,$F3$ and $F7$. It is observed from the figures that the proposed OB-SAC-ALO and OB-ac-ALO initiates convergence starting from earlier generations for all three unimodal functions in comparison to classical ALO. OB-ac-ALO attains the global optima in less than 200 generations for function $F1$ and less than 100 generation for function $F3$. OB-SAC-ALO also converge in 700 generation for function $F3$. These curves clearly exhibit that the inclusion of proposed OBL mechanism promises to search the closer candidate solutions starting from initial generations as compared to classical ALO. Then the varying acceleration coefficient and sine acceleration coefficient (SAC) integrated with OBL mechanism are capable of balancing between diversification and intensification to speed up the convergence. The behaviour shown by curves guarantee to converge global or near global optima [110].

The curves shown in Figure 3.2 and Figure 3.3 depict the convergence analysis of “two scalable multimodal functions($F10, F11$) and two fixed multimodal functions($F14$ and $F20$)”.

It is clearly apparent from the figure that the proposed OB-SAC-ALO and OB-ac-ALO exhibit sharp convergence starting from initial generation for functions $F10$ and $F11$ and end with the same behaviour till the last generations. Both the proposed variants OB-ac-ALO and OB-SAC-ALO are able to attain the optima in 300 and 700 generations respectively. In Figure 3.3, the proposed OB-SAC-ALO exhibits substantial convergence for function $F14$ as compared to other two algorithms but for function $F20$,the proposed algorithm shows better convergence than ALO. This conduct shows that the inclusion of OBL model promises to enhance diversification by to exploring the opposite candidate solution simultaneously with the original solutions and able to avoid local optima entrapment.

Figure 3.4 shows the convergence curves of three composition functions namely $F27, F28$ and $F29$. These functions are able to testify the performance of the proposed algorithms over complex composition problems. To solve these problems, the parameters of algorithm must be properly tuned and able to balance exploration and exploitation efficiently. Convergence analysis of Function $F27$ depicts slightlle better improvement and for functions $F28$ and $F29$ clearly show significant improvement as shown in Table 3.7 and Table 3.8.

3.6.2 Statistical analysis-Wilcoxon Ranksum Test

Wilcoxon ranksum test is a non-parametric test which is used to verify the statistical significance of OB-ac-ALO and OB-SAC-ALO proposed in this chapter. The data for this test is drawn in pair from two algorithms to be tested. The dimensions of the data drawn is 30 which is taken after simulating 30 independent runs. The confidence level is chosen as 0.95 to test the null hypothesis. The obtained p-values of the two pairs of ALO with proposed OB-ac-ALO and OB-SAC-ALO respectively proposed in this chapter are shown in Table 3.9. In Table, it is shown "that '+' means significant statistical difference (rejection of null hypothesis) at 0.05 level of significance, '-' designates no significant difference and '=' shows that the sample drawn from both the algorithms are same and no comparison is possible".

It can be clearly observed from the table that both OB-ac-ALO and OB-SAC-ALO proves to be statistical significant as compared to classical ALO for 19 and 18 problems respectively and behaves statistically similar for 5 problems.

3.6.3 Analysis of Proposed Variants

The analysis of proposed OB-ac-ALO and OB-SAC-ALO is performed using certain metrics and comparison is performed with classical ALO. This analysis establishes the improvement in speeding up the convergence and provide proper balance between diversification and intensification. The analysis is described as follows:

Analysis of trajectory

This analysis is used to demonstrate the behaviour of trajectories (position) of the candidate solutions before and after applying the improvement strategies. The trajectories are drawn for the first variable initiating from very first iteration to the maximum number of iteration (200). This analysis is capable of verifying the ability of proposed OB-ac-ALO and OB-SAC-ALO to find out the nearest positions of the global optima. The diagrammatic representation is described in Figure 3.5.

This analysis is performed by taking few benchmark functions of different characteristics including three unimodal function ($F1$,$F3, F7$), two multimodal functions ($F9$,$F11$),two fixed dimensions multimodal functions ($F14$,$F20$) and three composition functions($F27, F28, F29$). It is clearly evident from the figures that the inclusion of varying acceleration coefficient (ac) and sine acceleration coefficient (SAC) with OBL technique ensures the enhanced exploration of the search region at early generations and then converges at later generations for both the proposed algorithms. However, it is visible from the curves of function $F1$,$F3$ and $F7$ that the OB-ac-ALO exhibits good convergence at early generations than the OB-SAC-ALO but able to find the optima or near to optima as results shown in result tables. The OB-SAC-ALO also converges at later generations after showing enhanced exploration at early generations.

For function $F9$ and $F11$, both the algorithms show similar behaviour as exhibited in earlier functions. This is due to the fact that the proposed OB-ac-ALO and OB-SAC-ALO enhances exploration and show the searching ability in efficient manner and thus able to exhibit steady behaviour by approximating the opposite solutions near to global optima. The trajectories of compositions function $F27, F28$ and $F29$ are represented in the figure. The trajectories for OB-ac-ALO and OB-SAC-ALO are almost similar for function $F27$ and $F28$ but it shows steep convergence at later generations in function $F29$ for OB-ac-ALO.

Trajectory analysis of average distance between search agents

The concept of OBL mechanism states that there are 50% chances of a candidate solution either generated randomly or opposite candidates to be nearer to the optimal solution. It is better to select opposite candidate solution in spite of generating additional random numbers during evolution process for accelerating convergence. This concept is experimentally established by determining the average of distance (Euclidean) between the search agent before and after applying the updating strategies.

Formally, the Euclidean distance using two points is depicted in chapter 2 by formulating eq.(2.6) for D-dimensional search space and in eq.(2.7) for one dimension.

The obtained distance for proposed OB-ac-ALO and OB-SAC-ALO is analyzed and compared with classical ALO to authorize the improvement. The experiments are performed for 200 iterations over first variable and ten benchmark functions of wide variety including 3 unimodal($F1, F3, F7$),two multimodal($F9, F11$), two fixed dimension multimodal($F14, F20$) and three composition functions ($F27, F28, F29$) to verify the performance of proposed algorithms.

The determined absolute values of average distance of proposed OB-ac-ALO and OB-SAC-ALO can be analyzed by the curves shown in Figure 3.6. It is observed that the determined average distance using OB-ac-ALO approximates the closer distance to the global optima starting from initial generations as compared to OB-SAC-ALO and classical ALO for unimodal functions ($F1, F3$ and $F7$) and multimodal functions ($F9$ and $F11$). OB-SAC-ALO shows fluctuation of average distance in initial generations due to enhanced exploration after applying the proposed strategies. Then it becomes steady and smooth at later generations which determines the closer distance to the global optima for unimodal as well as multimodal functions. Since the obtained optimum for these functions are better than the classical ALO as shown in result tables, thus the determined average distance proved to be closer than the classical ALO. It is also observed from the curves that the newly developed OB-ac-ALO is able to determine closer distance for function $F14$ while comparing with OB-SAC-ALO and ALO. OB-SAC-ALO shows closer distance than classical ALO but it is accomplished at later generations. Similarly, function $F20$ shows abrupt behaviour at later generations for proposed algorithms though obtained optima for function $F20$ is better than the classical ALO. The curves showing average distance for compositions functions $F27, F28$ and $F29$ are depicted in figure. Overall analysis depicts that the proposed algorithms guarantees acceleration in convergence.

Elite convergence curve

These curves authorize the convergence of best (elite) candidate solutions of first variable. It is clearly evident from the curves that the convergence shown by the proposed algorithms OB-ac-ALO and OB-SAC-ALO is quite steep from initial generations as compared to classical ALO for unimodal and multimodal functions. This behaviour establishes that the rate of convergence accelerates as the number of iteration increases and tend to approximate the global or near to global optima as early as possible. The elite convergence behaviour of the composition functions also reflect that the fitness value of elite is almost similar at initial generations but OB-ac-ALO shows steep convergence as compared to OB-SAC-ALO and ALO. It is also evident from the results tables. This graphical interpretation proves that the employed acceleration coefficient (ac) in OB-ac-ALO and sine acceleration coefficient (SAC) in OB-SAC-ALO are well capable of accelerating the convergence rate of the elite candidate solutions and tends to find the global optima in early generations.

3.6.4 Computational complexity

The computational complexities for classical ALO and OB-L-ALO proposed in chapter 2 are described and compared in previous chapter. It is ensured in this chapter while proposing these two algorithms that there is no inclusion of any extra loop or function evaluation. This fact establishes that there is no change in the asymptotic computational complexities of both of these algorithms. It is observed that the complexities in worst case scenario of both OB-ac-ALO and OB-SAC-ALO are derived as $O(it_{max} * D * N) * O\ (f(X))$ where it_{max} maximum number of iterations, D represents dimensions, N is population size and $f(X)$ is the objective function.

3.7 Conclusion

In this chapter, two variants are proposed to improve the convergence of classical ALO. One algorithm namely OB-ac-ALO is proposed by employing a new parameter namely varying acceleration coefficient (ac) and hybridized with OBL mechanism. Similarly, the second algorithm OB-SAC-ALO is proposed by employing sine acceleration coefficient (SAC) in conjunction with OBL technique. The performance of both OB-ac-ALO and OB-SAC-ALO are verified using a set of 31 benchmark functions of wide variety. The OBL mechanism is applied around elite (best) candidate solution with acceleration coefficient in OB-ac-ALO and with sine acceleration coefficient (SAC) in proposed OB-SAC-ALO. This modification ensures the exploration of original as well as opposite candidate solution and adjusted with applied acceleration parameters to ensure acceleration in convergence.

Then the modified variants are analyzed using various metrics such as convergence curve, trajectories of best candidate solutions, average distance between the search agents before and after modifications and elite convergence curves. A non-parametric test namely Wilcoxon ranksum test is also performed to verify the statistical significance of the proposed OB-ac-ALO and OB-SAC-ALO and compared with the classical ALO.

Few observations can be highlighted in this study. The inclusion of opposition based learning model around the best candidate solution in both the proposed approaches is able to approximate the opposite and closer candidate solution to the global optima. This is established using the analysis of the average distance between search agents before and after applying the modifications. The use of acceleration coefficient guarantees the acceleration in convergence as validated by the convergence curves and trajectories of the elite candidate at each iteration while adjusting the proper balance between exploration and exploitation. However, both the approaches suffer with increase of computation time as compared to classical ALO for high dimensional problems. Yet, preservation of elitism property ensures the significant drop in average of objective function values at early stage. The employability of the proposed approaches proved to be significantly better than the classical ALO.

Table 3.1: Pseudo code of Proposed OB-ac-ALO algorithm

Input: *Population Size N, Maximum iteration* it_{max}*,lower bound L, Upper bound U and dimension D*

Output: *The best candidate solution* S_{elite}

Randomly initialize the initial population N of ants and antlions
Determine the objective(fitness) function value of antlions
Find out the best(with min fitness) antlion as the elite S_{elite}
Initialize iteration no. it_{curr}*=2*
    while $(\boldsymbol{it_{curr} \leq it_{max}})$
        for every ant $(\boldsymbol{i = 1, 2, 3, \ldots, N})$
          Find an ant lion S_{sel} *using Roulette wheel*
          Modify lower L and upper U boundaries with equations Eqs. $L^d(it) = \frac{L^d(it)}{I}$ *and* $U^d(it) = \frac{U^d(it)}{I}$*)*
            for every dimension $(\boldsymbol{j = 1, 2, 3, \ldots, D})$
              Perform random walk $rw_A(it)$ *around* S_{sel} *and* $rw_E(it)$ *around* S_{elite}
              using uniform distribution with Eq. $r_w(S_{A,n}^d)$ = [$cumsum(2r(it_1)$- 1), $cumsum(2r(it_2)$-1)… $cumsum(2r(it_{max})$-1)]
              Normalize random walk using Eqs. $r_w(S_{A,n}^d)$ = [$cumsum(2r(it_1)$-1), $cumsum(2r(it_2)$-1)… $cumsum(2r(it_{max})$-1)] *and*
              $S_{A,n}^d(it) = \frac{(S_{A,n}^d(it) - \min r_W(S_{A,n}^d))(U^d(it) - L^d(it))}{\max r_W(S_{A,n}^d) - \min r_W(S_{A,n}^d)} + L^d(it)$
            end for
          Modify the position of ant using Eq. $S_{A,n}^d(it) = \frac{r_{wA}(it) + r_{wE}(it)}{2}$
        end for
        Modify position of ant by applying coefficient e_{ac} *with opposition based*
        learning using Eq. $S_{A,i}(it) = e_{ac} \times (LB_i + UB_i - S_{elite}(it))$
        for every ant $(\boldsymbol{i = 1, 2, 3, \ldots, N})$
          Determine the fitness of all ants
        end for
        Substitute an antlion with its respective ant if it becomes fitter using Eq.
        $S_{AL,j}(it) = S_{A,i}(it)$ *if* $f(S_{A,i}(it)) < f(S_{AL,j}(it))$
        Modify S_{elite} *if an ant lion becomes fitter than the elite*
        Increment iteration i.e. it_{curr}*=*it_{curr}*+1*
    end while
Return elite

Table 3.2: Pseudo code of Proposed OB-SAC-ALO algorithm

Input: *Population Size N, Maximum iteration* it_{max}*,lower bound L, Upper bound U and dimension D*

Output: *The best candidate solution* S_{elite}

Randomly initialize the initial population N of ants and ant lions
Determine the objective(fitness) function value of antlions
Find out the best(with min fitness) antlion as the elite S_{elite}
Initialize iteration no. $it_{curr}=2$
    while $(it_{curr} \leq it_{max})$
        for every ant $(i = 1, 2, 3, \ldots, N)$
            Find an ant lion S_{sel} *using Roulette wheel*
            Modify lower L and upper U boundaries with equations Eqs. $L^d(it) = \frac{L^d(it)}{I}$ *and* $U^d(it) = \frac{U^d(it)}{I}$*)*
            for every dimension $(j = 1, 2, 3, \ldots, D)$
                Perform random walk $rw_A(it)$ *around* S_{sel}
                and $rw_E(it)$ *around* S_{elite} *using uniform distribution with Eq.* $r_w(S^d_{A,n}) = [cumsum(2r(it_1)\text{- }1), cumsum(2r(it_2)\text{-}1) \ldots cumsum(2r(it_{max})\text{-}1)]$
                Normalize random walk using Eqs. $r_w(S^d_{A,n}) = [cumsum(2r(it_1)\text{-}1), cumsum(2r(it_2)\text{-}1) \ldots cumsum(2r(it_{max})\text{-}1)]$ *and*
                $S^d_{A,n}(it) = \frac{(S^d_{A,n}(it) - \min r_w(S^d_{A,n}))(U^d(it) - L^d(it))}{\max r_w(S^d_{A,n}) - \min r_w(S^d_{A,n})} + L^d(it)$
            end for
            Modify the position of ant using Eq. $S^d_{A,n}(it) = \frac{r_{w_A}(it) + r_{w_E}(it)}{2}$
        end for
        Modify position of ant by applying coefficient c_{SAC} *with opposition based*
        learning using Eq. $S_{A,i}(it) = c_{SAC} \times (L_i + U_i - S_{elite}(it))$
        for every ant $(i = 1, 2, 3, \ldots, N)$
            Determine the fitness of all ants
        end for
        Substitute an antlion with its respective ant if it becomes fitter using Eq
        $S_{AL,j}(it) = S_{A,i}(it) \; if \; f(S_{A,i}(it)) < f(S_{AL,j}(it))$
        Modify S_{elite} *if an ant lion becomes fitter than the elite*
        Increment iteration i.e. $it_{curr} = it_{curr} + 1$
    end while
Return elite

Table 3.3: Average, Standard Deviation, Minimum, and Maximum of objective function values of unimodal test functions (10D)

Function	***Methods***	***Ave.***	***Stan. Dev.***	***Min.***	***Max.***
$F1$	**OB-ac-ALO**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-SAC-ALO	7.330E-33	2.680E-32	0.000E+00	1.130E-31
	ALO	7.870E-09	5.490E-09	2.390E-09	**2.580E-08**
$F2$	**OB-ac-ALO**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-SAC-ALO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	ALO	4.860E-01	8.850E-01	1.550E-05	**2.800E+00**
$F3$	**OB-ac-ALO**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-SAC-ALO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	ALO	8.310E-02	1.850E-01	1.360E-04	**8.750E-01**
$F4$	**OB-ac-ALO**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-SAC-ALO	1.160E-15	3.270E-15	2.230E-21	1.730E-14
	ALO	3.180E-03	6.160E-03	1.020E-04	**3.290E-02**
$F5$	**OB-ac-ALO**	**2.270E-04**	**3.680E-04**	**1.410E-03**	**2.400E-07**
	OB-SAC-ALO	4.080E-04	6.500E-04	7.610E-07	3.040E-03
	ALO	6.810E+01	1.990E+02	1.020E-04	**1.070E+03**
$F6$	OB-ac-ALO	2.920E-08	4.920E-08	5.320E-09	2.640E-07
	OB-SAC-ALO	1.250E-08	1.610E-08	3.240E-09	8.530E-08
	ALO	**8.390E-09**	**5.220E-09**	**2.150E-09**	2.220E-08
$F7$	**OB-ac-ALO**	**2.620E-04**	**1.920E-04**	**6.980E-04**	**9.880E-06**
	OB-SAC-ALO	4.030E-04	3.930E-04	1.200E-05	1.710E-03
	ALO	2.210E-02	1.230E-02	1.810E-03	**5.630E-02**

Table 3.4: Average, Standard Deviation, Minimum, and Maximum of objective function values of unimodal test functions (30D)

Function	***Methods***	***Ave.***	***Stan. Dev.***	***Min.***	***Max.***
$F1$	**OB-ac-ALO**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-SAC-ALO	2.280E-54	1.250E-53	0.000E+00	6.830E-53
	ALO	9.450E-06	5.540E-06	1.720E-06	2.110E-05
$F2$	**OB-ac-ALO**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-SAC-ALO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	ALO	3.650E+01	5.170E+01	2.090E-02	1.340E+02
$F3$	**OB-ac-ALO**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-SAC-ALO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	ALO	9.550E+02	5.080E+02	2.470E+02	2.160E+03
$F4$	**OB-ac-ALO**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-SAC-ALO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	ALO	1.210E+01	3.650E+00	4.630E+00	1.960E+01
$F5$	**OB-ac-ALO**	**5.530E-04**	**6.800E-04**	**2.950E-03**	**1.160E-05**
	OB-SAC-ALO	2.090E-03	1.930E-03	1.990E-04	8.050E-03
	ALO	1.040E+02	3.120E+02	1.640E+01	1.740E+03
$F6$	OB-ac-ALO	1.820E-05	1.600E-05	7.350E-05	1.140E-06
	OB-SAC-ALO	1.760E-05	1.030E-05	3.160E-06	4.360E-05
	ALO	**9.490E-06**	**6.100E-06**	**1.320E-06**	**2.920E-05**
$F7$	**OB-ac-ALO**	**1.670E-04**	**1.020E-04**	**3.760E-04**	**7.520E-06**
	OB-SAC-ALO	2.710E-04	2.670E-04	1.500E-05	1.050E-03
	ALO	1.100E-01	3.570E-02	5.700E-02	2.250E-01

Table 3.5: Average, Standard Deviation, Minimum, and Maximum of objective function values of multimodal test functions (10D)

Function	***Methods***	***Ave.***	***Stan. Dev.***	***Min.***	***Max.***
$F8$	**OB-ac-ALO**	**-2.880E+03**	**6.160E+02**	**-1.970E+03**	**-3.970E+03**
	OB-SAC-ALO	-2.430E+03	2.820E+02	-3.040E+03	-1.970E+03
	ALO	-2.470E+03	4.260E+02	-3.730E+03	-1.920E+03
$F9$	**OB-ac-ALO**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-SAC-ALO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	ALO	2.480E+01	1.080E+01	6.960E+00	**5.070E+01**
$F10$	**OB-ac-ALO**	**8.880E-16**	**4.010E-31**	**8.880E-16**	**8.880E-16**
	OB-SAC-ALO	1.100E-14	1.110E-14	4.440E-15	4.000E-14
	ALO	3.240E-01	5.540E-01	2.070E-05	**1.650E+00**
$F11$	**OB-ac-ALO**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-SAC-ALO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	ALO	2.140E-01	8.720E-02	6.640E-02	**4.060E-01**
$F12$	OB-ac-ALO	2.460E-08	2.300E-08	**2.100E-02**	1.250E-09
	OB-SAC-ALO	**1.490E-08**	**1.410E-08**	**7.020E-10**	**5.600E-08**
	ALO	1.790E+00	1.670E+00	8.230E-09	**5.610E+00**
$F13$	**OB-ac-ALO**	**1.470E-03**	**3.800E-03**	**1.100E-02**	**1.070E-08**
	OB-SAC-ALO	2.870E-03	6.220E-03	1.340E-08	2.100E-02
	ALO	1.800E-03	4.940E-03	3.840E-09	2.100E-02

Table 3.6: Average, Standard Deviation, Minimum, and Maximum of objective function values of multimodal test functions (30D)

Function	***Methods***	***Ave.***	***Stan. Dev.***	***Min.***	***Max.***
$F8$	**OB-ac-ALO**	**-1.250E+04**	**9.340E+01**	**-1.240E+04**	**-1.260E+04**
	OB-SAC-ALO	-6.490E+03	7.260E+01	-6.540E+03	-6.270E+03
	ALO	-5.440E+03	4.190E+01	-5.540E+03	-5.420E+03
$F9$	**OB-ac-ALO**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-SAC-ALO	1.890E-15	1.040E-14	0.000E+00	5.680E-14
	ALO	1.970E+00	8.390E-01	5.890E-04	3.520E+00
$F10$	**OB-ac-ALO**	**8.880E-16**	**4.010E-31**	**8.880E-16**	**8.880E-16**
	OB-SAC-ALO	4.440E-15	4.010E-30	4.440E-15	4.440E-15
	ALO	1.870E+00	6.900E-01	1.020E-03	3.090E+00
$F11$	**OB-ac-ALO**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-SAC-ALO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	ALO	1.190E-02	1.260E-02	3.470E-04	3.860E-02
$F12$	OB-ac-ALO	8.360E-07	6.410E-07	3.370E-06	1.780E-07
	OB-SAC-ALO	**6.650E-07**	**3.070E-07**	**1.470E-07**	**1.380E-06**
	ALO	1.000E+01	4.340E+00	5.750E+00	2.330E+01
$F13$	OB-ac-ALO	2.870E-03	6.720E-03	3.090E-02	3.830E-06
	OB-SAC-ALO	**1.480E-03**	**3.810E-03**	**2.100E-06**	**1.100E-02**
	ALO	1.320E+00	3.150E+00	1.140E-05	1.050E+01

Table 3.7: Average, Standard Deviation, Minimum, and Maximum of objective function values of fixed dimension multimodal functions

Function	***Methods***	***Ave.***	***Stan. Dev.***	***Min.***	***Max.***
$F14$	OB-ac-ALO	1.860E+00	1.310E+00	6.900E+00	9.980E-01
	OB-SAC-ALO	**1.490E+00**	**7.700E-01**	**9.980E-01**	**3.970E+00**
	ALO	1.790E+00	1.340E+00	1.950E-08	4.950E+00
$F15$	OB-ac-ALO	3.380E-03	6.790E-03	2.050E-02	5.980E-04
	OB-SAC-ALO	3.380E-03	6.790E-03	3.080E-04	2.050E-02
	ALO	4.180E-03	7.370E-03	5.550E-04	**2.040E-02**
$F16$	**OB-ac-ALO**	**-1.030E+00**	**6.780E-16**	**-1.030E+00**	**-1.030E+00**
	OB-SAC-ALO	-1.030E+00	6.780E-16	-1.030E+00	-1.030E+00
	ALO	-1.030E+00	6.780E-16	-1.030E+00	-1.030E+00
$F17$	**OB-ac-ALO**	**3.980E-01**	**1.690E-16**	**3.980E-01**	**3.980E-01**
	OB-SAC-ALO	**3.980E-01**	**1.690E-16**	**3.980E-01**	**3.980E-01**
	ALO	3.980E-01	1.690E-16	3.980E-01	3.980E-01
$F18$	**OB-ac-ALO**	**3.000E+00**	**0.000E+00**	**3.000E+00**	**3.000E+00**
	OB-SAC-ALO	3.000E+00	0.000E+00	3.000E+00	3.000E+00
	ALO	3.000E+00	0.000E+00	3.000E+00	3.000E+00
$F19$	OB-ac-ALO	-7.540E+00	2.690E+00	-2.630E+00	-1.020E+01
	OB-SAC-ALO	**-8.970E+00**	**2.170E+00**	**-1.020E+01**	**-5.100E+00**
	ALO	-6.280E+00	2.930E+00	-1.020E+01	-2.630E+00
$F20$	OB-ac-ALO	-7.900E+00	2.960E+00	-2.770E+00	-1.040E+01
	OB-SAC-ALO	**-8.950E+00**	**2.710E+00**	**-1.040E+01**	**-2.770E+00**
	ALO	-6.670E+00	3.420E+00	-1.040E+01	**-1.840E+00**
$F21$	OB-ac-ALO	-7.470E+00	3.410E+00	-2.420E+00	-1.050E+01
	OB-SAC-ALO	**-8.080E+00**	**3.350E+00**	**-1.050E+01**	**-2.420E+00**
	ALO	-6.270E+00	3.660E+00	-1.050E+01	-1.860E+00
$F22$	**OB-ac-ALO**	**-3.863E+00**	**3.160E-15**	**-3.863E+00**	**-3.863E+00**
	OB-SAC-ALO	**-3.863E+00**	**3.160E-15**	**-3.863E+00**	**-3.863E+00**
	ALO	-3.863E+00	3.160E-15	-3.863E+00	-3.863E+00
$F23$	**OB-ac-ALO**	**-3.322E+00**	**1.810E-15**	**-3.322E+00**	**-3.322E+00**
	OB-SAC-ALO	**-3.322E+00**	**1.810E-15**	**-3.322E+00**	**-3.322E+00**
	ALO	-3.322E+00	1.810E-15	-3.322E+00	-3.322E+00

Table 3.8: Average, Standard Deviation, Minimum, and Maximum of objective function values of composition benchmark functions

Function	***Methods***	***Ave.***	***Stan. Dev.***	***Min.***	***Max.***
$F24$	**OB-ac-ALO**	**2.500E+03**	**0.000E+00**	**2.500E+03**	**2.500E+03**
	OB-SAC-ALO	2.630E+03	3.270E-04	2.630E+03	2.630E+03
	ALO	2.630E+03	1.275E-03	2.630E+03	2.630E+03
$F25$	OB-ac-ALO	2.540E+03	2.017E+01	2.518E+03	2.600E+03
	OB-SAC-ALO	**2.536E+03**	**1.312E+01**	**2.518E+03**	**2.580E+03**
	ALO	2.538E+03	2.512E+03	2.577E+03	1.477E+01
$F26$	OB-ac-ALO	2.690E+03	2.100E+01	2.620E+03	2.700E+03
	OB-SAC-ALO	**2.680E+03**	**2.320E+01**	**2.620E+03**	**2.700E+03**
	ALO	2.680E+03	2.430E+01	2.630E+03	2.700E+03
$F27$	**OB-ac-ALO**	**2.700E+03**	**1.100E-01**	**2.700E+03**	**2.700E+03**
	OB-SAC-ALO	2.700E+03	1.040E-01	2.700E+03	2.700E+03
	ALO	2.700E+03	1.090E-01	2.700E+03	**2.700E+03**
$F28$	**OB-ac-ALO**	**2.840E+03**	**9.170E+01**	**2.700E+03**	**2.900E+03**
	OB-SAC-ALO	2.970E+03	1.720E+02	2.700E+03	3.100E+03
	ALO	3.010E+03	1.440E+02	2.700E+03	3.110E+03
$F29$	**OB-ac-ALO**	**3.000E+03**	**0.000E+00**	**3.000E+03**	**3.000E+03**
	OB-SAC-ALO	3.480E+03	2.950E+02	3.160E+03	3.890E+03
	ALO	3.490E+03	2.920E+02	3.170E+03	**3.930E+03**
$F30$	**OB-ac-ALO**	**7.220E+05**	**1.470E+06**	**3.100E+03**	**3.670E+06**
	OB-SAC-ALO	7.220E+05	1.470E+06	3.100E+03	3.670E+06
	ALO	8.270E+05	1.520E+06	3.100E+03	**3.670E+06**
$F31$	OB-ac-ALO	4.460E+03	3.980E+02	3.880E+03	6.190E+03
	OB-SAC-ALO	4.460E+03	3.980E+02	3.880E+03	6.190E+03
	ALO	**4.420E+03**	**4.140E+02**	**3.880E+03**	**6.170E+03**

Table 3.9: P-values of Wilcoxon ranksum test

Function	OB-SAC-ALO vs ALO		OB-ac-ALO vs ALO	
	p-values	conclusion	p-values	conclusion
$F1$	1.720E-12	+	1.21E-12	+
$F2$	1.210E-12	+	1.21E-12	+
$F3$	1.210E-12	+	1.21E-12	+
$F4$	1.210E-12	+	1.21E-12	+
$F5$	3.020E-11	+	3.02E-11	+
$F6$	1.700E-03	-	1.53E-02	-
$F7$	3.020E-11	+	3.02E-11	+
$F8$	6.480E-12	+	1.33E-12	+
$F9$	1.700E-12	+	1.20E-12	+
$F10$	1.180E-12	+	1.18E-12	+
$F11$	1.210E-12	+	1.21E-12	+
$F12$	3.020E-11	+	3.02E-11	+
$F13$	1.700E-08	+	2.29E-08	+
$F14$	1.870E-01	-	8.78E-01	-
$F15$	9.880E-01	-	5.15E-01	-
$F16$	N/A	=	N/A	=
$F17$	N/A	=	N/A	=
$F18$	N/A	=	N/A	=
$F19$	4.630E-04	+	7.05E-02	+
$F20$	4.350E-02	+	4.03E-02	+
$F21$	1.050E-01	-	3.20E-02	+
$F22$	N/A	=	N/A	=
$F23$	N/A	=	N/A	=
$F24$	2.33E-11	+	1.36E-20	+
$F25$	6.30E-01	-	8.67E-01	-
$F26$	8.98E-01	-	2.56E-01	-
$F27$	3.37E-02	+	7.84E-03	+
$F28$	6.63E-05	+	8.88E-10	+
$F29$	5.38E-02	+	1.39E-20	+
$F30$	8.93E-01	-	7.42E-01	-
$F31$	4.99E-01	-	4.99E-01	-
	23/31(18 ‘+’ , 5 ‘=’)		24/31(19 ‘+’ , 5 ‘=’)	

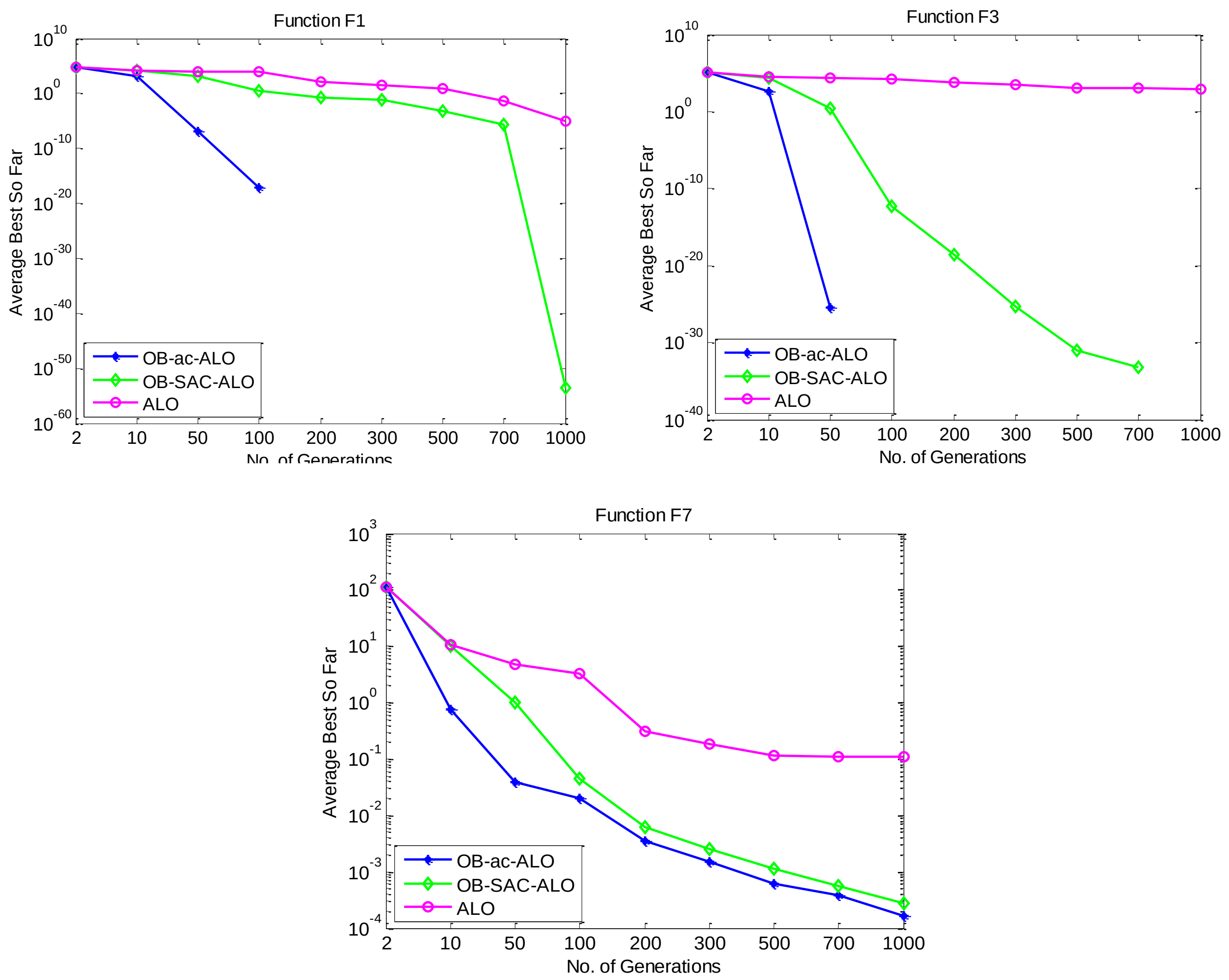


Figure 3.1**:** Convergence curves of three of the unimodal functions $F1$,$F3$ and $F7$

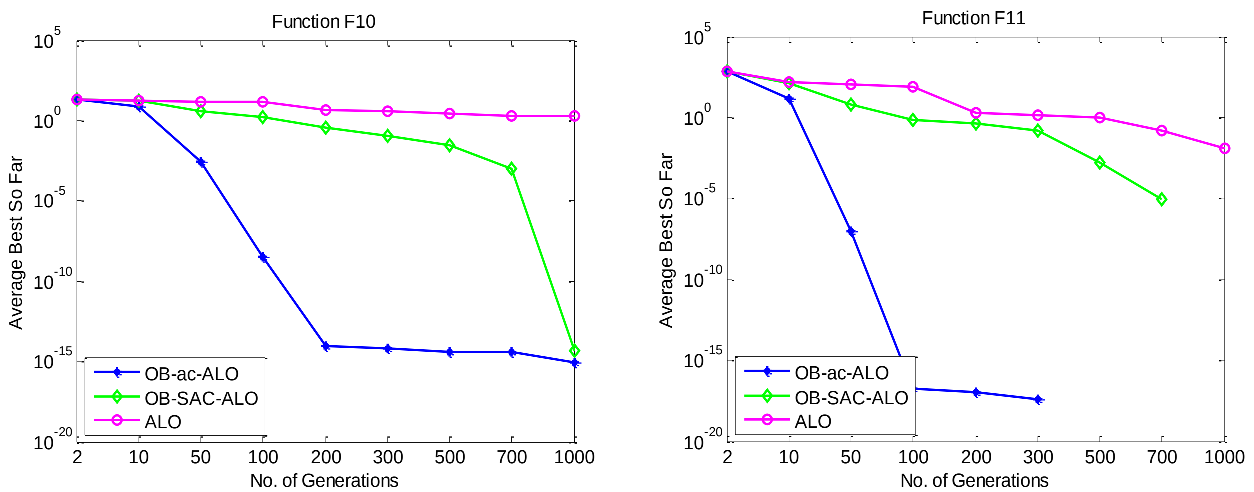


Figure 3.2**:** Convergence curves of three multimodal functions $F10$ and $F11$

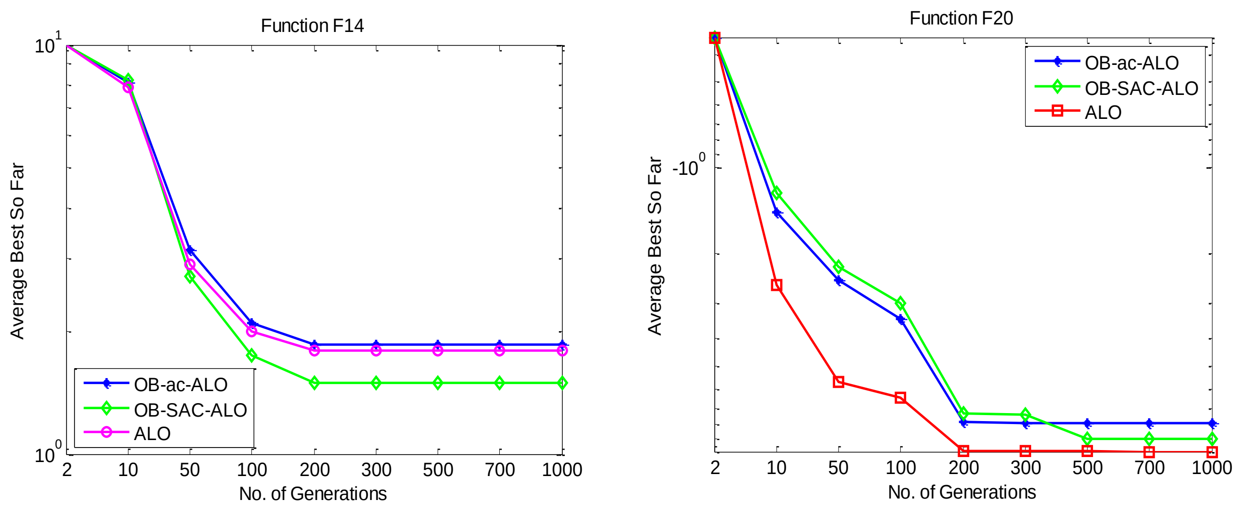


Figure 3.3**:** Convergence curves of two fixed dimension multimodal functions $F14$ and $F20$

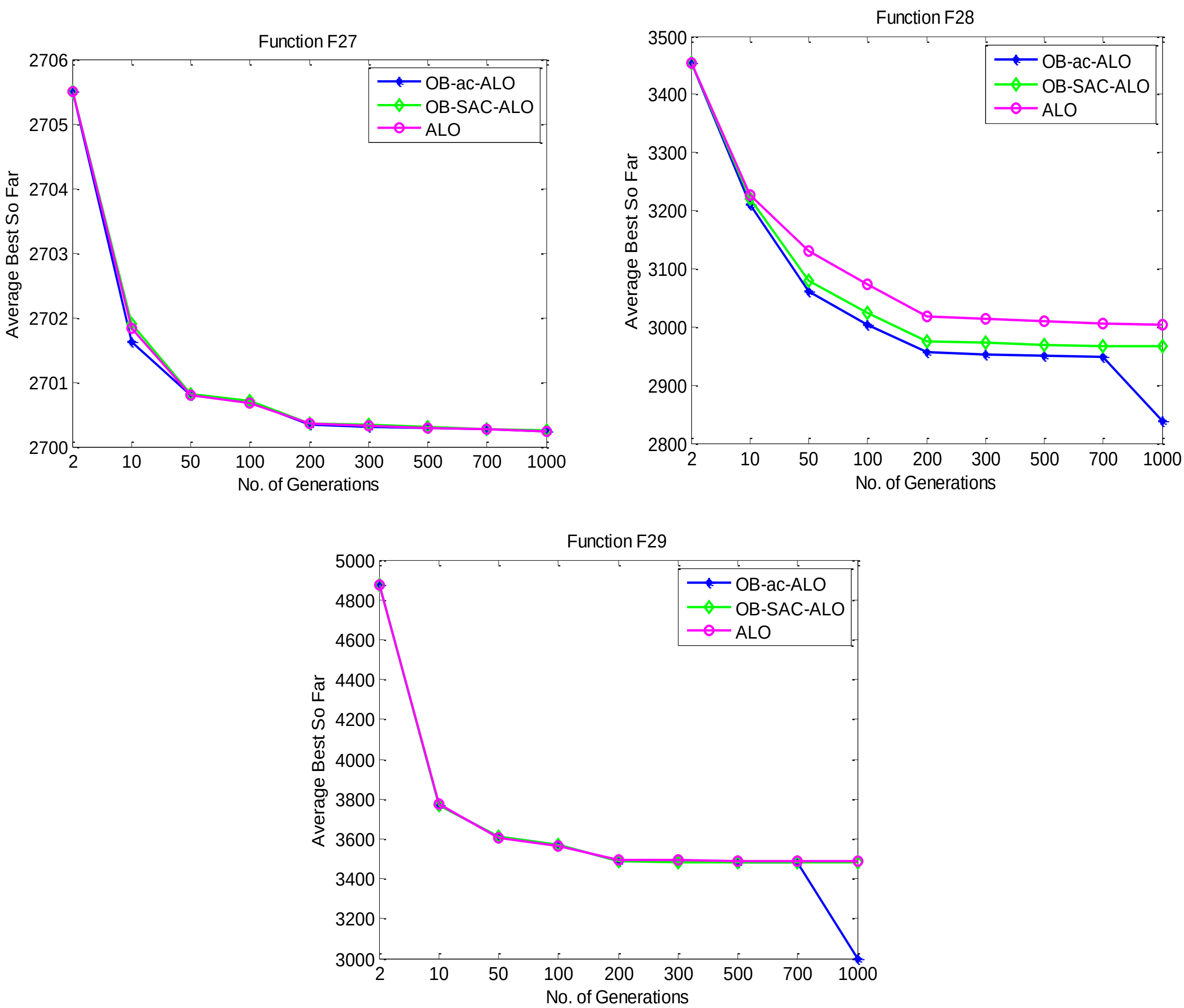


Figure 3.4: Convergence curves of three composition functions $F27, F28$ and $F29$

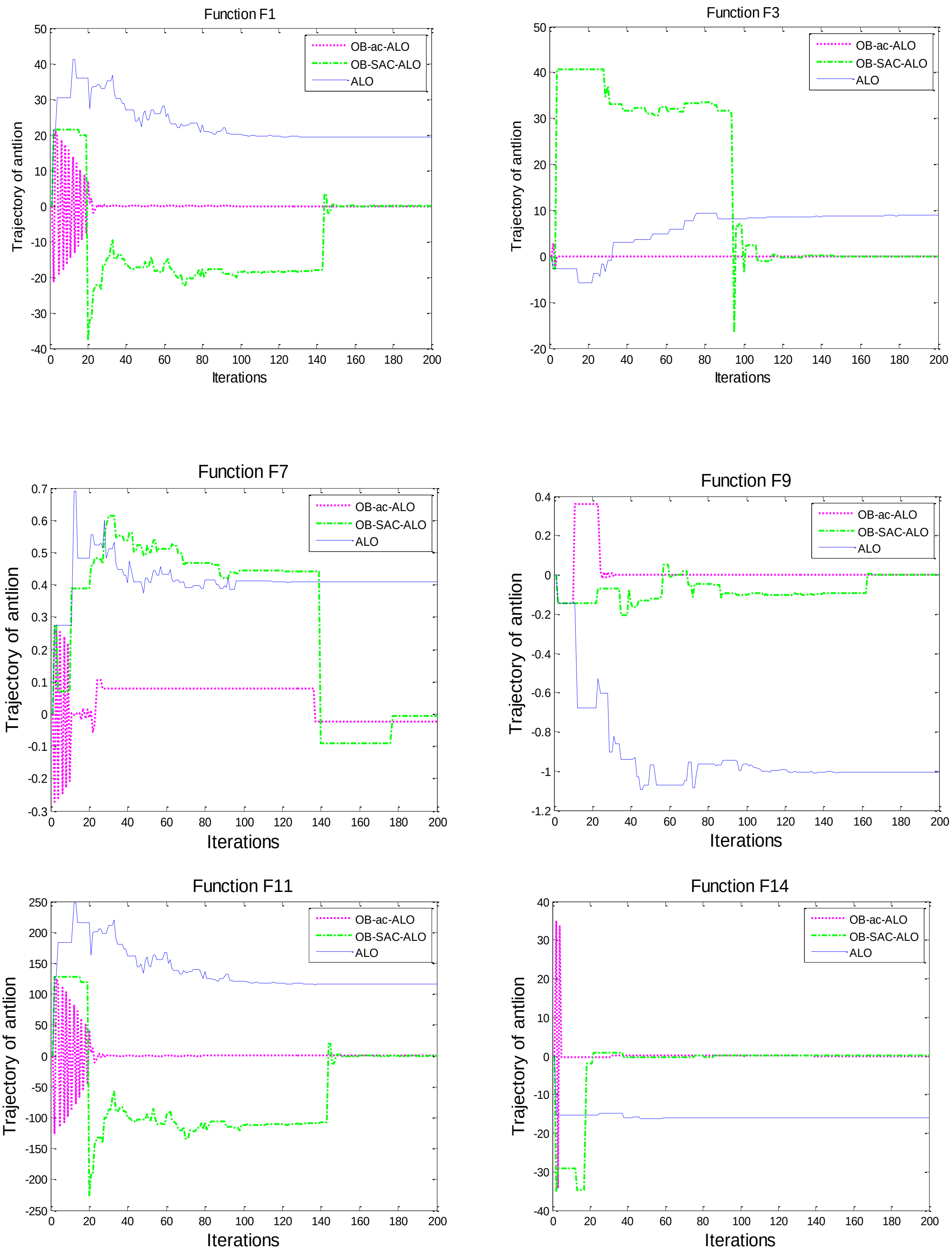


Figure 3.5: Trajectories of antlion

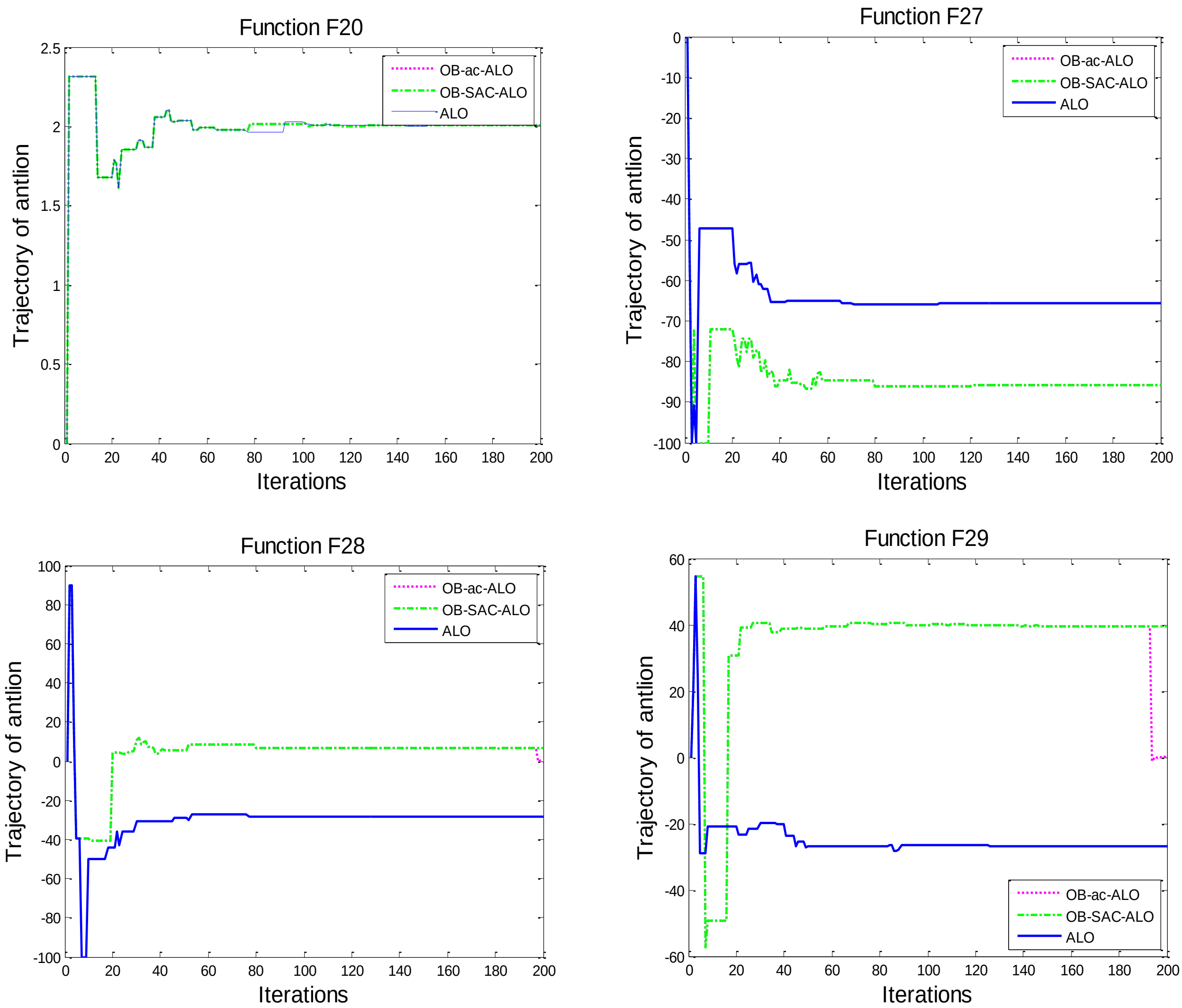


Figure 3.5: (Continued)

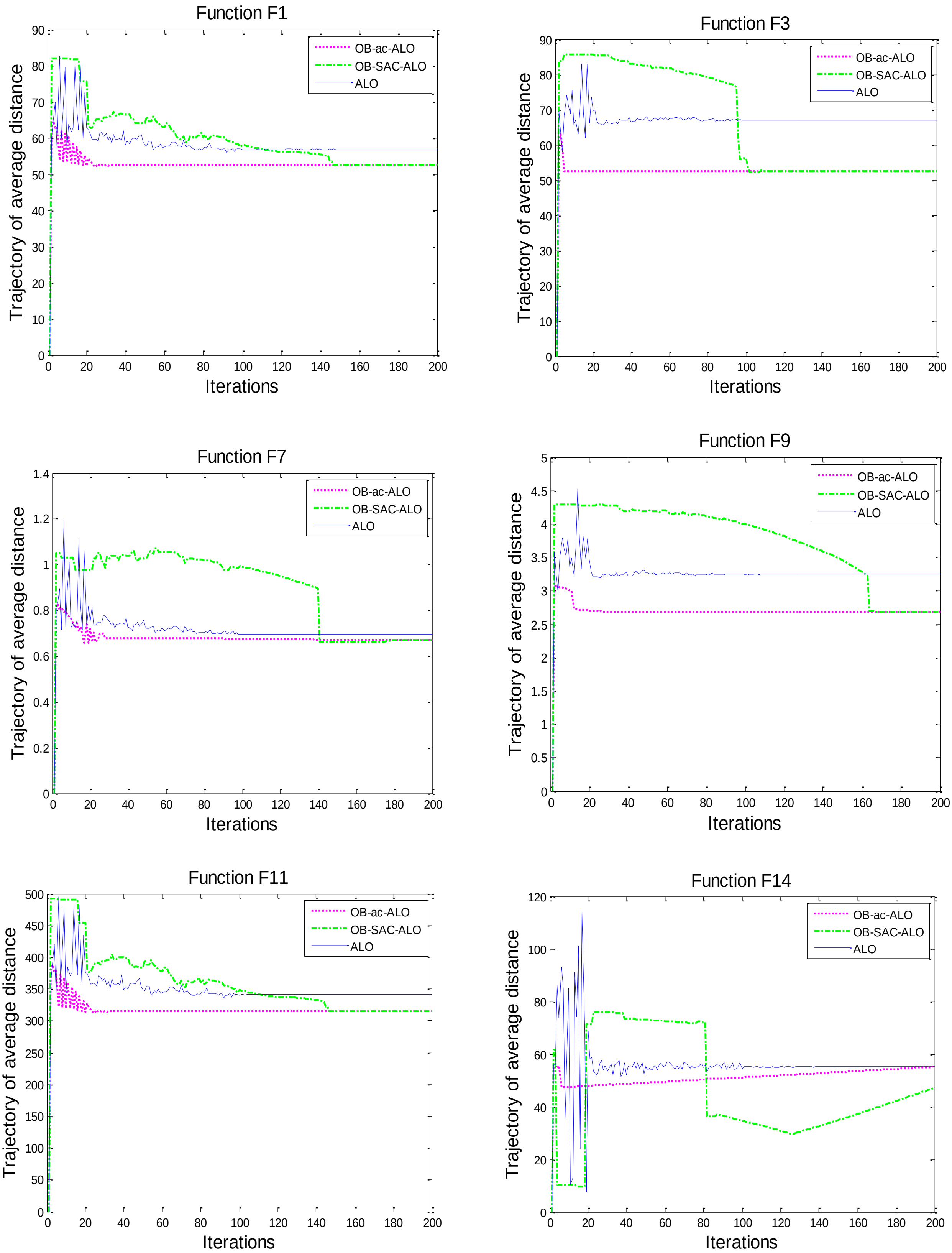


Figure 3.6: Trajectory of average distance between search agents

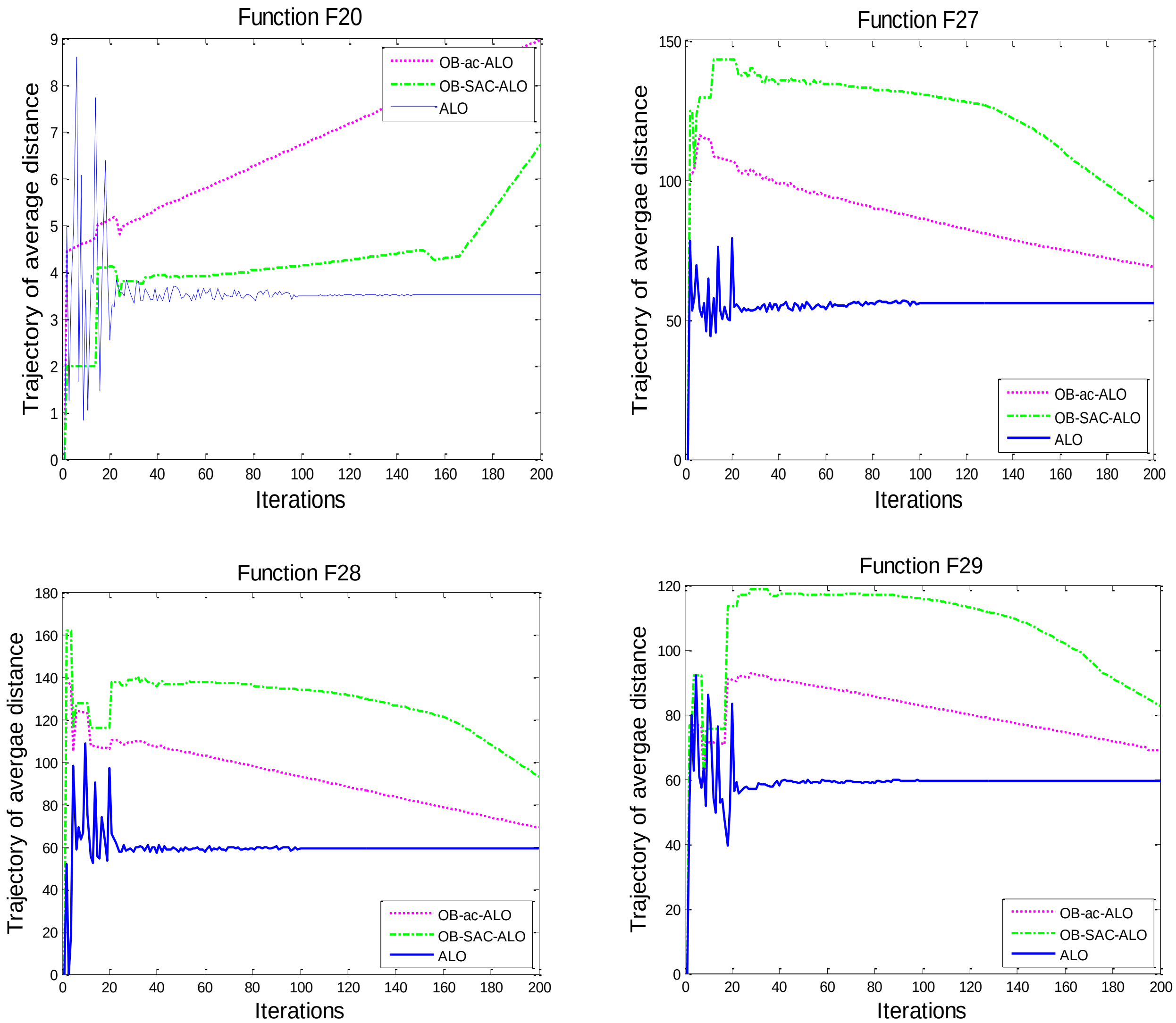


Figure 3.6: (Continued)

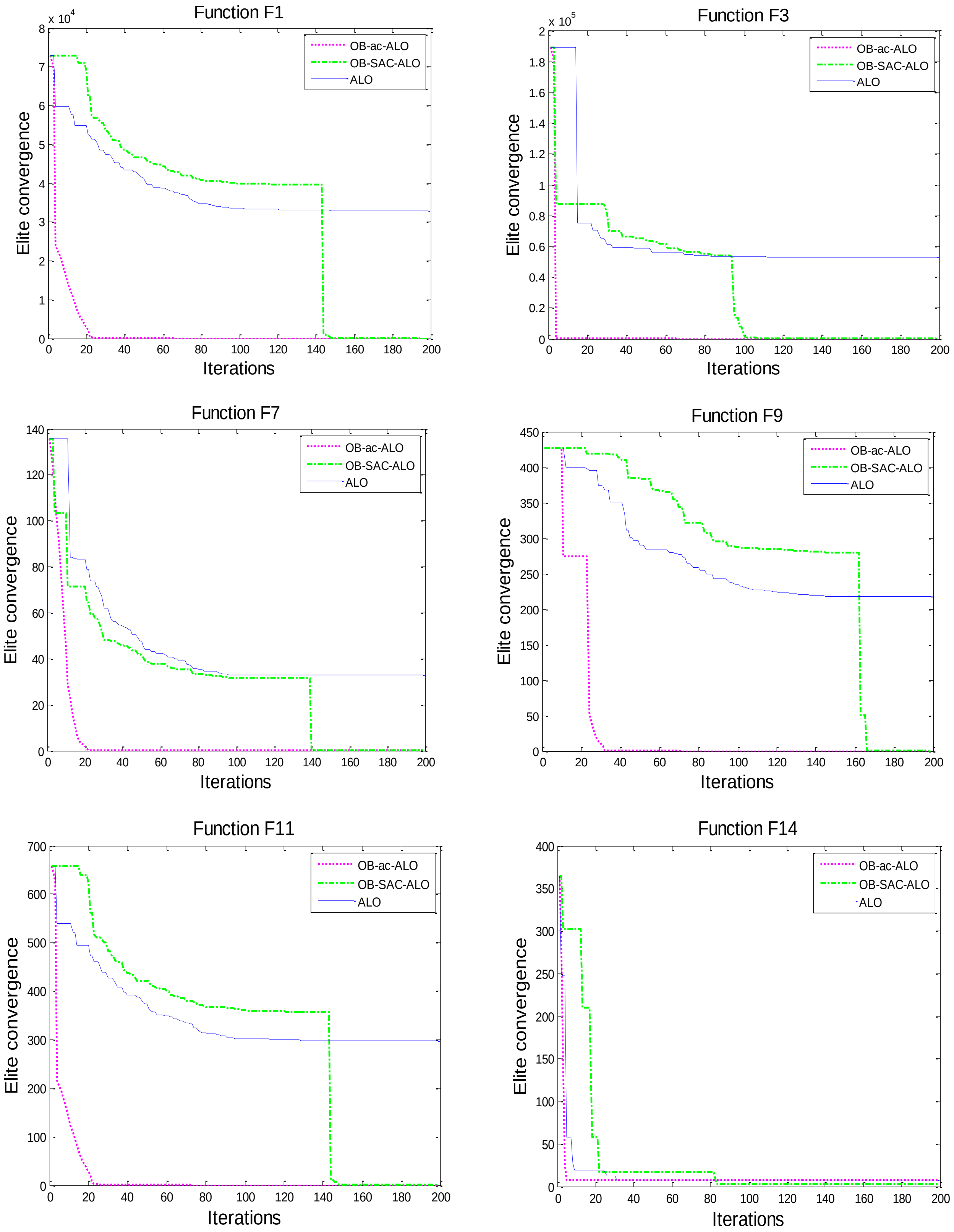


Figure 3.7: Convergence curve of elite(best)candidate solution

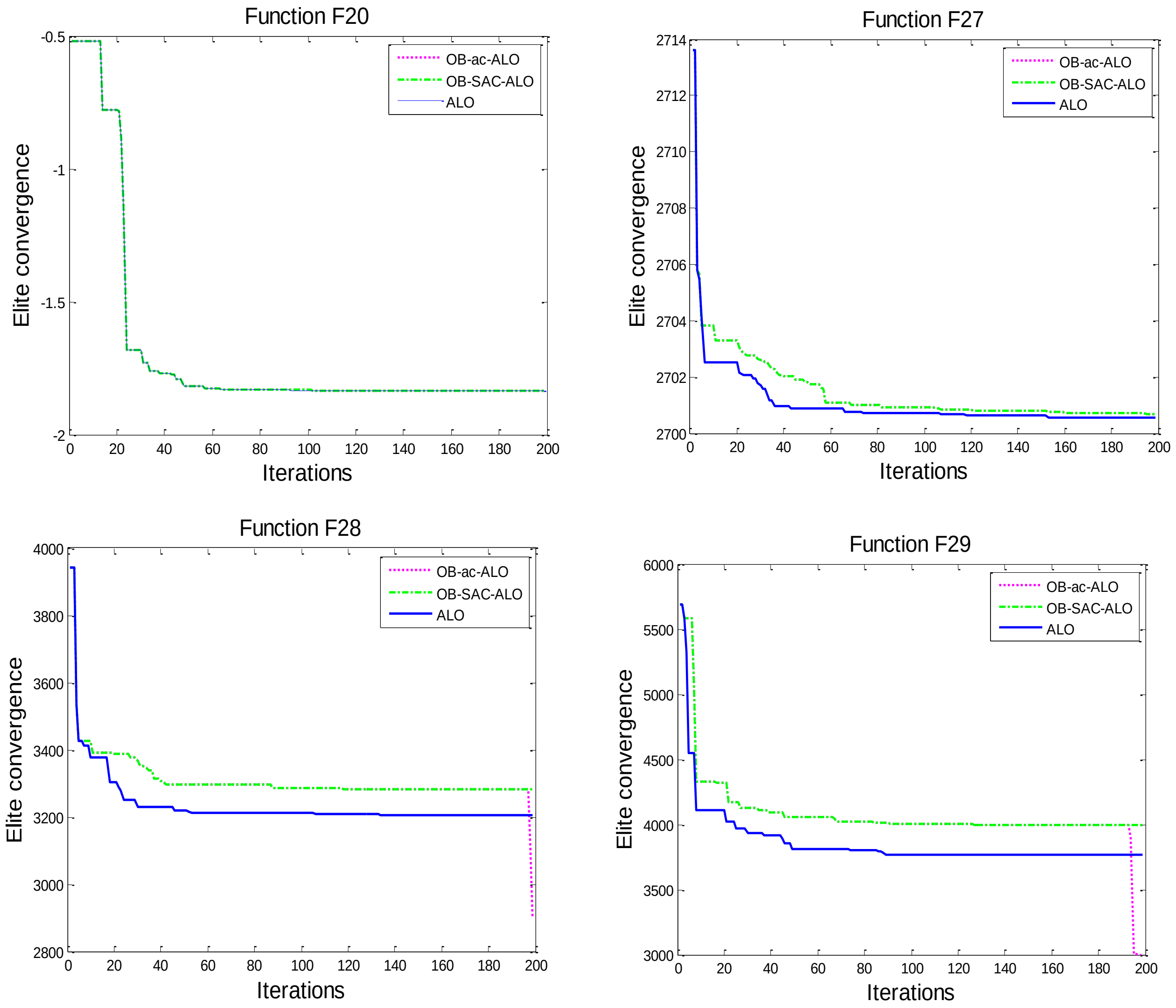


Figure 3.7: (Continued)

CHAPTER 4

Lévy Flight Distributed Opposition based Antlion Optimizer with Acceleration Coefficient for Continuous Optimization Problems

4.1 Introduction

The performance of any optimization algorithm can be improved by tuning applied stochastic operators which must be capable of exploring search region as much as possible in early phase of generation and exploitation to increase the convergence at later phase of generation. In ALO, the exploration of the algorithm can be enhanced by manipulating the random walk which is responsible to explore the search region. In the same way, convergence can be accelerated by good exploitation with appropriate acceleration parameter. This chapter proposes an extended modified opposition based ALO which tries to utilize the inclusion of three operators: (1) Lévy Flight distribution based random walk (2) Opposition based Learning around the best candidate solution (3) Acceleration parameter hybridized with OBL mechanism. This modified algorithm is named as Opposition based Antlion Optimizer using Lévy Flight distribution (OB-LF-ALO).

The classical ALO algorithm suffers from stagnation to local optima and requires diversified exploration with appropriate blending of exploitation. In this chapter, the extended proposed variant of classical ALO tries to expand stability between diversification and intensification during process of evolution. The performance of newly developed variant is evaluated using same set of benchmark test functions used in chapter 2 and 3 as well given in Appendix I. The impact of random numbers generated using lévy flight and updated population after employing opposition based learning during evolution is analyzed using certain metrics such as trajectories, elite convergence curve, and average of absolute distance between search agents before and after improving the algorithm. A non-parametric Wilcoxon ranksum test is used to exhibit its statistical significance. Computational complexity of proposed optimization algorithms is also derived and compared with classical ALO.

Partial contents of this chapter has been published as:

- **Dinkar, S. K., & Deep, K. (2018). An efficient opposition based Lévy Flight Antlion Optimizer for optimization problems. Journal of Computational Science,29,119-141.(SCIE,IF-1.925)**

This chapter is organized as follows: Section 4.2 describes related literature and motivation. Section 4.3 represents the proposed OB-LF-ALO algorithm including conceptual material about employed strategies. Section 4.4 depicts the description of benchmark test functions and experiment setting. Section 4.5 shows the computational results comparison and discussion as per the characteristics of benchmark functions. Section 4.6 depicts convergence and other analysis using various metrics to authenticate newly developed OB-LF-ALO and classical ALO. Afterwards, Section 4.7 gives concluding remarks.

4.2 Motivation

Though the classical ALO shows significant diversity of the search region still there is a scope of improvement in the basic search pattern by exploring the unvisited areas of the region and able to diminish the convergence errors. Lévy Flight (LF) is an efficient way to model the random walks of ants instead of using uniform distribution. LF has random initiated step length preferred to come out as probability distribution of power law distribution [113]. As per the research work published in Nature, it has been observed that the hunting pattern of prey-predator rely on the optimal walk based on Lévy flight distribution (LF) [114-115]. Later on, the concept of LF was employed to various optimization algorithms such as cuckoo search (CS) for generating new candidate solutions [116]. FA [117], ABC [104] and GWO [118] optimization algorithms were also enhanced using LF distribution. Some more work were introduced to enhance the performance of the metaheuristics algorithms. The LF distribution is employed to determine improved candidate solutions while improving the exploration of the search region in cuckoo search [116,119]. The concept of LF is also employed to improve the performance of basic PSO [120-121].

The second strategy applied in this work is opposition based learning (OB) model as utilized in chapter 2 and 3 for convergence acceleration. This model approximates the opposite candidate solutions concurrently along with normal solutions [98]. The randomization process does not guarantee to cover each and every candidate solution of the region. Determining the opposite candidate solutions is better to approximate as mathematically, the opposite numbers lie closer to the best solution [105]. The performance of differential evolution is also enhanced by employing OBL mechanism to the shuffled differential evolution [101]. In [106], opposition based learning (OBL) is implemented in shuffled frog leaping for parameter identification.

Third strategy is used in conjunction with OBL mechanism to enhance the acceleration and to adjust the exploration and exploitation. The generated random solution after applying LF distribution and OBL mechanism may exhibit abrupt behaviour. This conduct may be adjusted using a parameter known as varying acceleration cofficient (ac). This parameter is applied in chapter 3 also and proved to be an efficient way to accelerate the exploitation after exploration and accelerate the convergence speed.

4.3 Proposed Opposition based Lévy Flight Antlion Optimizer (OB-LF-ALO)

This section depicts the new developed variant to improve the performance of classical ALO. Though the exploration in classical ALO is well ensured by selection of antlion and ant's random walks around the antlion. The exploitation is achieved by adaptive shrinking of the trap built by antlion in each generation. It is better idea to enhance the coverage of the unexplored region of the search space so that unvisited candidate solutions may also be explored. The candidate solutions in classical ALO seem to be entrapped in local optima specifically in later generations. Increased exploration of the search region ensures to avoid local optima during evolution process.

Lévy Flight is proved to be efficient distribution based on random step lengths to diversify the search region effectively. It can alleviate the searching efficiency of the candidate solutions even at the deeper areas of the search region and tends to search global optima. In classical ALO the random walk is implemented using uniform distribution whereas in the proposed method, this uniform distribution is replaced with LF distribution. The second strategy opposition based learning with acceleration coefficient is applied to accelerate convergence rate. The elite (best) antlion is determined in each generation about which opposition strategy is employed to generate the new opposite candidate solution as new population for next generation. This integrated approach promises enhanced performance in terms of exploration by employing lévy flight distribution and convergence acceleration using opposition based strategy.

4.3.1 Lévy Flight Distribution

The distribution based on lévy flight (LF) is a kind of random walk which generate random numbers in principle on the basis of random steps and jumps determined by step length [122-123]. The LF distribution can be defined as:

$$L\acute{e}vy(\gamma) \sim u = t^{-\gamma},\ 1 < \gamma \leq 3 \tag{4.1}$$

The step length in LF distribution based random walk is heavy tailed probability distribution. The Lévy random number can be determined as:

$$Lévy(\gamma) \sim \frac{\varphi \times \mu}{|\alpha|^{\gamma}}, \quad (4.2)$$

Here μ and α both represent normal distribution. φ is defined as follows:

$$\varphi = \left[\frac{ℸ(1+\gamma) \times \sin(\pi \times \gamma/2)}{ℸ\left(\left(\frac{1+\gamma}{2}\right) \times \gamma \times 2^{\frac{\gamma-1}{2}}\right)} \right]^{1/\gamma}, \quad (4.3)$$

where ℸ is a standard gamma function, γ is taken to be 1.5.

Here $j = \frac{\mu}{|\alpha|^{\gamma}}$ defines the step length of the random distribution for $|j| \geq |j_0|$ and j_0 is the smallest step length [124].

The use of LF distributed random numbers in place of uniform random numbers improves the local search ability of global optima of the proposed algorithm. It also reduces the chances of the algorithm to be trapped in local optima due to its diverse capability of exploring the search region. This is imperative from the discussion that LF distribution guarantees the enhanced exploration of candidate solutions and better approximate the global optima than the original algorithm.

The proposed algorithm is employed with a step size J generated using lévy flight distribution as follows:

$$J(it) = 0.001 \times j(it) \quad (4.4)$$

where it is the current iteration.

Due to the unpredictable jumps of LF distribution, the new candidate solutions may sometime jump out of the search region. Thus the size may be chosen in such a way that the exploitation of the search region locally must be performed in efficient manner [125].

Hence the LF based random walk can be described as shown in eq. (4.5).

$$rw(S_{A,n}^{d}) = [cumsum(2 * J(it_1) - 1), cumsum(2 * J(it_2) - 1), \ldots, cumsum(2 * J(it_{max}) - 1) \quad (4.5)$$

Where J is LF distribution as defined in eqs. (4.1), (4.2), (4.3) and (4.4).

The impact of applying LF distribution as to enhance the exploration capability of the proposed algorithm can be visualized in Figure 4.1. The figure shows uniformly generated and LF distributed generated random population for size 200. It is apparent from the Figure 4.1 that the inclusion of LF distribution visits more search area as compared to the uniform distribution.

So this strategy ensures to expand the exploration of search region and useful to jump out of the local solutions while tending to identify the global optima.

4.3.2 Opposition Based Learning (OBL) Model

Commonly, Nature Inspire Optimization Techniques obey the black box functionality. In these algorithms, the candidate solutions are initialized randomly and then search the available search region effectively to improve the quality of candidate solutions so that the optimal solution is obtained until some termination criteria is reached. Utilization of this strategy guarantees the prominent acceleration in convergence and the definition of opposite numbers is defined in Section 2.3.2 of Chapter 2.

4.3.3 Acceleration Coefficient (ac)

Acceleration parameter e_{ac} as defined in chapter 3 is used to improve evolutionary process which is decreased adaptively as the number of iterations increases. This parameter e_{ac} is combined with OBL mechanism as follows:

$$S_{A,i}(it) = e_{ac} \times (LB_i + UB_i - S_{elite}(it)) \qquad (4.6)$$

4.3.4 Proposed OB-LF-ALO algorithm

The described strategies are applied to define proposed OB-LF-ALO algorithm in two phases: at first step the uniform distributed random numbers are replaced with random numbers generated using LF distribution as shown in eq. (4.5). This updation ensures the exploration augmentation of the search region. After improving random walk, new positions of ants are determined and compared with elite to find out whether elite is updated or not.

In second step: This process of evolution follows the employment of OBL around the updated elite (best antlion). Then it is combined with acceleration factor to balance evolutionary process. It results in improved position of candidate solution and fitness is determined for each ant. After evaluation, the fitness value of elite is compared with every ant's fitness and ant with better fitness value becomes elite antlion which is preserved as new elite and carried to the next generation. These two strategies employed in two steps promise to span more search area and acceleration in convergence. The computational steps of developed algorithm are described in Table 4.1 respectively.

4.4 Experimental Setup

Benchmark Test Functions

The same set of benchmarks functions as utilized in previous chapters to evaluate the performance of proposed variant is used. It is depicted depicted in Appendix I.

Experimental Setup and Parameter Setting

The similar setting of parameter is chosen as described in section 3.4 of chapter 3. The proposed algorithm OB-LF-ALO is compared with classical ALO to test the authority of this algorithm. The results are obtained on 10 dimension with 500 iterations and 30 dimension with 1000 iterations (stopping criteria) respectively on 23 traditional test suite as given in Appendix I. For both the algorithms, 30 independent runs are performed. For 8 composition functions, results are determined on 10 dimension and 1000 iteration taking 51 independent runs. "All the experiments have been performed on MATLAB R2014a on Intel(R) Core(TM) i5-7200 CPU @ 2.50GHz-2.71GHz with 8 GB RAM".

4.5 Results and Discussion

The proposed algorithm is developed for continuous optimization problems. So, it is worthy to compare it with the classical ALO which is also designed for continuous optimization problems. Table 4.2, Table 4.3, Table 4.4, Table 4.5, Table 4.6 and Table 4.7 show the results for different dimensions of various categories of problems.

Performance of exploitative behaviour for unimodal functions $F1$-$F7$

The results for 10 and 30 dimensions are shown in Table 4.2 and Table 4.3 respectively. It is clear from both the tables that for functions $F1$-$F7$, the proposed OB-LF-ALO shows better results than classical ALO. For functions $F1$-$F4$ the proposed OB-LF-ALO algorithm achieves global optima. For $F7$, the average of objective function values is significantly better for proposed OB-LF-ALO also. The standard deviation is steady throughout and establishes the stability and superiority of the proposed algorithm. It is evident from the results that the proposed applied strategy of lévy flight distribution and opposition based learning model with acceleration coefficient shows exploration in early stages of generation and exploitation in the later stages to search the exact global point efficiently for functions $F1$-$F7$.

Performance of explorative behaviour for multimodal functions $F8$-$F1$

The description of results on multimodal functions are depicted in Table 4.4 and Table 4.5 for 10 and 30 dimensions respectively. The average of objective function values for functions $F8$-$F13$ shows significant improvement for newly developed OB-LF-ALO in comparison to classical ALO. For functions $F9$ and $F11$ the proposed algorithm attains the best result and reach to global optima. This implies that the inclusion of lévy flight distribution enhances the diversity of search region and the proposed technique is capable of searching steeper point of the region. Hence this mechanism is efficient to determine a good solution than the uniform distribution applied in classical ALO.

Performance on fixed dimension multi modal test functions (Functions $F14$-$F23$)

These benchmark functions are similar to multimodal functions except that these functions are having fixed dimensions. Due to the varying nature of dimensions, these functions behave differently than the other multimodal functions. The results for these functions are depicted in Table 4.6. It can be observed from the results given in table that both the algorithms show same behaviour for functions $F16$, $F17$, $F18$,$F22$ and $F23$ and obtained results are also same. OB-LF-ALO outperforms the classical ALO for function $F14$ whereas the OB-LF-ALO show slight superiority over classical ALO for function $F15$ as far as average objective function value is concerned. For function $F19$, $F20$ and $F21$, the OB-LF-ALO outperforms classical ALO in terms of average function Value. The obtained values of standard deviation show remain steady and authorizes the improved performance of OB-LF-ALO.

Comparison of results on Composition Functions ($F24 - F31$)

Results of composition functions are depicted in Table 4.7. These functions define complex problem matching with the real world optimization applications and henceforth difficult to get the global optima. It is evident from the table that the proposed OB-LF-ALO outperforms the classical ALO for functions$F24, F27$ and $F28$ significantly. It is evident form the table that the standard deviation for these three functions is zero which exhibits the steady and stable performance of the proposed algorithm for these three functions. However, OB-LF-ALO is not able to beat classical ALO for remaining composition functions.

4.6 Analysis of Results

To verify the performance of OB-LF-ALO, following analysis are performed:

- 4.6.1 Analysis of Convergence Behaviour
- 4.6.2 Statistical significance- Wilcoxon ranksum test
- 4.6.3 Proposed Algorithm analysis
  - Search history behaviour
  - Trajectory behaviour
  - Average distance analysis between search agents

Elite convergence curve

4.6.4 Computational complexity

4.6.1 Analysis of Convergence Behaviour

The convergence behaviour of OB-LF-ALO method is analyzed with respect to classical ALO. A set of benchmark functions containing each type of functions are chosen and the corresponding convergence curves are drawn to verify the acceleration in convergence of new algorithm. On X-axis, the number of generations have been plotted against the average of best objective function (fitness) values performing 30 runs independently on log scale of Y-axis.

Figure 4.2 depicts the convergence behaviour of three unimodal functions $F1$,$F3$ and $F7$. It is clearly evident from the curves that the newly developed OB-LF-ALO algorithm shows convergence starting from initial generation. The curve in Figure 4.2 clearly shows the superior convergence of OB-LF-ALO over classical ALO. This behaviour guarantees that the inclusion of LF distribution enhances the exploration of search region followed by opposition based model which is capable of approximating the opposite candidate solutions nearer to the global optima. Since these functions contain a single optima, thus this convergence behaviour of proposed algorithm ensures to converge to a global optima [110].

Figure 4.3 and Figure 4.4 represent the converging capability of three multimodal functions ($F9$,$F10$,$F11$) and non-scalable multimodal functions ($F14$,$F20$). It can be clearly observed from the Figure 4.3 that the proposed OB-LF-ALO exhibits very steep acceleration in convergence from early generations and exhibits superior and significant acceleration in convergence rate for all three multimodal functions. The proposed algorithm exhibits significant convergence rate for fixed multimodal functions $F14$ and $F20$ which shows that OB-LF-ALO outperforms the classical ALO. Figure 4.5 exhibits the convergence behaviour for composition functions $F28$ and $F29$. It is clearly evident that the classical ALO shows better convergence in early generations for both the functions but OB-LF-ALO depicts sharp and steep convergence at later generations and performs better than the classical ALO for both functions. The multimodal and composition functions contain many optima and prone to be entrapped in local optima thus restricted to reach global optima. But the random jumps in LF distribution ensures to visit the unexplored region of the search space and tries to avoid local optima. This philosophy helps the proposed algorithm to reach near to global optima with a very high rate of convergence acceleration due to opposition based learning model followed by LF distribution.

4.6.2 Statistical significance- Wilcoxon Ranksum Test

The applied non-parametric Wilcoxon ranksum test investigates the statistical significance of proposed OB-C-ALO with respect to classical ALO. This test analyzes the distribution of sampled data drawn from two algorithms in pair. The obtained result in terms of p-values authorizes the data drawn from one algorithm is statistically significant than the data drawn from other algorithm.

The confidence level is taken as 0.95. The sampled data drawn for 30 dimensions of 23 state-of-the-art and 10 dimensions for 8 composition problems from two algorithms as used in pair from 30 independent runs to reject the null hypothesis. This hypothesis checks that the data drawn of continuous distribution from two algorithms with equal median against the alternative that actually they are not. The obtained results in terms of p-values are shown in Table 4.8. This table depicts the comparison of statistical significance for proposed OB-LF-ALO with classical ALO. The '+' denotes that the statistical difference is significant at 0.05 level of significance, '-' indicates that there is no statistical difference and '=' depicts that the sampled data drawn from the pair of algorithm are similar and comparison is not possible.

4.6.3 Analysis of Proposed Variant

The verification of the performance of OB-LF-ALO is authorized by employing few analysis metrics to investigate its superiority over the classical ALO. The aim of performing this analysis is to establish the improvement of acceleration in convergence rate, exploration capability and exploitation ability of OB-LF-ALO. For this analysis, four search agents are employed by choosing few unimodal, multimodal, on-scalable fixed dimension multimodal and composition functions for maximum of 200 iterations. Analysis is performed using given below metrics:

Search history behaviour

This analysis is performed to verify the searching behaviour of the random population before and after employing the proposed strategies. Figure 4.6 demonstrates this behaviour for few benchmark suits by drawing the population over the contour of the respective functions. It is clearly evident that after applying LF distribution followed by opposition based learning, the search agents of proposed OB-LF-ALO are more accurate and efficient to search the promising region of the search space as compared to the classical ALO. This behaviour can be easily visualized by analyzing the search history of functions $F1, F3, F9, F11$ and $F14$. It is also visible that the proposed algorithm is significantly better to search the promising region at early stages and guarantees to accelerate the convergence rate.

Trajectory behaviour

This analysis presents the position of the elite antlion starting from the first iteration to the last iteration. It signifies the changes in the position of elite antlion after employing the proposed mechanism of LF distribution followed by opposition based learning with acceleration coefficient in proposed OB-LF-ALO as the number of iterations increased. Its impact on updating the position is then compared to classical ALO. The graphical representation is shown in Figure 4.7.

A set of ten functions including 3 unimodal functions($F1$,$F3$,$F7$),3 multimodal functions ($F9$,$F10$,$F11$),2 fixed dimensions functions ($F14$,$F20$) and 2 composition functions($F28, F29$) have been chosen to analyze antlion's trajectories. It is evident from Figure 4.7 that the random jump of LF distribution has a great impact on antlion's trajectory which enhances the exploration behaviour to come out of local optima at early stage of generations. Then the opposition based learning around elite antlion finds the corresponding opposite positions and responsible for a stationary behaviour of trajectories around the global optimum position. This behaviour of antlion's position promises the enhanced exploration due to LF distribution and capable of avoiding entrapment in local optima during initial generations and then accelerate the convergence due to employing acceleration coefficient with opposition based learning while ensuring to converge a point. The behaviour of obtained positions of antlion promises that the algorithm searches prominent regions of the space as compared to classical ALO. It also ensures the enhancement in exploration of search region and guarantees to converge at optima while avoiding stagnation to local optima [110].

Average distance analysis between search agents

This analysis is performed by finding the average (Euclidean) distance between the original and updated positions before and after applying the improvement strategies of the search agents. Euclidean distance between two points $X(x_1, x_2, \dots, x_D)$ and $Y(y_1, y_2, \dots, y_D)$ in a D-dimensional search space can be defined as also described in chapter 2 and chapter 3:

$$d(X,Y)=||X,Y||=\sqrt{\sum_{i=1}^{D}(x_i - y_i)^2} \quad (4.7)$$

The eq.(4.9) shown above can be simplified for a one dimensional search space:

$$d(X, Y) = \left||X, Y|\right| = |x - y| \quad (4.8)$$

The average distance is determined using above equation during each iteration for classical ALO and proposed OB-LF-ALO after applying improvement strategies. The maximum iteration to be chosen are 200 of first dimension of first variable.

This analysis is useful to determine the impact of employed LF distribution followed by opposition based learning with acceleration parameter. The determined distance is capable of verifying candidate solutions which are possibly nearer to global optima. For this purpose same ten functions including three unimodal ($F1, F3, F7$), three multimodal ($F9, F10, F11$),two fixed dimension ($F14, F20$) and two composition ($F28$,$F29$) functions are chosen.

It is clearly evident from the Figure 4.8 that the proposed OB-LF-ALO technique is approximating the closer candidate solution as compared to classical ALO and tends to find global optima with improved convergence speed than the original algorithm. The trajectory for unimodal functions and multimodal functions clearly established the better approximation of the candidate solutions near to the global optimum and guarantees to accelerate the convergence speed towards to global solution. But for one fixed dimension function($F20$) shows slightly different behaviour in which the candidate solutions are not approximating at closer positions of global solution though the proposed OB-LF-ALO determines better fitness value than the original algorithm as shown in result table. The proposed OB-LF-ALO also exhibits trajectories nearer to global optima than classical ALO for both $F28$ and $F29$ functions. Overall analysis shows that the proposed algorithm guarantees acceleration in convergence.

Elite convergence curve

Another metric to verify the performance of OB-LF-ALO is convergence of elite antlion. This convergence rate is shown as a curve of best antlion (elite) of first variable. The same ten functions as used in the previous section have been analyzed for elite convergence curve and depicted in Figure 4.9.

It is clearly observed from Figure 4.9 that the fitness drops significantly even at initial iterations which signifies that the antlions' fitness of proposed OB-LF-ALO converges during the course of iterations as compared to classical ALO. However, the convergence behaviour is unusual for function $F28$ and $F29$ where the curves show steady behaviour at earlier generations but exhibit steep convergence and abrupt jumping behaviour at later generations and prove to have better convergence than classical ALO. This analysis validates high acceleration in convergence with the increase in number of iterations and tries efficiently to search global optima with more precision. The employability of LF distribution ensures the enhanced exploration and avoidance of local optima stagnation. The opposition based learning (OB) is capable of approximating the nearer solution to the best antlion (elite) and ensures the acceleration in convergence towards a point.

4.6.4 Computational Complexity

Computational complexity plays a vital role in development of any algorithm. A better technique to modify an existing algorithm should not increase the computational complexity of proposed algorithm. In proposed OB-LF-ALO, although three new strategies are employed to improve the performance of algorithm, yet it is being taken care of that there should not any enhancement in computational complexity.

It is ensured not to include any extra loop and the evaluation of objective function is performed after applying all the strategies to determine the updated position. This theory establishes that there is no change in asymptotic computational complexity and it remains same as of classical ALO as described in chapter 2 and chapter 3. Thus, it is observed as $O(it_{max} * D * N) * O\ (f(X))$ where it_{max} maximum number of iteration, D represents dimensions, N is population size and $f(X)$ defined as the objective function for worst case scenario.

4.7 Conclusion

This chapter aims to extend and improve the ability of classical ALO by firstly, enhancing the exploration of search region efficiently using lévy flight distributed random numbers in place of uniform random numbers. Secondly, opposition based model and thirdly acceleration parameter in conjunction with OBL mechanism applied to the elite antlion to approximate the closer candidate solution to the global optima which enhances the convergence acceleration rate. These three approaches are integrated to propose OB-LF-ALO.

The performance of the proposed algorithm is verified over 31 benchmark functions of variety of characteristics including unimodal, multimodal, fixed dimensional multimodal and composition functions in order to validate the exploitation, exploration and ability to avoid local optima stagnation of the proposed algorithm. The OB-LF-ALO algorithm is extensively analyzed using convergence curve and other useful metrics and compared with classical ALO. The inclusion of lévy flight distributed random numbers in proposed OB-LF-ALO algorithm has a potential to enhance the exploration of the unvisited region of the search space which can be observed from the convergence curves. After that, integration of opposition based model around the elite antlion is able to approximate the closer candidate solutions and thus proposed OB-LF-ALO is able to accelerate the convergence rate efficiently. The algorithm analysis in terms of various metrics like search history analysis, trajectories analysis, elite convergence curve and average distance between search agents digs out the efficiency of proposed algorithm in comparison to classical ALO. The search history is useful to indicate the search behaviour of search agents for showing exploration of promising region of the search space before and after applying the proposed strategies. Elitism approach demonstrates the significant drop even at initial generation as verified from convergence curve. Local optima stagnation is efficiently avoided in proposed OB-LF-ALO as it is evident that for nine functions, the proposed algorithm is capable to reach global optima.

Though the applied strategies are able to enhance the overall performance of classical ALO. But heavy tailed probability distribution of step length in random walk of lévy flight causes enormous rise in computational time of proposed OB-LF-ALO as compared to classical ALO for high dimension problems.

Table 4.1: Computational Steps of Proposed OB-LF-ALO algorithm

Input: *Population Size N, Maximum iteration it_{max},lower bound L, Upper bound U and dimension D*

Output: *The best candidate solution S_{elite}*

Randomly initialize the initial population N of ants and antlions
Determine the objective(fitness) function value of antlions
Find out the best(with min fitness) antlion as the elite S_{elite}
Initialize iteration no. it_{curr}=2
    while $(\boldsymbol{it_{curr}} \leq \boldsymbol{it_{max}})$
        for every ant $(\boldsymbol{i = 1, 2, 3, \ldots, N})$
            Find an antlion S_{sel} using Roulette wheel
            Modify lower L and upper U boundaries with equations Eqs. $L^d(it) = \frac{L^d(it)}{I}$ *and* $U^d(it) = \frac{U^d(it)}{I}$*)*
            for every dimension $(\boldsymbol{j = 1, 2, 3, \ldots, D})$
                Perform random walk $rw_A(it)$ around S_{sel}
                and $rw_E(it)$ around S_{elite} using LF distribution with Eq. $rw(S_{A,n}^d)$=[$cumsum(2* J(it_1)$- 1),$cumsum(2* J(it_2)$-1),…,$cumsum(2*J(it_{max})$-1)
                Normalize random walk using Eqs $S_{A,n}^d(it) = \frac{(S_{A,n}^d(it) - \min rw(S_{A,n}^d))(U^d(it) - L^d(it))}{\max rw(S_{A,n}^d) - \min rw(S_{A,n}^d)} + L^d(it)$*and*
$rw(S_{A,n}^d)$=[$cumsum(2* J(it_1)$- 1),$cumsum(2* J(it_2)$-1),…,$cumsum(2*J(it_{max})$-1)
            end for
            Modify the position of ant using Eq. $S_{A,n}^d(it) = \frac{r_{wA}(it) + r_{wE}(it)}{2}$
        end for
        Modify position of ant by applying acceleration parameter e_{ac} integrated
        With opposition based learning(OBL) model using
        Eq. $S_{A,i}(it) = e_{ac} \times (LB_i + UB_i - S_{elite}(it))$
        for every ant $(\boldsymbol{i = 1, 2, 3, \ldots, N})$
            Determine the fitness of all ants
        end for
        Substitute an antlion with its respective ant if it becomes fitter using Eq.
        $S_{AL,j}(it) = S_{A,i}(it) \text{ if } f(S_{A,i}(it)) < f(S_{AL,j}(it))$
        Modify S_{elite} if an antlion becomes fitter than the elite
        Increment iteration i.e. it_{curr}=it_{curr}+1
    end while
Return elite

Table 4.2: Average, Standard Deviation, Minimum, and Maximum of objective function values of unimodal functions (10D)

Function	***Methods***	***Ave.***	***Stan. Dev.***	***Min.***	***Max.***
$F1$	**OB-LF-ALO**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	ALO	7.870E-09	5.490E-09	2.390E-09	**2.580E-08**
$F2$	**OB-LF-ALO**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	ALO	4.860E-01	8.850E-01	1.550E-05	**2.800E+00**
$F3$	**OB-LF-ALO**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	ALO	8.310E-02	1.850E-01	1.360E-04	**8.750E-01**
$F4$	**OB-LF-ALO**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	ALO	3.180E-03	6.160E-03	1.020E-04	**3.290E-02**
$F5$	**OB-LF-ALO**	**6.290E-02**	**1.070E-01**	**2.920E-05**	**4.030E-01**
	ALO	6.810E+01	1.990E+02	1.020E-04	**1.070E+03**
$F6$	OB-LF-ALO	1.200E-03	1.290E-03	5.900E-06	5.100E-03
	ALO	**8.390E-09**	**5.220E-09**	**2.150E-09**	2.220E-08
$F7$	**OB-LF-ALO**	**3.110E-04**	**4.150E-04**	**1.670E-05**	**1.990E-03**
	ALO	2.210E-02	1.230E-02	1.810E-03	**5.630E-02**

Table 4.3: Average, Standard Deviation, Minimum, and Maximum of objective function values of unimodal functions (30D)

Function	***Methods***	***Ave.***	***Stan. Dev.***	***Min.***	***Max.***
$F1$	**OB-LF-ALO**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	ALO	9.450E-06	5.540E-06	1.720E-06	2.110E-05
$F2$	**OB-LF-ALO**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	ALO	3.650E+01	5.170E+01	2.090E-02	1.340E+02
$F3$	**OB-LF-ALO**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	ALO	9.550E+02	5.080E+02	2.470E+02	2.160E+03
$F4$	**OB-LF-ALO**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	ALO	1.210E+01	3.650E+00	4.630E+00	1.960E+01
$F5$	**OB-LF-ALO**	**9.590E-03**	**1.280E-02**	**2.310E-04**	**4.740E-02**
	ALO	1.040E+02	3.120E+02	1.640E+01	1.740E+03
$F6$	OB-LF-ALO	5.860E-04	5.890E-04	3.820E-05	2.200E-03
	ALO	**9.490E-06**	**6.100E-06**	**1.320E-06**	**2.920E-05**
$F7$	**OB-LF-ALO**	**9.730E-05**	**8.690E-05**	**1.480E-06**	**3.090E-04**
	ALO	1.100E-01	3.570E-02	5.700E-02	2.250E-01

Table 4.4: Average, Standard Deviation, Minimum, and Maximum of objective function values of multimodal functions (10D)

Function	***Methods***	***Ave.***	***Stan. Dev.***	***Min.***	***Max.***
$F8$	**OB-LF-ALO**	**-1.340E+03**	**8.610E+00**	**-1.360E+03**	**-1.330E+03**
	ALO	-2.470E+03	4.260E+02	-3.730E+03	-1.920E+03
$F9$	**OB-LF-ALO**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	ALO	2.480E+01	1.080E+01	6.960E+00	**5.070E+01**
$F10$	**OB-LF-ALO**	**8.880E-16**	**4.010E-31**	**8.880E-16**	**8.880E-16**
	ALO	3.240E-01	5.540E-01	2.070E-05	**1.650E+00**
$F11$	**OB-LF-ALO**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	ALO	2.140E-01	8.720E-02	6.640E-02	**4.060E-01**
$F12$	**OB-LF-ALO**	**1.300E-03**	**2.250E-03**	**2.300E-09**	**9.130E-03**
	ALO	1.790E+00	1.670E+00	8.230E-09	**5.610E+00**
$F13$	OB-LF-ALO	5.720E-03	1.280E-02	1.580E-06	6.350E-02
	ALO	1.800E-03	4.940E-03	3.840E-09	**2.100E-02**

Table 4.5: Average, Standard Deviation, Minimum, and Maximum of objective function values of multimodal functions (30D)

Function	***Methods***	***Ave.***	***Stan. Dev.***	***Min.***	***Max.***
$F8$	**OB-LF-ALO**	**-1.080E+04**	**3.780E+03**	**-1.260E+04**	**-2.130E+03**
	ALO	-5.440E+03	4.190E+01	-5.540E+03	-5.420E+03
$F9$	**OB-LF-ALO**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	ALO	1.970E+00	8.390E-01	5.890E-04	3.520E+00
$F10$	**OB-LF-ALO**	**8.880E-16**	**4.010E-31**	**8.880E-16**	**8.880E-16**
	ALO	1.870E+00	6.900E-01	1.020E-03	3.090E+00
$F11$	**OB-LF-ALO**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	ALO	1.190E-02	1.260E-02	3.470E-04	3.860E-02
$F12$	**OB-LF-ALO**	**3.200E-05**	**8.410E-05**	**1.460E-08**	**4.540E-04**
	ALO	1.000E+01	4.340E+00	5.750E+00	2.330E+01
$F13$	**OB-LF-ALO**	**1.890E-04**	**3.900E-04**	**1.720E-07**	**1.940E-03**
	ALO	1.320E+00	3.150E+00	1.140E-05	1.050E+01

Table 4.6: Average, Standard Deviation, Minimum, and Maximum of objective function values of fixed dimensional multimodal functions

Function	*Methods*	*Ave.*	*Stan. Dev.*	*Min.*	*Max.*
$F14$	**OB-LF-ALO**	**1.300E+00**	**5.310E-01**	**9.980E-01**	**2.9800E+00**
	ALO	1.790E+00	1.340E+00	1.950E-08	4.9500E+00
$F15$	OB-LF-ALO	1.510E-03	3.570E-03	5.100E-04	2.040E-02
	ALO	4.180E-03	7.370E-03	5.550E-04	**2.040E-02**
$F16$	**OB-LF-ALO**	**-1.030E+00**	**6.7800E-16**	**-1.030E+00**	**-1.030E+00**
	ALO	-1.030E+00	6.780E-16	-1.030E+00	**-1.030E+00**
$F17$	**OB-LF-ALO**	**3.980E-01**	**1.6900E-16**	**3.980E-01**	**3.980E-01**
	ALO	3.980E-01	1.690E-16	3.980E-01	**3.980E-01**
$F18$	**OB-LF-ALO**	**3.000E+00**	**0.0000E+00**	**3.000E+00**	**3.000E+00**
	ALO	3.000E+00	0.000E+00	3.000E+00	**3.000E+00**
$F19$	**OB-LF-ALO**	**-7.790E+00**	**2.810E+00**	**-1.020E+01**	**-2.630E+00**
	ALO	-6.280E+00	2.930E+00	**-1.020E+01**	**-2.630E+00**
$F20$	**OB-LF-ALO**	**-8.520E+00**	**2.970E+00**	**-1.040E+01**	**-2.770E+00**
	ALO	-6.670E+00	3.420E+00	-1.040E+01	**-1.840E+00**
$F21$	**OB-LF-ALO**	**-7.100E+00**	**3.580E+00**	**-1.050E+01**	**-2.420E+00**
	ALO	-6.270E+00	3.660E+00	**-1.050E+01**	-1.860E+00
$F22$	**OB-LF-ALO**	**-3.863E+00**	**3.160E-15**	**-3.863E+00**	**-3.863E+00**
	ALO	-3.863E+00	3.160E-15	-3.863E+00	-3.863E+00
$F23$	**OB-LF-ALO**	**-3.322E+00**	**1.810E-15**	**-3.322E+00**	**-3.322E+00**
	ALO	-3.322E+00	1.810E-15	-3.322E+00	-3.322E+00

Table 4.7: Average, Standard Deviation, Minimum, and Maximum of objective function values of composition benchmark functions

Function	*Methods*	*Ave.*	*Stan. Dev.*	*Min.*	*Max.*
$F24$	**OB-LF-ALO**	**2.500E+03**	**0.000E+00**	**2.500E+03**	**2.500E+03**
	ALO	2.630E+03	1.275E-03	2.630E+03	2.630E+03
$F25$	OB-LF-ALO	2.600E+03	7.066E-01	2.596E+03	2.600E+03
	ALO	2.538E+03	1.477E+01	2.577E+03	2.512E+03
$F26$	OB-LF-ALO	2.700E+03	4.349E-01	2.697E+03	2.700E+03
	ALO	2.680E+03	2.430E+01	2.630E+03	2.700E+03
$F27$	OB-LF-ALO	2.705E+03	9.557E-02	2.705E+03	2.705E+03
	ALO	2.700E+03	1.090E-01	2.700E+03	2.700E+03
$F28$	**OB-LF-ALO**	**2.900E+03**	**0.000E+00**	**2.900E+03**	**2.900E+03**
	ALO	3.010E+03	1.440E+02	2.700E+03	**3.110E+03**
$F29$	**OB-LF-ALO**	**3.000E+03**	**0.000E+00**	**3.000E+03**	**3.000E+03**
	ALO	3.490E+03	2.920E+02	3.170E+03	3.930E+03
$F30$	OB-LF-ALO	4.224E+06	1.406E+06	3.100E+03	4.755E+06
	ALO	8.270E+05	1.520E+06	3.100E+03	3.670E+06
$F31$	OB-LF-ALO	5.623E+03	5.444E+02	5.406E+03	7.653E+03
	ALO	4.420E+03	4.140E+02	3.880E+03	6.170E+03

Table 4.8: P-values of Wilcoxon ranksum test

Function	OB-LF-ALO vs ALO	
	p-values	conclusion
$F1$	1.21E-12	+
$F2$	1.21E-12	+
$F3$	1.21E-12	+
$F4$	1.21E-12	+
$F5$	3.02E-11	+
$F6$	3.02E-11	-
$F7$	3.02E-11	+
$F8$	4.73E-06	+
$F9$	1.20E-12	+
$F10$	1.18E-12	+
$F11$	1.21E-12	+
$F12$	3.02E-11	+
$F13$	7.60E-07	+
$F14$	2.38E-02	+
$F15$	8.94E-01	-
$F16$	N/A	=
$F17$	N/A	=
$F18$	N/A	=
$F19$	1.73E-01	-
$F20$	1.69E-01	-
$F21$	7.62E-01	-
$F22$	N/A	=
$F23$	N/A	=
$F24$	1.36E-20	+
$F25$	3.84E-20	-
$F26$	2.21E-06	-
$F27$	3.30E-18	-
$F28$	1.78E-09	+
$F29$	1.39E-20	+
$F30$	2.04E-12	-
$F31$	5.28E-16	-
21/31(16 ‘+’ , 5 ‘=’)		

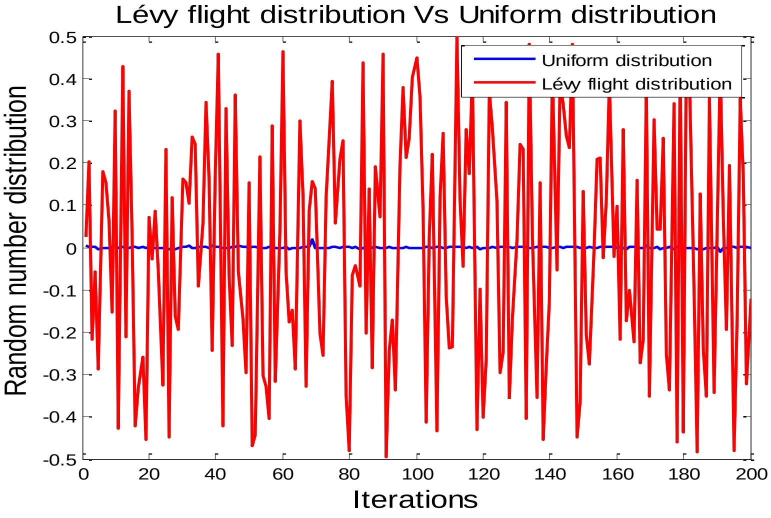


Figure 4.1: Demonstration of random population using lévy flight and uniform distribution

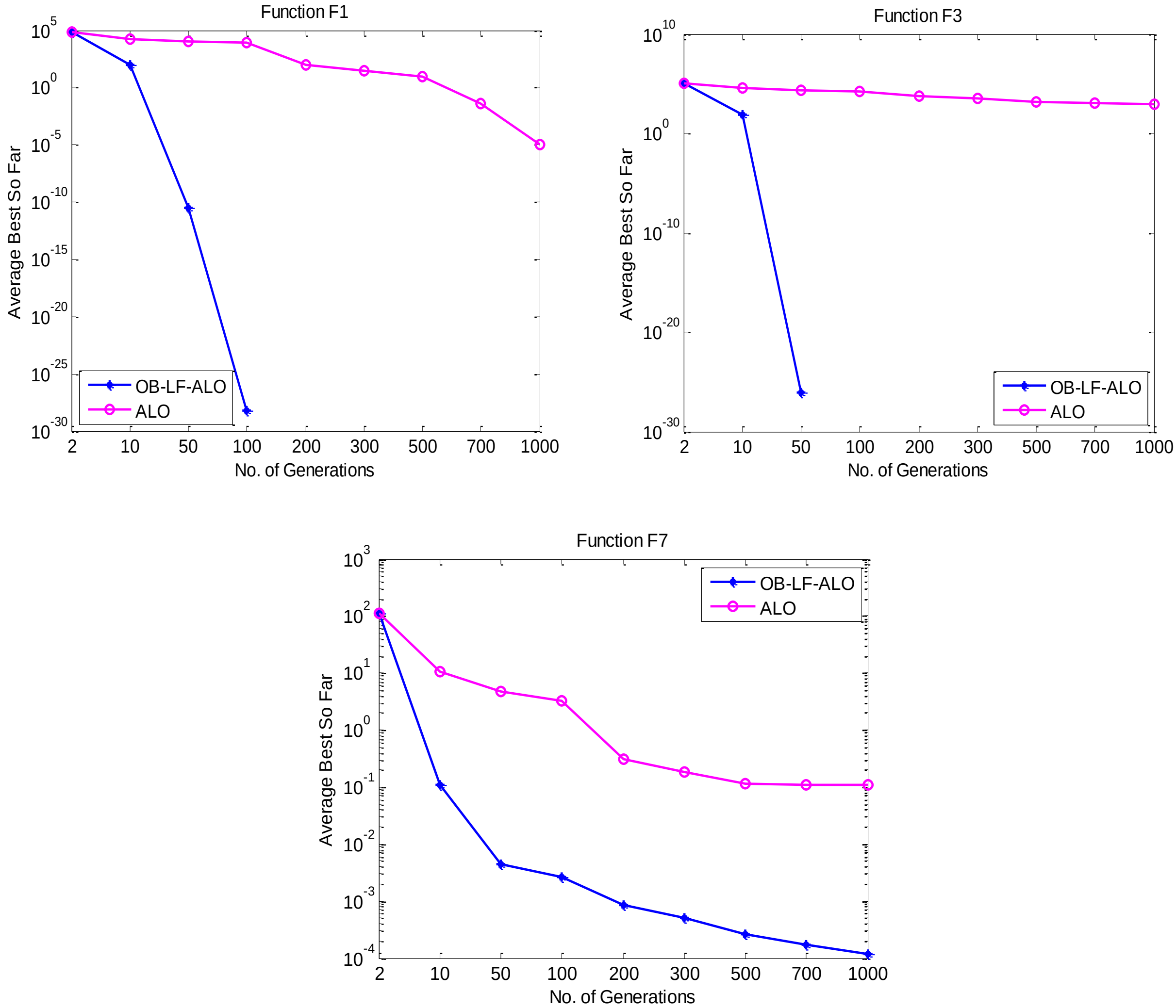


Figure 4.2**:** Convergence behaviour on three of the unimodal functions $F1$,$F3$ and $F7$

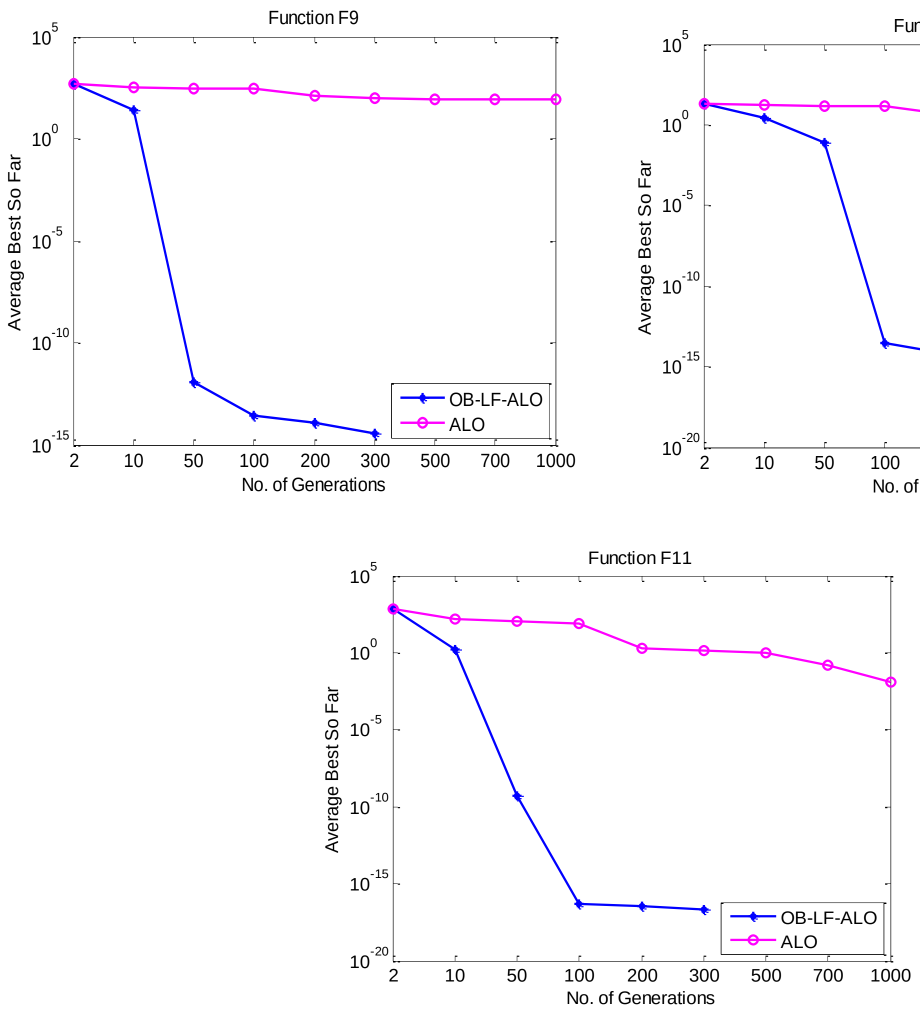


Figure 4.3**:** Convergence behaviour on three multimodal functions $F9$,$F10$ and $F11$.

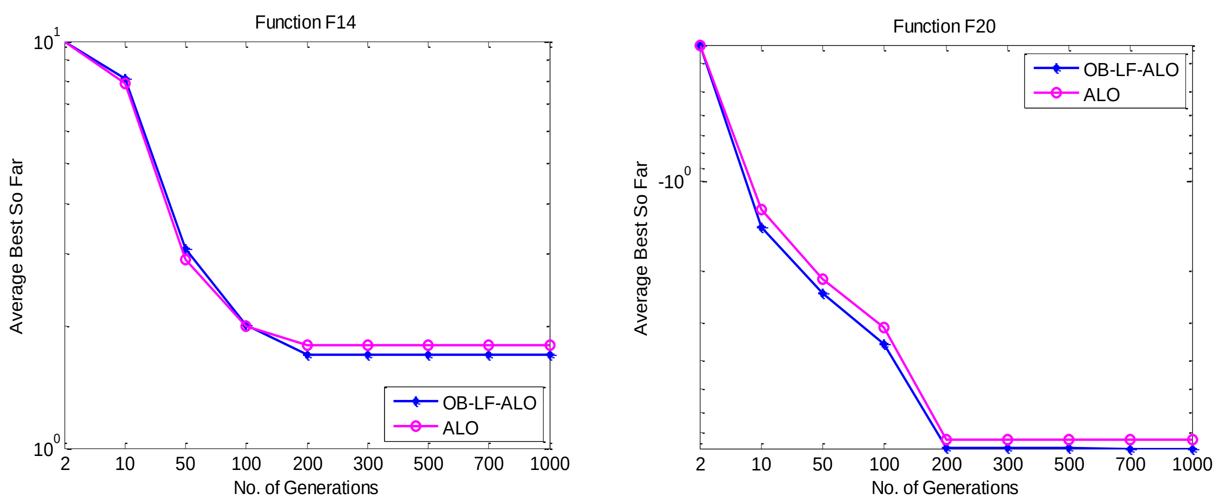


Figure 4.4**:** Convergence behaviour on two fixed dimension multimodal functions $F14$ and $F20$

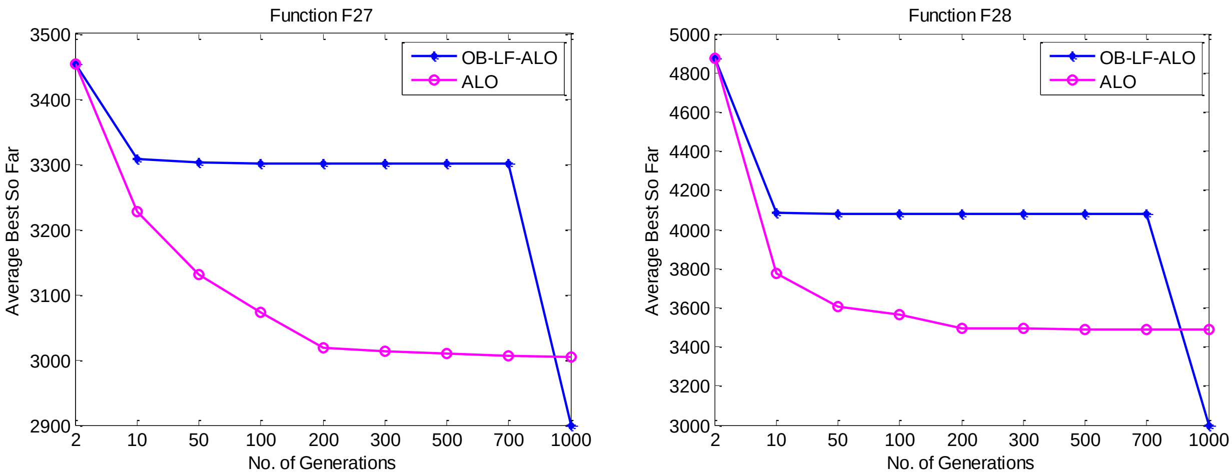


Figure 4.5 : Convergence behaviour on two composition functions $F27$ and $F28$

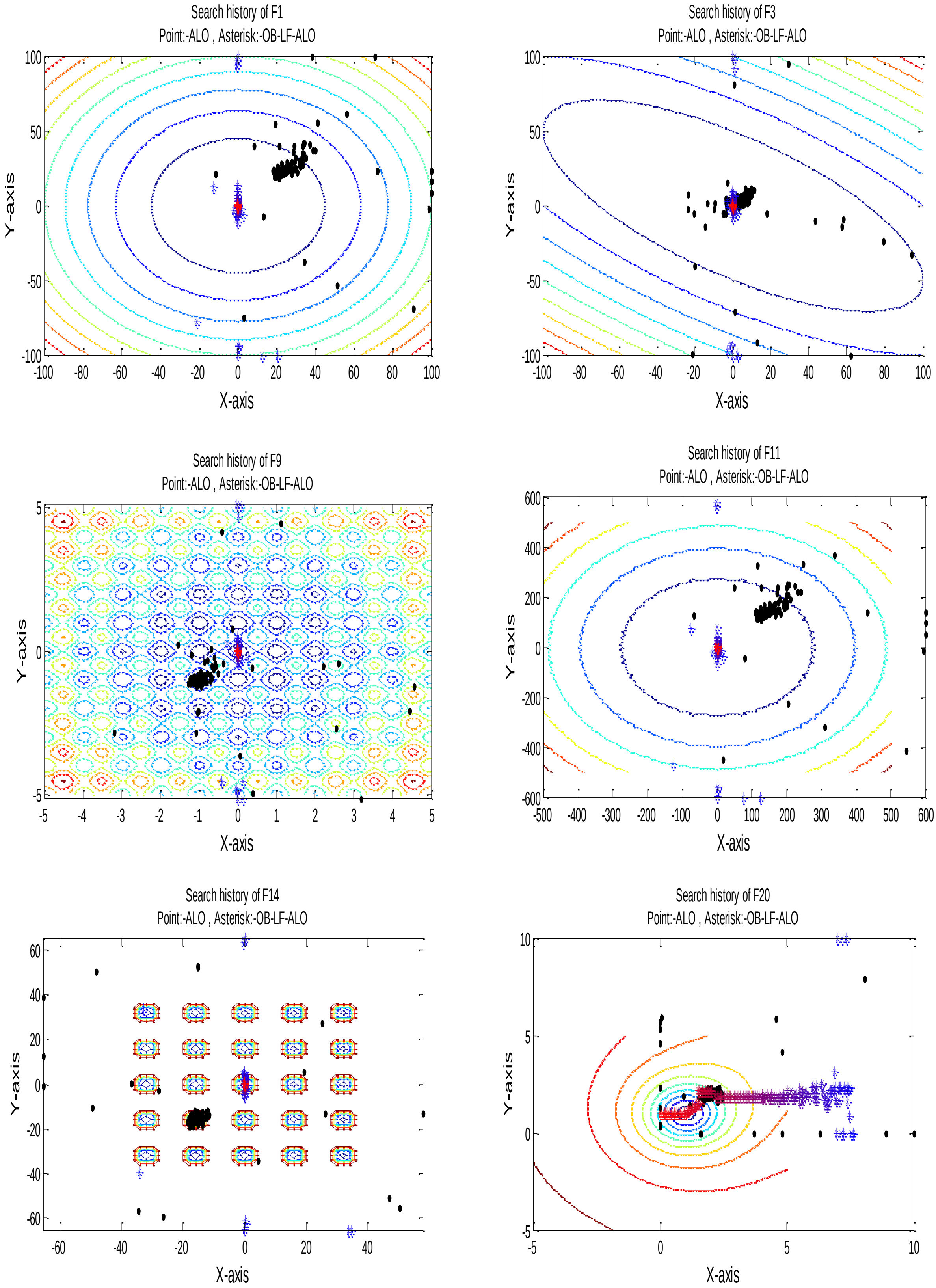


Figure 4.6: Search history analysis of functions $F1, F3, F9, F11, F14$ and $F20$

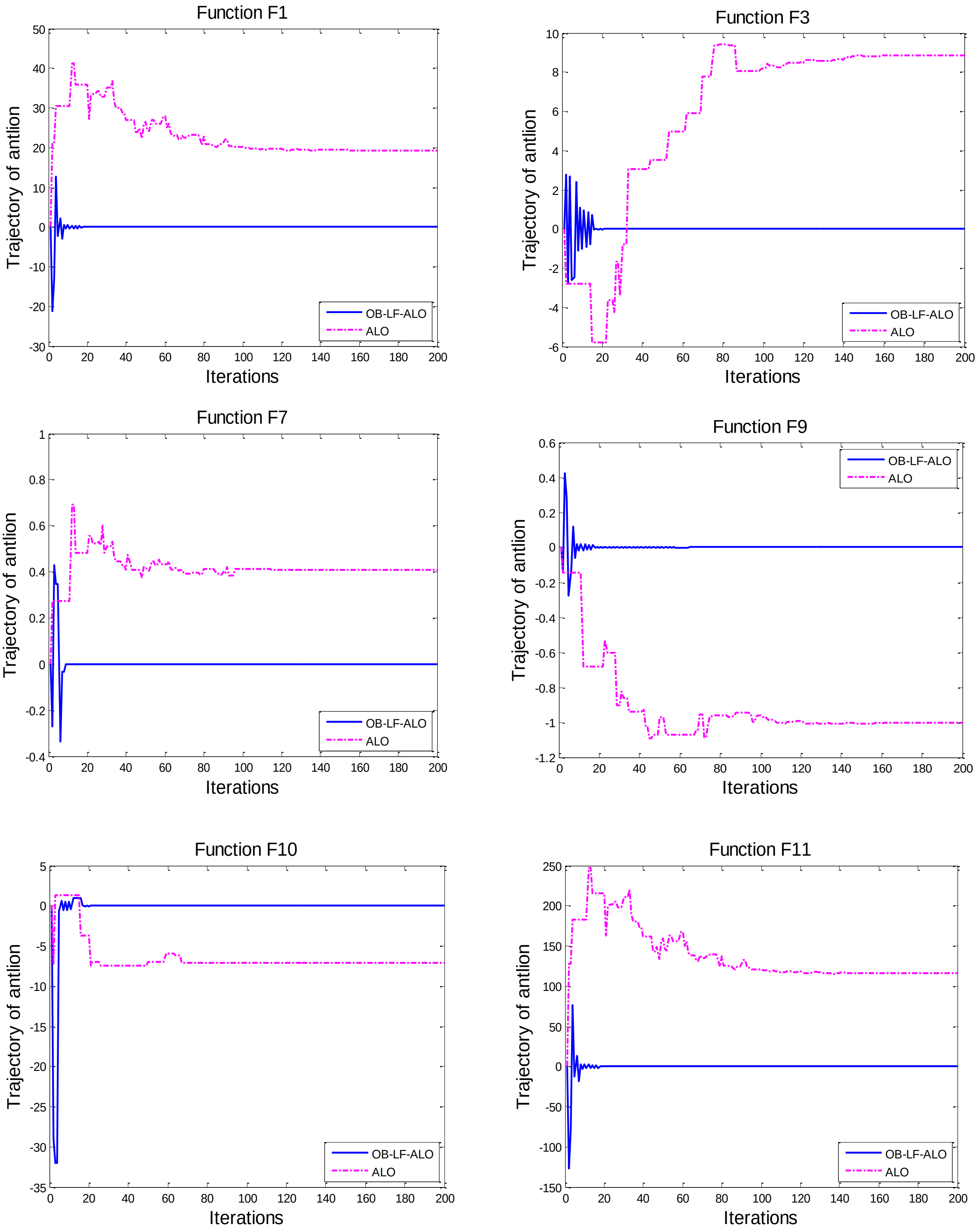


Figure 4.7: Trajectories of antlion

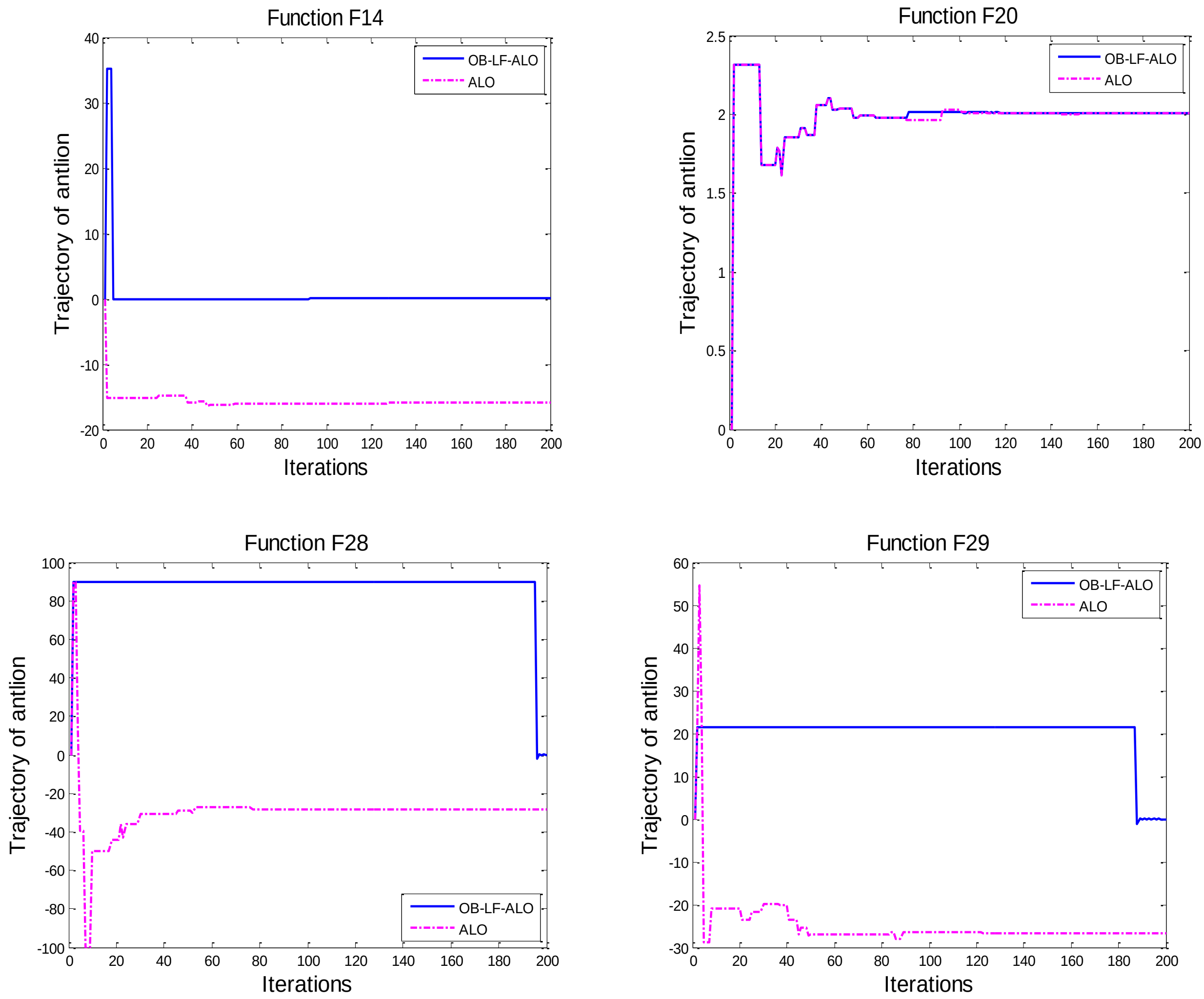


Figure 4.7: (Continued)

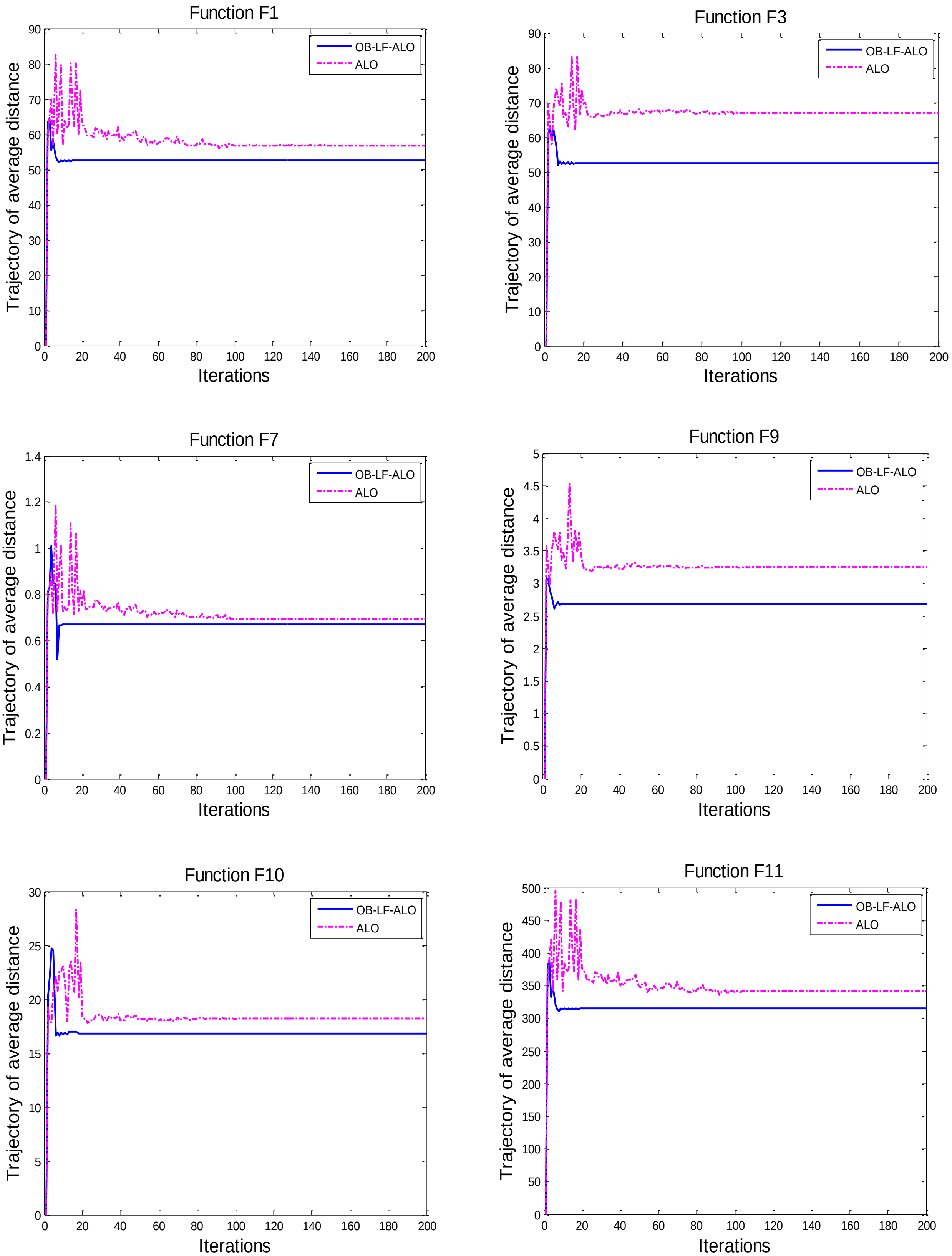


Figure 4.8: Average distance analysis between search agents

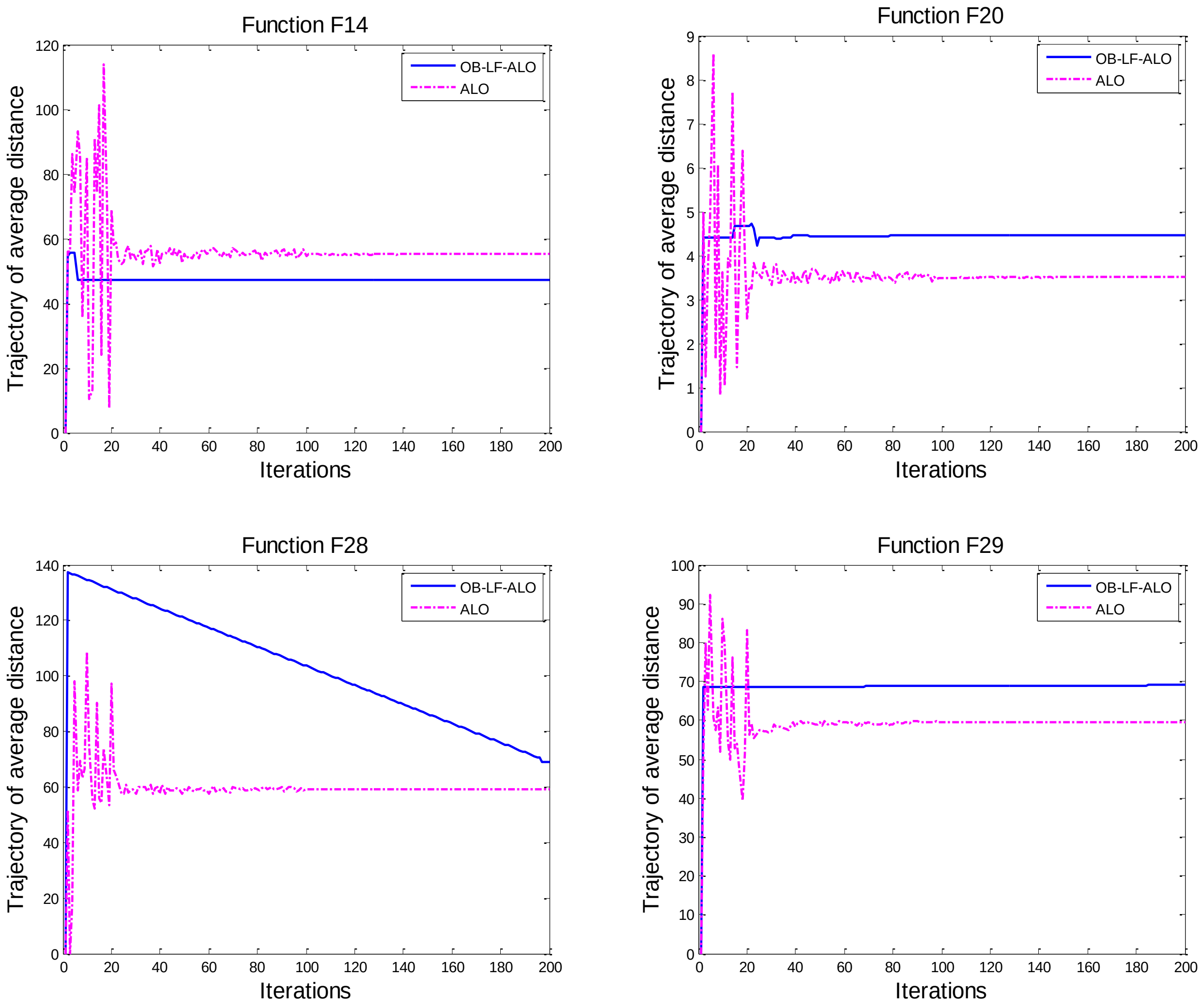


Figure 4.8: (Continued)

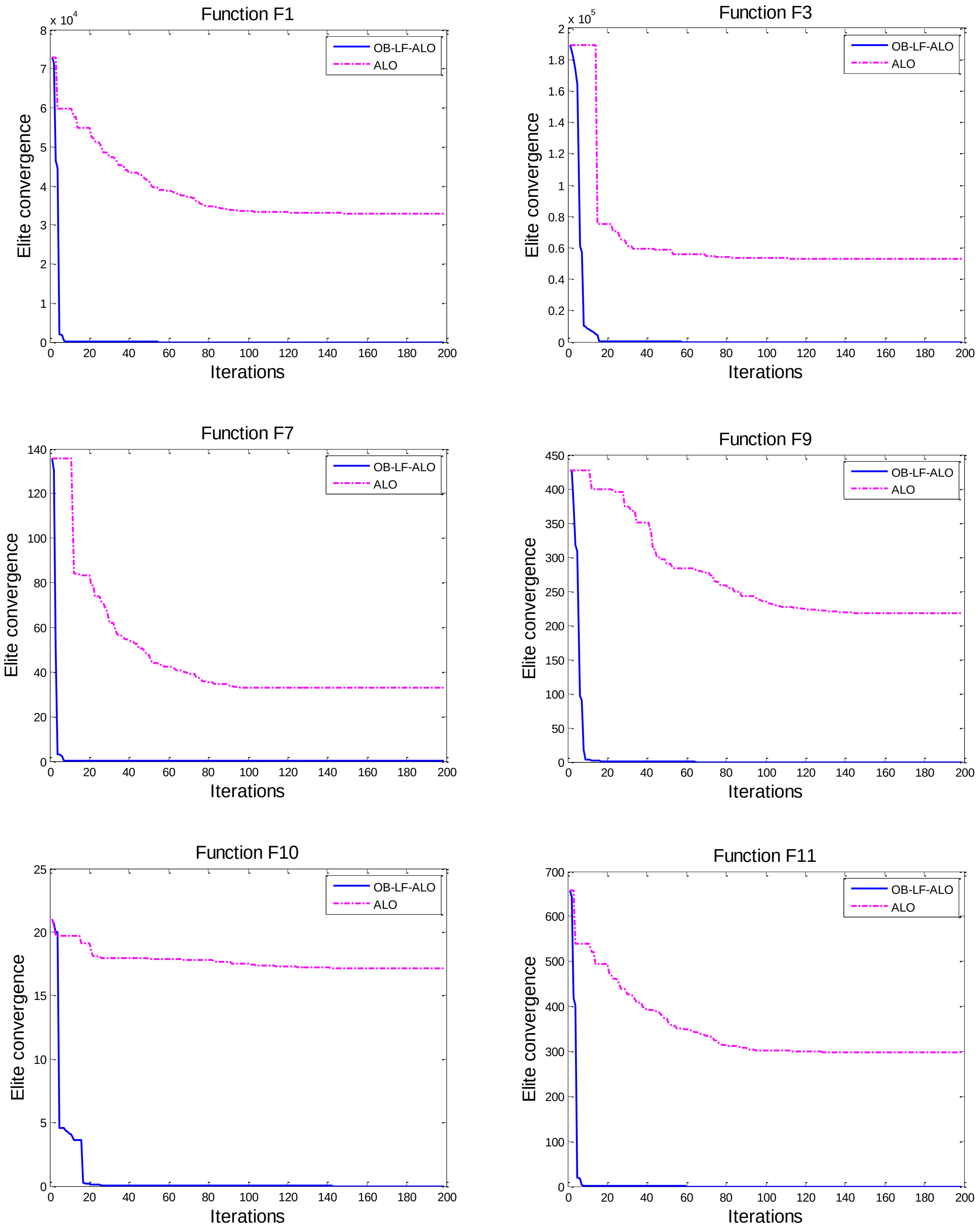


Figure 4.9: Elite convergence curve

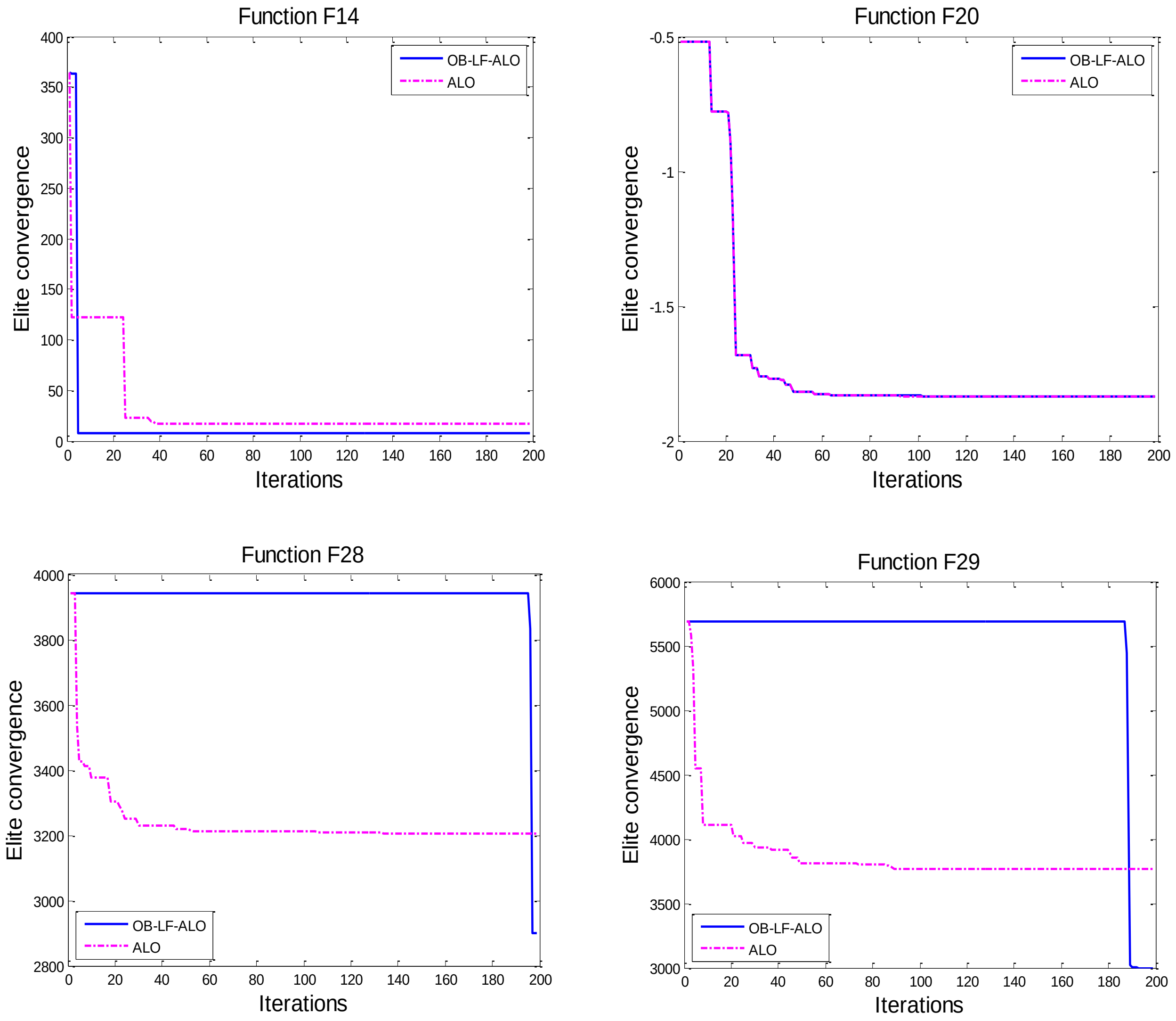


Figure 4.9: (Continued)

CHAPTER 5

Cauchy Distributed Opposition based Antlion optimizer with Acceleration Coefficient (OB-C-ALO) for Continuous Optimization Problems

5.1 Introduction

This chapter is divided into two parts. In first part, another extended variant of opposition based ALO is proposed to enhance the efficiency of classical ALO. Three strategies are employed to do so. (1) Cauchy distribution based crossover mechanism to manipulate the random walk for exploration enhancement of the search region. (2) Opposition based learning(OBL) mechanism is applied to the best candidate solution to explore the nearby positions of global optima.(3) Acceleration coefficient(ac) parameter to adjust the exploitation at later stages of the iterations and thus able to increase the convergence of proposed algorithm.

Tendency of classical ALO to be entrapped in local optima requires increased diversification of the search space to come out of local solution. Proposed variant of classical ALO attempts improve balance between diversification and intensification while speeding up the convergence. To evaluate the performance, same set of benchmark functions utilized in Chapters 2, 3 and 4 has been chosen. To evaluate the impact of Cauchy distribution based random numbers and opposition based learning (OBL) model integrated with acceleration coefficient, various metrics have considered such as trajectories, elite convergence curve, and average of absolute distance between search agents before and after improving the algorithm. A non-parametric Wilcoxon ranksum test is used to exhibit its statistical significance and computational complexity of the proposed algorithm is also discussed.

In second part of this chapter, the performance of all the proposed variants of classical ALO developed in previous chapters as well as in this chapter are compared and analyzed using the similar set of benchmark functions and analysis metrics

---

Partial content of this chapter has been published as:

- **Dinkar, S. K., & Deep, K. (2019). Opposition based Antlion Optimizer using Cauchy distribution and its application to data clustering problem. Neural Computing and Applications, Springer. https: //doi.org/10.1007/s00521-019-04174-0. (SCIE, IF-4.213).**

This chapter is organized as follows: Section 5.2 describes related literature and motivation. Section 5.3 represents the proposed OB-C-ALO algorithm including conceptual material about employed strategies. Section 5.4 depicts the description of benchmark test functions and experimental setting. Section 5.5 shows the computational results, comparison and discussion as per the characteristics of benchmark functions. Section 5.6 depicts convergence and other analysis using various metrics to authenticate newly developed OB-C-ALO and classical ALO. Afterwards, Section 5.7 represents performance comparison of all the modified variants proposed in previous chapters. Section 5.8 gives concluding remarks.

5.2 Motivation and Related Literature

The mechanism of random number generation during the searching process of the region plays a vital role. The aim is to improve the searching of unvisited area of the search region so that the proposed algorithm is capable of avoiding local optima during evolution process[107]. Then opposition based learning (OBL) is employed after position update using random walk at later part of generation as a second strategy. After employing Cauchy distribution followed by OBL mechanism, an acceleration coefficient is merged with OBL technique as third strategy[111] to maintain balance between early and later part of evolutionary process[112].

5.3 Proposed Opposition based ALO with Cauchy distribution(OB-C-ALO)

This section describes the extended version of opposition based ALO using new exploration strategy namely Cauchy distribution to improve the performance of classic ALO. In spite of good convergence performance of classical ALO, some candidate solutions may not be able to move towards global optima and may get trapped in local optima. It leads the necessity to improve the diversification of search region during process of evolution. For accomplishing it, the random numbers are generated using Cauchy distribution (CD) in place of uniformly generated random numbers in classical ALO. At later stages, the opposite points of updated positions of candidates solutions are determined using opposition based learning (OBL) to accelerate the convergence and exploitation in conjunction with acceleration coefficient. The applied mechanism are described in the following subsections.

5.3.1 Cauchy distribution

The search ability of an optimization algorithm can be improved by approximating the candidate solution nearby to the global solution in each generation during evolution process. During this process, the candidate solutions initially generated randomly, can move to better positions. In classical ALO, random walk of ants around antlion is performed using uniform distribution. In this work, it is replaced with the Cauchy distribution in order to enhance the exploration. The Cauchy distribution (CD) is successfully employed in many nature inspired optimization algorithms such as Krill Herd [107], PSO [127-128] and DE [129-130].

The probability distribution function (PDF) of Cauchy distribution (CD) denoted by $C(a,b)$ can be defined as [131-132]:

$$f(x;a,b) = \frac{1}{\pi b}\left[\frac{b^2}{(x-a)^2+b^2}\right] \tag{5.1}$$

Here a describes the location of the peak of distribution known as location parameter and $b > 0$ specifies the scale parameter. In this work, the experimental analysis establishes that the appropriate values of a and b are chosen to be as 1 and 5 respectively after performing a number of experiments with different values of a and b.

The Cauchy distributed function can be defined as:

$$F(x;a,b) = \frac{1}{\pi}\arctan\left(\frac{x-a}{b}\right) + \frac{1}{2} \tag{5.2}$$

Random numbers generated uniformly and using Cauchy distribution over 200 numbers of iterations are shown in Figure 5.1. It is evident from the figure that inclusion of Cauchy distribution in random walk spans more search area and provides new candidate solutions which may not be approximated using uniform distribution. This mechanism promises to explore the search region while avoiding stagnation to local optima and shows tendency of moving towards global optima.

Replacing uniform distributed random numbers with Cauchy distributed random numbers can be mathematically reformulated as

$$r_w(S_{A,n}^d) = [cumsum(C(it_1)\text{-}1),\ cumsum(C(it_2)\text{-}1)\ldots\ cumsum(C(it_{max})\text{-}1)] \tag{5.3}$$

Here C represents the Cauchy distributed random numbers and $cumsum$ defines the cumulative sum of random numbers in each iteration it.

5.3.2 Opposition based learning (OBL) Model

The definition of opposite numbers is defined in Section 2.3.2 of Chapter 2. It is concluded that the fitness of the candidate solutions is directly proportional to the distance between these solutions to global optima. Hence, the candidates nearer to the optima will be fitter as compared to other candidates.

5.3.3 Acceleration Coefficient (ac) Parameter

For accelerating the exploitation at later stages, a parameter e_{ac} is applied to approximate the exploitation convergence which is decreased adaptively as the number of iteration increases. This parameter e_{ac} can be determined as shown in eq. (3.2), section 3.3.1 of chapter 3.

The parameter e_{ac} is hybridized after applying OBL technique. Mathematically, this process can be defined by reformulating in eq. (5.4) as follows:

$$S_{A,i}(it) = e_{ac} \times (L_i + U_i - S_{elite}(it)) \quad (5.4)$$

5.3.4 Proposed OB-C-ALO algorithm

The strategies discussed above are employed to propose Opposition based ALO with Cauchy distribution (OB-C-ALO) to improve the performance of classical ALO in terms of refining balance between exploration and exploitation. The random walk is improved by applying Cauchy distributed random numbers as first strategy to enhance the exploration. It is followed by second strategy as OBL mechanism to generate the opposite positions of updated population. Then third strategy as exploitation parameter to adjust balance between exploration and exploitation which is integrated with OBL.

Pseudo code of newly developed OB-C-ALO is described in Table 5.1.

5.4 Experimental Setup

The performance of proposed OB-C-ALO and comparison of all modified variants are demonstrated by choosing similar set of benchmark test functions as depicted in Appendix I.

Benchmark test functions

These functions are of different characteristics including unimodal, multimodal, fixed dimension multimodal [108] and composition functions [109] and have been tested to verify the performance of proposed OB-C-ALO in terms of exploration and exploitation capability of algorithm.

Experimental and parameter setting

The performance of proposed OB-C-ALO is verified by comparing with classical ALO For scalable problems (unimodal and multimodal functions) by performing the experiments are for 10 and 30 dimensions. The performance of proposed algorithm is measured in terms of average, standard deviation, maximum and minimum of objective function values taken over 30 independent runs. These 30 independent runs have been performed by generating an initial population size of 30. Stopping criteria is fixed at 500 and 1000 iterations for 10 and 30 dimensions respectively. For composition functions, the problems are taken of 10 dimensions with stopping criteria of 1000 iteration having 51 independent runs. All the experiments have been performed on MATLAB R2014a on Intel(R) Core(TM) i5-7200 CPU @ 2.50GHz-2.71 GHz with 8GB RAM.

5.5 Experimental Results and discussion

The results on various benchmark test function of proposed OB-C-ALO algorithm are shown in different tables. Table 5.2, Table 5.3, Table 5.4, Table 5.5 exhibit results on 10 and 30 dimensions(scalable) for unimodal and multimodal functions whereas Table 5.6 shows the results on fixed dimensions multimodal respectively. Table 5.7 depicts results of composition functions.

Evaluation of exploitation capability (Function $F1 - F7$)

The results of proposed OB-C-ALO over unimodal functions are shown in Table 5.2 and Table 5.3 for 10 and 30 dimensions respectively. It is clear from the tables that the proposed OB-C-ALO attains global optima (average objective function values) for functions $F1 - F4$ with zero standard deviation which shows that the proposed algorithm is stable throughout the evolution process and performs significantly better than the classical ALO for both 10 and 30 dimensions. It is also evident from the tables that the proposed OB-C-ALO shows slightly better average value than classical ALO for functions $F5$ and $F7$. This performance in terms of average fitness values shows that inclusion of Cauchy distributed random numbers in random walk followed by OBL mechanism in conjunction with acceleration coefficient is capable of reaching global optima or near to global optima at later stages during exploitation phase. The results clearly demonstrates that OBL mechanism could find the opposite candidate solution nearer to the global optima in comparison to other algorithms. It is also evident from the table that the average value of standard deviation is steady throughout the evolution process and authorizes constancy and superiority over classical ALO.

Evaluation of exploration capability (Function $F8 - F13$)

The results for multimodal functions $F8 - F13$ are depicted in Table 5.4 and Table 5.5 for 10 and 30 dimensions respectively. It is evident from the tables that the proposed OB-C-ALO performs better in terms of average objective function values for functions $F8 - F12$ for 10 dimension and for functions $F8 - F13$ for 30 dimensions as compared to classical ALO. It is also clear from the tables that OB-C-ALO attains global optima for functions $F9$ and $F11$ for both 10 and 30 dimension respectively. The analysis of results clearly demonstrates the enhanced exploration capability of proposed OB-C-ALO by employing Cauchy distributed random walk. Also the results of functions $F9$ and $F11$ indicates that OBL mechanism with acceleration coefficient is able to accelerate the convergence at later part of evolution. The average standard deviation for functions $F8 - F12$ also authorises the superiority and stability of proposed algorithm over classical ALO.

Evaluation of exploration capability of fixed dimension multimodal functions ($F14 - F23$)

Though these functions are multimodal yet the behaviour of these functions are contrary to the scalable multimodal function due to non-scalability. Hence it is always interesting to analyse these fixed dimensional functions. The results are exhibited in Table 5.6 which are taken when the stopping criteria is to achieve a maximum of 1000 iterations. It is evident from the table that the proposed OB-C-ALO performs slightly better in function $F14$ and $F15$ than classical ALO in terms of average objective function value and steady standard deviation. Proposed algorithms also attain better average objective function values for functions $F19$,$F20$ and $F21$ than the classical ALO. The obtained results for functions $F16$,$F17$,$F18$,$F22$ and $F23$ are similar for OB-C-ALO and classical ALO. After comparison of results, it can be concluded that the applied strategies are efficiently able to improve the performance of proposed OB-C-ALO.

Results on Composition Functions ($F24 - F31$)

The results for composition functions are exhibited in Table 5.7. These functions define complex problem matching with the real world optimization applications and henceforth difficult to get the global optima. It is evident from the table that the proposed OB-C-ALO outperforms the classical ALO for functions $F24, F27, F28, F29, F30$ and $F31$ significantly in terms of average objective function values. It can be observed from the result tables that the standard deviation for proposed OB-C-ALO is steady as compared to classical ALO exhibiting the stable performance.

The employed strategies proved to be more efficient for proposed OB-C-ALO as compared to classic ALO.

5.6 Analysis

The proposed algorithm is analyzed using various performance metrics as follows:

- 5.6.1 Convergence Behaviour
- 5.6.2 Statistical analysis- Wilcoxon ranksum Test
- 5.6.3 Analysis of Proposed Variant
    - Search history analysis
    - Trajectory analysis
    - Trajectory analysis of average distance between search agents
    - Elite convergence curve
- 5.6.4 Computational Complexity

5.6.1 Convergence Behaviour

To evaluate the performance of proposed OB-C-ALO algorithm, graphical interpretation of convergence behaviour of OB-C-ALO algorithm is analyzed and compared with classical ALO. Various benchmark test functions of wide variety of characteristics are chosen to authorize the performance of OB-C-ALO. The number of generations and the best average value of objective functions for 30 independent runs have been plotted on horizontal-axis and log scale of vertical-axis respectively.

In Figure 5.2, three curves showing convergence behaviour over fixed number of generations for unimodal functions $F1$,$F3$ and $F7$ are depicted. It is evident from the curves that the proposed OB-C-ALO starts converging in early generations than the classical ALO. The curves are clearly showing that proposed OB-C-ALO attains global optima in 100 and 10 generations for function $F1$ and $F3$ respectively which shows superiority of proposed algorithm over other versions. The proposed OB-C-ALO shows better convergence behaviour for function $F7$. This behaviour demonstrates that inclusion of OBL followed by enhanced exploration using Cauchy distribution is highly impactful for convergence acceleration and capable of finding global optima at early stages of generations. This behaviour of proposed algorithm promises to converge to global optima [110].

In Figure 5.3, curves of three scalable multimodal function $F9$, $F10$ and $F11$ are drawn and compared with classical ALO. The proposed algorithm shows sharp convergence from the initial generations and attains global optima for function $F9$ and $F11$ in around 300 generations each. Function $F10$ also exhibits excellent convergence accelerations. It can be clearly observed that the employed strategies are useful for achieving high convergence speed even at early stages of generations.

Figure 5.4 depicts convergence curves for fixed or non-scalable multimodal functions $F14$ and $F20$. Proposed OB-C-ALO exhibits ample improvement in convergence as compared to classical ALO. However, for function $F20$, OB-C-ALO shows better convergence behaviour than classical ALO. Overall, it can be concluded that inclusion of OBL model promises to enhance diversification by exploring the opposite candidate solution simultaneously with the original solutions and able to avoid entrapment in local optima. Figure 5.5 shows convergence curves for functions $F29$ and $F31$. It is evident from the curves that the proposed OB-C-ALO is clearly a winner as compared to classical ALO which can be also observed from the tables of results. These composition functions contain many optima like multimodal functions and prone to be entrapped in local optima thus restricted to reach global optima. But the inclusion of Cauchy distribution in random walk ensures to explore the unvisited region of the search space and tries to avoid local optima. Due to this fact, the proposed algorithm is able to reach global optima or near to optima with increased rate of convergence due to opposition based learning model followed by Cauchy distribution.

5.6.2 Statistical analysis-Wilcoxon Ranksum Test

A non-parametric Wilcoxon ranksum test is applied to investigate the statistical significance of proposed OB-C-ALO with respect to classical ALO. This test analyze the distribution of sampled data drawn from two algorithms in pair. The obtained result in terms of p-values authorizes the data drawn from one algorithm is statistically significant than the data drawn from other algorithm.

The confidence level is taken as 0.95. The sampled data drawn for 30 dimensions from two different algorithms is used in pair from 30 independent runs to reject the null hypothesis. This hypothesis checks that the data drawn of continuous distribution from two algorithms with equal median against the alternative that actually they are not. The obtained results in terms of p-values are shown in Table 5.8. This table depicts the comparison of statistical significance for proposed OB-C-ALO with classical ALO. The '+' denotes that the statistical difference is significant at 0.05 level of significance, '-' indicates that there is no statistical difference and '=' depicts that the sampled data drawn from the pair of algorithm are similar and comparison is not possible.

It is clearly demonstrated from Table 5.8 that the proposed OB-C-ALO is statistically significant for most of the functions i.e.20 test problems and remain same for 5 problems $F16, F17$, $F18$,$F22$ and $F23$ where both the algorithms obtain optima. Out of these problems, OB-C-ALO is statistical significant for 6 problems out of 8 problems of composition functions. Overall, it can be concluded that the proposed OB-C-ALO clearly outperforms the classical ALO.

5.6.3 Analysis of Proposed Variant

The modified version OB-C-ALO of classical ALO is proposed to enhance exploration and acceleration in convergence to achieve proper balance between exploration and exploitation at early as well as later stages of generations. These improvements can be authorized by employing certain metrics such as (1) Search history analysis (2) Trajectory analysis(3) Average distance between search agents before and after applying improvement strategies and (4)Elite convergence for analyzing the performance of proposed version. For this purpose some benchmark test functions of different characteristics have been used. All the experiments for analysis are performed by employing four search agents for 30 dimensions over 200 iterations and compared with classical ALO with same parameter setting. The analysis using various metric is as follows:

Search history analysis

This analysis is performed to verify the searching ability of the search agents of original ALO and the proposed OB-C-ALO after applying the improvement strategies. The distribution of population is drawn over the contour of the search space of the respective functions as shown in Figure 5.6. To establish this analysis, the contour and surface plot of one unimodal function $F3$ and one multimodal function $F11$ are drawn. The search agents with 'square' and 'asterisks' represents the classical ALO and proposed OB-C-ALO over the search space respectively. It is clearly evident from the figure that after replacing uniformly distributed random numbers from random walk with Cauchy distributed random numbers, the proposed technique is more proficient in searching prominent region accurately. It is also visible that implementing OBL mechanism with acceleration coefficient tends to search nearby positions of global optima. It can be concluded that the proposed OB-C-ALO is spanning the prominent region of the search space at early generations and promises to accelerate the convergence rate.

Analysis of trajectory

The trajectories of best (elite) candidate solution depicted in Figure 5.7 as graphical form to analyze the movement behaviour of best candidate solution towards the global optima. Trajectories of elite candidate solution i.e. first variable are drawn up to maximum (200) iterations. This analysis is useful to investigate that the proposed algorithm OB-C-ALO is capable to determine the nearer positions to global optima as compared to classical ALO.

Few problems of benchmark test functions of wide characteristics are used to perform this analysis including unimodal function ($F1$,$F3, F7$), multimodal functions ($F9, F10$,$F11$),two fixed dimensions multimodal, ($F14$,$F20$) and two composition($F28$,$F29$) functions. It is clearly evident from the figure that the proposed OB-C-ALO shows enhanced exploration due to employment of Cauchy distribution at early generations. It is visible that the positions of elite in OB-C-ALO shows sharp acceleration in convergence for functions $F1$,$F3, F7$, $F9$ and $F11$ and authorises that the use of OBL mechanism with acceleration coefficient parameter is capable of finding nearer positions to the global optima in comparison to classical ALO. Similarly, for function $F10$ the exploration is enhanced with great extent at early generations and then converges to the nearby positions of global optima. However, for function $F14$,the average objective function values of OB-C-ALO is better than classical ALO as shown in result table but the trajectory of the candidate solution depicts abrupt behaviour initially but gets steady at later generations. Whereas, for function $F20$, the proposed OB-C-ALO is not able to approximate this nearby positions of optima. The trajectory of composition function $F28$ shows similar behaviour as of classical ALO but it approximates the closer positions to global optima at the later generations as compared to classical ALO. Similarly, the trajectory of function $F29$ is far than the ALO but approximates the nearer positions at later generations. It can be concluded that the performance of OB-C-ALO outperform the classical ALO as far as the searching of closer positions of the global optima.

Trajectory analysis of average distance between search agents

The absolute average distance between search agents before and after employing improvements strategies is determined in each iteration. The obtained value is than compared with the average distance of classical ALO to establish the performance of OB-C-ALO. The experiments are conducted for 200 iterations over first variable by choosing ten benchmark test functions including three unimodal($F1, F3, F7$),,three multimodal($F9, F10, F11$),two fixed dimension multimodal($F14, F20$) and two composition ($F29$,$F30$) functions.to validate the performance of OB-C-ALO.

The curves shown in Figure 5.8 depicts the determined average distance using proposed OB-C-ALO. From the figure, it is shown that the average distance is fluctuating at early generations due to enhanced exploration caused by Cauchy distribution. Then it gets smooth and steady once distance becomes lesser or closer to optimal solution at later generations. This behaviour is caused by employing OBL integrated with exploitation adjustment parameter. For functions $F1, F3, F7, F9, F10$ and $F11$,the determined average distance shows steady behaviour around the optimal position as OB-C-ALO performs better than classical ALO as shown in tables of results. This implies that the obtained distance is closer to optima as compared to other algorithms. The graph for function $F14$ shows that proposed algorithm displays closer distance as compared to classical ALO at later generations. However, function $F20$ shows uneven performance at later generation though OB-C-ALO obtains better results in comparison to classical ALO. Some unusual behaviour is shown by the curves for functions $F29$ and $F30$ where the determined average distance is far away as compared to classical ALO but it gets closer and coincides at later generations. However the results in terms of mean value and standard deviation of objective function values are better for proposed OB-C-ALO in comparison to classical ALO. Overall, this analysis establishes that the obtained average distance using proposed OB-C-ALO promises to accelerate the convergence.

Elite convergence curve

This metric verifies the convergence capability of elite candidate solutions over 200 iterations. This convergence rate is shown as a curve of best antlion (elite) of first variable. The ten functions as used in the previous section have been analyzed for elite convergence curve and depicted in Figure 5.9.

The curves for elite convergence as depicted in Figure 5.9 clearly shows that the fitness of elite solution drops significantly starting from initial iterations which shows that the elite solution starts to converge even at initial iterations for proposed OB-C-ALO as compared to classical ALO. The classical ALO shows better convergence behaviour for function $F28$ in initial iterations but curve of proposed OB-C-ALO overtakes it at later generations. The curve of $F29$ depicts better convergence steadily than ALO and further drops at later generation and exhibits the superiority of proposed OB-C-ALO over classical ALO. This analysis validates high acceleration in convergence with the increase in number of iterations and tries efficiently to search global optima with more precision. The employability of Cauchy distribution ensures the enhanced exploration and avoidance of local optima stagnation. The opposition based learning

(OB) is capable of approximating the nearer solution to the best ant lion (elite) and ensures the acceleration in convergence towards a point.

5.6.4 Computational Complexity

Though the initial population is generated using OBL mechanism, yet there is no additional inclusion of functional evaluation or any extra loop during evolution process to determine the updated position. It ensures no change in asymptotic computational complexity and it remains same as of classical ALO as described in chapter 2, chapter 3 and chapter 4. Thus, it is observed as $O(it_{max} * D * N) * O\ (f(X))$ where it_{max} maximum number of iteration, D represents dimensions, N is population size and $f(X)$ defined as the objective function for worst case scenario.

5.7 Performance Comparison of Proposed Variants of Classical ALO

This section presents the perfromance comparison among proposed variants namely OB-L-ALO, OB-ac-ALO, OB-SAC-ALO, OB-LF-ALO and OB-C-ALO of classical ALO in previous as well as in this chapter.

5.7.1 Results and Discussion

The performance of proposed versions described in previous chapters are compared in terms of average, standard deviation, minimum and maximum objective function values taken over 30 independent runs. The same parameter setting is chosen for comparison and the obtained results are depicted in Table 5.9 and Table 5.10 for unimodal functions, Table 5.11 and Table 5.12 for multimodal functions, Table 5.13 for fixed dimensions multimodal and Table 5.14 for composition functions.

Comparison of results on unimodal functions ($F1$-$F7$)

The results on unimodal functions validate the exploitation capability of an optimization algorithms due to presence of single optima. The results are represented in Table 5.9 and Table 5.10 for 10 and 30 dimensions respectively. The average value of objective functions in Table 5.9 shows that the three proposed algorithms namely OB-C-ALO,OB-LF-ALO and OB-ac-ALO are capable of obtaining the exact global optima for function $F1$,$F2$.$F3$ and $F4$ in both 10 and 30 dimension. However, OB-SAC-ALO achieves global optima for function $F2$ and $F3$ for 10 dimension and $F2$, $F3$ and $F4$ for 30 dimension. OB-C-ALO performs better than other algorithms in terms of average fitness value for function $F5$ and OB-ac-ALO wins for function

$F7$ for 10 dimension. But OB-ac-ALO obtains better average fitness value for Function $F5$ and OB-LF-ALO performs better for function $F7$ in 30 dimensional problems. However, no proposed algorithm performs better than ALO for function $F6$.

Comparison of results on multimodal functions ($\boldsymbol{F8}$-$\boldsymbol{F13}$)

These problems are used to investigate the explorative capability of proposed algorithms. Obtained results are presented in Tables 5.11 and 5.12 for 10 and 30 dimensions respectively. Four algorithms namely OB-C-ALO, OB-LF-ALO, OB-ac-ALO and OB-SAC-ALO attains the optima value for function $F9$ and $F11$ as evident from the average fitness value for 10 and 30 dimensions. For function $F8$, OB-C-ALO obtains best value and for function $F12$, OB-C-ALO and OB-ac-ALO perform better than other algorithms for 10 dimensions. OB-L-ALO exhibits its superiority in function $F13$ as compared to other algorithms. For 30 dimension, OB-ac-ALO performs better for function $F8$ and OB-SAC-ALO wins in case of function $F12$ and $F13$. The less fluctuated values of standard deviations for proposed algorithm describe the steady and stable performance achieved by the proposed algorithms. The overall obtained results conclude the superior and enhanced performance of proposed algorithms as compared to classical ALO.

Comparison of results on fixed dimension multimodal functions ($\boldsymbol{F14-F23}$)

Analyzing the behaviour of fixed dimension multimodal functions by observing the results is interesting due to the non-scalable nature of these problems. The results are exhibited in Table 5.13 which are taken when the stopping criteria is to achieve a maximum of 1000 iterations. It is evident from the table that the OB-LF-ALO performs slightly better in function $F14$ and OB-L-ALO shows its superiority in $F15$ as compared to other variants of classical ALO in terms of average objective function value and steady standard deviation. Interestingly, all proposed algorithms including ALO exhibit same results and able to obtain global optima for function $F16$, $F17$, $F18$,$F22$ and $F23$. However, OB-L-ALO performs better for remaining $F19$, $F20$ and $F21$ by keeping OB-C-ALO at second place. After analysing the results table, it can be concluded that the applied proposed strategies for developing new algorithms are capable of enhancing exploration as well as exploitation ability of the classical ALO.

Results on Composition Functions ($\boldsymbol{F24-F31}$)

The obtained results for composition functions are depicted in Table 5.14. The evaluation of proposed variants of classical ALO over these problem reflect the ability of these algorithms to solve complex and hybrid problems related to the real life optimization problems. For function $F24$, three algorithms namely OB-C-ALO, OB-LF-ALO and OB-ac-ALO performs equally better than other algorithms. For function $F25$ and $F26$, OB-SAC-ALO performs better in terms of obtained average objective function value. For function $F27$ and 29 , OB-C-ALO clearly beats other proposed algorithms including classical ALO. Whereas OB-ac-ALO performs better for function $F28$ and equally better with OB-C-ALO and OB-LF-ALO for function $F29$. For function $F30$ and $F31$, OB-L-ALO shows its superiority over other algorithms. It is clearly evident that no one algorithm gets superior results for all composition functions however, OB-C-ALO outperforms other algorithms for majority of the composition functions as compared to other techniques.

5.7.2 Analysis

The performance of all proposed algorithms is analyzed using various performance metrics as follows:

- 5.7.2.1 Convergence Behaviour
- 5.7.2.2 Analysis of all proposed variants
  - Trajectory analysis
  - Trajectory analysis of average distance between search agents
  - Elite convergence curve

5.7.2.1 Convergence Behaviour

The comparison among all the proposed versions of classical ALO is performed in terms of convergence curve. A set of benchmark functions of different characteristics have been used to validate the performance. The number of generations and the best average value of objective functions for 30 independent runs have been plotted on horizontal-axis and log scale of vertical-axis respectively.

Figure 5.10 shows convergence curves of three unimodal function $F1$,$F3$ and $F7$. The curves in graphs depict that three algorithms namely OB-C-ALO, OB-LF-ALO and OB-ac-ALO are able to achieve global optima in 100 generations for function $F1$. For function $F3$, OB-SAC-ALO also obtains optima in addition to first three algorithms but OB-C-ALO obtains the global optima in even 10 generations and outperforms other algorithms marginally. For function $F7$,OB-LF-ALO performs better in comparison to other algorithms. This convergence behaviour clearly demonstrates the superiority of applied techniques to classical ALO.

The convergence curves for two multimodal functions $F10$ and $F11$ are shown in Figure 5.11. OB-C-ALO, OB-LF-ALO and OB-ac-ALO exhibit fluctuation in early generations but converge at same point at later generations which can be verified by observing the result Table 5.11. The OB-SAC-ALO algorithm performs better than OB-L-ALO and classical ALO but not able to match with other algorithms. However OB-L-ALO also performs significantly better than classical ALO. Overall, it can be concluded that all the proposed algorithms in this thesis outperforms classical ALO.

Figure 5.12 demonstrates the convergence behaviour of two fixed dimensional multimodal functions $F14$ and $F20$. OB-SAC-ALO and OB-L-ALO outperform other proposed algorithms for functions $F14$ and $F20$ respectively. Figure 5.13 exhibits the curves for composition functions $F29$ and $F30$. It is observed from the curves that OB-C-ALO outperforms other proposed variants in function $F29$ and function $F30$ however OB-L-ALO performs almost similar to OB-C-ALO for function $F30$.

5.7.2.2 Analysis of all proposed variants

This subsection elaborates the various analysis to investigate the performance of proposed versions of classical ALO. These analysis try to verify the global exploration and exploitation by applying different analysis metrics such as: (1) Trajectory analysis (2) Average distance between search agents before and after applying improvement strategies and (3) Elite convergence for analyzing the performance of proposed version. For this purpose a set of benchmark test functions of different characteristics have been used. All the experiments for analysis are performed by employing four search agents for 30 dimensions over 200 iterations and compared with same parameter setting.

Analysis of trajectory

It investigates the behaviour of trajectory of best (elite) candidate solution determined in each iteration whether it is able to approximate the nearby trajectory or positions to global optima. Figure 5.14 represents the trajectories of best antlion for all proposed algorithms. The curves show the enhanced exploration at early generations due to applied strategies except and after few generations, trajectories become steady. For unimodal ($F1$,$F7$) and multimodal ($F9$, $F11$), the proposed OB-SAC-ALO exhibit abrupt exploration but gets closer to optima at later generations. However OB-C-LAO, OB-LF-ALO and Ob-ac-ALO become steady and stable at early generations. For fixed dimensional function $F14$,the average objective function values of OB-

C-ALO and OB-SAC-ALO are better than classical ALO as shown in result table but the trajectory of the candidate solution is depicting abrupt behaviour initially but gets steady at later generations. For function $F20$, it is observed from the curves that all proposed algorithms are able to approximate this nearby positions of optima. The trajectories of OB-LF-ALO and OB-C-ALO for composition functions $F28$ and $F29$ show highly abrupt behaviour where both of these algorithms are able to approximate the nearby position of optima at almost last generations in comparison to other algorithms.

Trajectory analysis of average distance between search agents

This analysis is performed by determining the average of absolute distance between the search agents or candidate solutions before and after applying the improvement strategies. The value of this distance is determined in each iteration. The obtained value is than compared among all the proposed version of classical ALO represented in terms of curves. The experiments are evaluated for 200 iterations over first variable by choosing a set of benchmark test functions containing two unimodal($F1, F7$),,two multimodal($F9, F11$),two fixed dimension multimodal($F14, F20$) and two composition ($F28$,$F29$) functions.to investigate the performance of proposed algorithms.

The curves shown in Figure 5.15 depicts fluctuation of determined distances in initial generations. Then it gets smooth and steady once distance becomes lesser or closer to optimal solution at later generations. This behaviour is caused by using OBL integrated with exploitation adjustment parameter. For functions $F1, F7, F9$ and $F11$ the determined average distance shows steady behaviour around the optimal position for all the proposed algorithms except classical ALO. It exhibits that the obtained distance is closer to optima for proposed variants of ALO as compared to classical. The graph for function $F14$ shows that proposed OB-C-ALO and OB-SAC-ALO algorithm determine closer distance at later generations after showing abrupt behaviour at initial generations. For function $F20$ shows uneven performance for all proposed algorithms except OB-L-ALO which shows steady and stable distance through all the iterations. The proposed algorithms OB-C-ALO,OB-LF ALO,OB-ac-ALO and OB-SAC-ALO exhibit unusual behaviour in spite of determining better average distance to the optima for function $F28$ and $F29$ as compared to classical ALO. However the results in terms of mean value and standard deviation of objective function values are better for proposed algorithms in comparison to classical ALO. Overall, this analysis establishes that the obtained average distance using proposed methods promise to accelerate the convergence.

Elite convergence curve

These curves are drawn to investigate the convergence capability of elite candidate solutions over 200 iterations. The same set of benchmark functions as chosen for trajectory and average distance are taken to analyze the behaviour of these curves.

The curves for elite convergence as depicted in Figure 5.16 clearly shows that the fitness of elite solution drops significantly starting from initial iterations for OB-LF-ALO followed by OB-C-ALO for functions $F1$ and $F7$. It shows that OB-LF-ALO outperforms other algorithm for function $F1$ and $F7$ as evident from the result tables as compared to other algorithms. The same pattern can be observed from the curves of functions $F9$ and $F11$. However,OB-ac-ALO,OB-SAC-ALO and OB-C-ALO show better convergence for function $F14$. OB-C-ALO outperforms other algorithms for function $F28$ and shows better fitness value for elite. For function $F29$, OB-C-ALO perform better and achieves best fitness value at later iterations though classical ALO performs better at initial generations. OB-ac-ALO also performs well and achieves same convergence as of OB-C-ALO at later iterations. This analysis validates high acceleration in convergence with the increase in number of iterations and tries efficiently to search global optima with more precision. The employability of applied techniques ensure the enhanced exploration and avoidance of local optima stagnation. The opposition based learning (OB) is capable of approximating the nearer solution to the best ant lion (elite) and ensures the acceleration in convergence towards a point.

5.8 Conclusion

In the first part of this chapter, a novel opposition based antlion optimizer with Cauchy distribution (OB-C-ALO) is proposed to improve the performance of classical ALO in terms of avoiding stagnation to local optima and convergence acceleration. The Cauchy distribution based random walk is capable of exploring search region in early generations followed by OBL mechanism with acceleration coefficient to improve local search at later generations. The obtained results on benchmark test functions authorize the superiority of proposed OB-C-ALO over classical ALO on unimodal, multimodal, fixed dimension multimodal and composition functions. The results on these functions are evident to show enhanced exploration and acceleration in convergence.

The wide analysis is also performed in terms of convergence curves, search history analysis, trajectory of best candidate solution, average distance between candidate solutions before and after applying improvement strategies and elite convergence curves. The search history of benchmark problems determine the searching behaviour of search agents before and after applying new strategies as shown from contour diagram of test suit. The trajectories of best candidate solutions depicts the changes in movement of its position towards the global optima during evolution process. The determination of average distance between search agents is capable of quantifying the impact of opposition based learning (OBL) by using opposite and initially generated random candidate solution as the less average distance exhibits the nearer solutions to the optima. However, the applied strategy causes random walk to be more complex and computationally intensive which enhances processing time of proposed OB-C-ALO as compared to classical ALO.

The second part of the chapter describes the comparison among all the proposed variants namely OB-C-ALO, OB-LF-ALO, OB-ac-ALO, OB-SAC-ALO and OB-L-ALO including classical ALO over a set of 31 benchmark test problems. It is evident from the results that three variants namely OB-C-ALO, OB-LF-ALO and OB-ac-ALO obtain exact global optima for eleven benchmark functions ($F1$,$F2$,$F3$,$F4$,$F9$,$F11$,$F16$,$F17$,$F18$, $F22$, $F23$). For functions $F5$ and $F8$(10 dimension) OB-C-ALO outperforms the other variants whereas OB-SAC-ALO performs better than other algorithms for functions $F12$ and $F13$. OB-LF-ALO performs best for function $F14$ and OB-L-ALO outperforms all other variant for function $F19$, $F20$ and $F21$. For composition functions, OB-C-ALO performs better for functions $F24$, $F27$, $F29$, $F30$ and $F31$. This comparison clearly demonstrates the importance of the statements of No Free Lunch Theorem. The performance order can be concluded as OB-C-ALO $>$ OB-ac-ALO $>$ OB-LF-ALO $>$ OB-SAC-ALO $>$ OB-L-ALO $>$ ALO.

Table 5.1**:** Pseudo code of Proposed OB-C-ALO algorithm

Input: *Population Size N, Maximum iteration* t_{max},*lower bound L, Upper bound U and dimension D*

Output: *The best candidate solution* S_{elite}

1 *Randomly initialize the initial population N of ants and antlions*

2 *Determine the objective(fitness) function value of antlions*

3 *Find out the best(with min fitness) antlion as the elite* S_{elite}

4 *Initialize iteration no.* $it_{curr}=2$

5 ***while*** $(\boldsymbol{it_{curr} \leq it_{max}})$

6 ***for every ant*** $(\boldsymbol{i = 1, 2, 3, \dots, N})$

7 *Find an ant lion* S_{sel} *using Roulette wheel*

8 *Modify lower L and upper U boundaries with equations Eqs.* $L^d(it) = \frac{L^d(it)}{I}$ *and* $U^d(it)=\frac{U^d(it)}{I}$)

9 ***for every dimension*** $(\boldsymbol{j = 1, 2, 3, \dots, D})$

10 *Perform random walk* $rw_A(it)$ *around* S_{sel}

11 *and* $rw_E(it)$ *around* S_{elite} *using Cauchy distribution with Eq.* $r_w(S_{A,n}^d)$ = [$cumsum(C(it_1)$-1), $cumsum(C(it_2)$-1)… $cumsum(C(it_{max})$-1)]

12 *Normalize random walk using Eqs* $S_{A,n}^d(it) = \frac{(S_{A,n}^d(it) - \min r_w(S_{A,n}^d))(U^d(it) - L^d(it))}{\max r_w(S_{A,n}^d) - \min r_w(S_{A,n}^d)} + L^d(it)$ *and* $rw(S_{A,n}^d)$=[$cumsum(2* C(it_1)$- 1),$cumsum(2* C(it_2)$-1),…,$cumsum(2*C(it_{max})$-1)

13 ***end for***

14 *Modify the position of ant using Eq.* $S_{A,n}^d(it)=\frac{r_{wA}(it)+r_{wE}(it)}{2}$

15 ***end for***

16 *Modify position of ant by applying acceleration parameter* e_{ac} *integrated*

17 *with opposition based learning(OBL) model using Eq.* $S_{A,i}(it) = e_{ac} \times (L_i + U_i - S_{elite}(it))$

18 ***for every ant*** $(\boldsymbol{i = 1, 2, 3, \dots, N})$

19 *Determine the fitness of all ants*

20 ***end for***

21 *Substitute an antlion with its respective ant if it becomes fitter using Eq.* $S_{AL,j}(it) = S_{A,i}(it)\ if\ f(S_{A,i}(it)) < f(S_{AL,j}(it))$

22 *Modify* S_{elite} *if an ant lion becomes fitter than the elite*

23 *Increment iteration i.e.* $it_{curr}=it_{curr}+1$

24 ***end while***

25 *Return elite*

Table 5.2 Average, Standard Deviation, Minimum, and Maximum of objective function values of unimodal functions (10 D)

Function	***Methods***	***Ave.***	***Std. Dev.***	***Min***	***Max***
$F1$	ALO	7.870E-09	5.490E-09	2.390E-09	**2.580E-08**
	OB-C-ALO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
$F2$	ALO	4.860E-01	8.850E-01	1.550E-05	**2.800E+00**
	OB-C-ALO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
$F3$	ALO	8.310E-02	1.850E-01	1.360E-04	**8.750E-01**
	OB-C-ALO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
$F4$	ALO	3.180E-03	6.160E-03	1.020E-04	**3.290E-02**
	OB-C-ALO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
$F5$	ALO	6.810E+01	1.990E+02	1.020E-04	**1.070E+03**
	OB-C-ALO	2.040E-04	3.910E-04	1.370E-07	1.830E-03
$F6$	ALO	**8.390E-09**	**5.220E-09**	**2.150E-09**	2.220E-08
	OB-C-ALO	4.580E-08	5.100E-08	2.870E-09	2.140E-07
	OB-L-ALO	3.320E-08	3.000E-08	7.360E-09	**1.160E-07**
$F7$	ALO	2.210E-02	1.230E-02	1.810E-03	**5.630E-02**
	OB-C-ALO	**2.660E-04**	**2.200E-04**	**9.950E-06**	**8.540E-04**

Table 5.3: Average, Standard Deviation, Minimum, and Maximum of objective function values of unimodal functions (30 D)

Function	***Methods***	***Ave.***	***Std. Dev.***	***Min***	***Max***
$F1$	ALO	9.450E-06	5.540E-06	1.720E-06	2.110E-05
	OB-C-ALO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
$F2$	ALO	3.650E+01	5.170E+01	2.090E-02	1.340E+02
	OB-C-ALO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
$F3$	ALO	9.550E+02	5.080E+02	2.470E+02	2.160E+03
	OB-C-ALO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
$F4$	ALO	1.210E+01	3.650E+00	4.630E+00	1.960E+01
	OB-C-ALO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
$F5$	ALO	1.040E+02	3.120E+02	1.640E+01	1.740E+03
	OB-C-ALO	**3.070E-03**	**4.720E-03**	**1.580E-06**	**1.950E-02**
$F6$	ALO	**9.490E-06**	**6.100E-06**	**1.320E-06**	**2.920E-05**
	OB-C-ALO	2.430E-05	1.650E-05	2.860E-06	6.450E-05
$F7$	ALO	1.100E-01	3.570E-02	5.700E-02	2.250E-01
	OB-C-ALO	**2.160E-04**	**1.880E-04**	**7.520E-06**	**6.610E-04**

Table 5.4 Average, Standard Deviation, Minimum, and Maximum of objective function values of multimodal functions (10D)

Function	***Methods***	***Ave.***	***Std. Dev.***	***Min***	***Max***
$F8$	ALO	-2.470E+03	4.260E+02	-3.730E+03	-1.920E+03
	OB-C-ALO	**-2.970E+03**	**6.230E+02**	**-3.970E+03**	**-2.090E+03**
$F9$	ALO	2.480E+01	1.080E+01	6.960E+00	**5.070E+01**
	OB-C-ALO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
$F10$	ALO	3.240E-01	5.540E-01	2.070E-05	**1.650E+00**
	OB-C-ALO	**8.880E-16**	**4.010E-31**	**8.880E-16**	**8.880E-16**
$F11$	ALO	2.140E-01	8.720E-02	6.640E-02	**4.060E-01**
	OB-C-ALO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-L-ALO	7.210E-10	2.110E-09	0.000E+00	9.600E-09
$F12$	ALO	1.790E+00	1.670E+00	8.230E-09	**5.610E+00**
	OB-C-ALO	**1.490E-08**	**1.410E-08**	**7.020E-10**	**5.600E-08**
$F13$	ALO	1.800E-03	4.940E-03	3.840E-09	2.100E-02
	OB-C-ALO	2.930E-03	4.940E-03	3.240E-09	1.100E-02

Table 5.5 Average, Standard Deviation, Minimum, and Maximum of objective function values of multimodal functions (30D)

Function	***Methods***	***Ave.***	***Std. Dev.***	***Min***	***Max***
$F8$	ALO	-5.440E+03	4.190E+01	-5.540E+03	-5.420E+03
	OB-C-ALO	**-1.180E+04**	**1.420E+03**	**-1.260E+04**	**-8.440E+03**
$F9$	ALO	1.970E+00	8.390E-01	5.890E-04	3.520E+00
	OB-C-ALO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
$F10$	ALO	1.870E+00	6.900E-01	1.020E-03	3.090E+00
	OB-C-ALO	**8.880E-16**	**4.010E-31**	**8.880E-16**	**8.880E-16**
$F11$	ALO	1.190E-02	1.260E-02	3.470E-04	3.860E-02
	OB-C-ALO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
$F12$	ALO	1.000E+01	4.340E+00	5.750E+00	2.330E+01
	OB-C-ALO	**1.090E-06**	**6.340E-07**	**3.560E-07**	**3.560E-06**
$F13$	ALO	1.320E+00	3.150E+00	1.140E-05	1.050E+01
	OB-C-ALO	**2.550E-03**	**5.430E-03**	**4.310E-06**	**2.100E-02**

Table 5.6: Average, Standard Deviation, Minimum, and Maximum of objective function values of fixed dimensional multimodal functions

Function	*Methods*	*Ave.*	*Std. Dev.*	*Min*	*Max*
$F14$	ALO	1.790E+00	1.340E+00	1.950E-08	4.950E+00
	OB-C-ALO	**1.490E+00**	**7.700E-01**	**9.980E-01**	**3.970E+00**
$F15$	ALO	4.180E-03	7.370E-03	5.550E-04	**2.040E-02**
	OB-C-ALO	3.380E-03	6.790E-03	3.080E-04	2.050E-02
$F16$	ALO	-1.030E+00	6.780E-16	-1.030E+00	-1.030E+00
	OB-C-ALO	**-1.030E+00**	**6.780E-16**	**-1.030E+00**	**-1.030E+00**
$F17$	ALO	3.980E-01	1.690E-16	3.980E-01	3.980E-01
	OB-C-ALO	**3.980E-01**	**1.690E-16**	**3.980E-01**	**3.980E-01**
$F18$	ALO	3.000E+00	0.000E+00	3.000E+00	3.000E+00
	OB-C-ALO	**3.000E+00**	**0.000E+00**	**3.000E+00**	**3.000E+00**
$F19$	ALO	-6.280E+00	2.930E+00	-1.020E+01	-2.630E+00
	OB-C-ALO	**-9.230E+00**	**2.140E+00**	**-1.020E+01**	**-2.630E+00**
$F20$	ALO	-6.670E+00	3.420E+00	-1.040E+01	-1.840E+00
	OB-C-ALO	**-8.720E+00**	**2.860E+00**	**-1.040E+01**	**-3.720E+00**
$F21$	ALO	-6.270E+00	3.660E+00	-1.050E+01	-1.860E+00
	OB-C-ALO	**-9.190E+00**	**2.780E+00**	**-1.050E+01**	**-2.420E+00**
$F22$	ALO	-3.863E+00	-3.863E+00	3.160E-15	-3.863E+00
	OB-C-ALO	**-3.863E+00**	**-3.863E+00**	**3.160E-15**	**-3.863E+00**
$F23$	ALO	-3.322E+00	-3.322E+00	1.810E-15	-3.322E+00
	OB-C-ALO	**-3.322E+00**	**-3.322E+00**	**1.810E-15**	**-3.322E+00**

Table 5.7: Average, Standard Deviation, Minimum, and Maximum of objective function values of composition benchmark functions

Function	*Methods*	*Ave.*	*Std. Dev.*	*Min*	*Max*
$F24$	**OB-C-ALO**	**2.500E+03**	**0.000E+00**	**2.500E+03**	**2.500E+03**
	ALO	2.630E+03	1.275E-03	2.630E+03	2.630E+03
$F25$	OB-C-ALO	2.545E+03	2.516E+03	2.600E+03	2.416E+01
	ALO	2.538E+03	2.512E+03	2.577E+03	1.477E+01
$F26$	OB-C-ALO	2.693E+03	1.398E+01	2.647E+03	2.700E+03
	ALO	2.680E+03	2.430E+01	2.630E+03	2.700E+03
$F27$	OB-C-ALO	2.700E+03	1.142E-01	2.700E+03	2.701E+03
	ALO	2.700E+03	1.090E-01	2.700E+03	2.700E+03
$F28$	**OB-C-ALO**	**2.865E+03**	**7.594E+01**	**2.702E+03**	**2.900E+03**
	ALO	3.010E+03	1.440E+02	2.700E+03	**3.110E+03**
$F29$	**OB-C-ALO**	3.000E+03	**0.000E+00**	**3.000E+03**	**3.000E+03**
	ALO	3.490E+03	2.920E+02	3.170E+03	3.930E+03
$F30$	**OB-C-ALO**	**1.387E+05**	**4.681E+05**	**3.100E+03**	**4.681E+05**
	ALO	8.270E+05	1.520E+06	3.100E+03	3.670E+06
$F31$	**OB-C-ALO**	**3.823E+03**	**2.735E+02**	**3.500E+03**	**4.641E+03**
	ALO	4.420E+03	4.140E+02	3.880E+03	6.170E+03

Table 5.8: P-values of Wilcoxon ranksum test

Function	OB-C-ALO vs ALO	
	p-values	conclusion
$F1$	1.21E-12	+
$F2$	1.21E-12	+
$F3$	1.21E-12	+
$F4$	1.21E-12	+
$F5$	3.02E-11	+
$F6$	2.43E-05	-
$F7$	3.02E-11	+
$F8$	4.24E-12	+
$F9$	1.20E-12	+
$F10$	1.18E-12	+
$F11$	1.21E-12	+
$F12$	3.01E-11	+
$F13$	5.52E-08	+
$F14$	3.18E-01	-
$F15$	2.64E-01	-
$F16$	N/A	=
$F17$	N/A	=
$F18$	N/A	=
$F19$	6.10E-03	+
$F20$	9.30E-01	-
$F21$	3.74E-02	+
$F22$	N/A	=
$F23$	N/A	=
$F24$	1.36E-20	+
$F25$	3.15E-01	-
$F26$	5.20E-03	-
$F27$	9.97E-02	+
$F28$	2.71E-10	+
$F29$	1.39E-20	+
$F30$	8.95E-05	+
$F31$	3.64E-13	+
25/31(20 ‘+’ , 5 ‘=’)		

Table 5.9: Comparison of results on unimodal functions (10D)

Function	***Methods***	***Ave.***	***Std. Dev.***	***Min***	***Max***
$F1$	**OB-C-ALO**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-LF-ALO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-ac-ALO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-SAC-ALO	7.330E-33	0.000E+00	2.680E-32	1.130E-31
	OB-L-ALO	1.090E-11	0.000E+00	3.860E-11	2.050E-10
	ALO	7.870E-09	2.390E-09	5.490E-09	**2.580E-08**
$F2$	**OB-C-ALO**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-LF-ALO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-ac-ALO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-SAC-ALO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-L-ALO	1.850E-18	0.000E+00	1.010E-17	5.550E-17
	ALO	4.860E-01	1.550E-05	8.850E-01	**2.800E+00**
$F3$	**OB-C-ALO**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-LF-ALO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-ac-ALO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-SAC-ALO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-L-ALO	1.730E-32	0.000E+00	6.950E-32	3.450E-31
	ALO	8.310E-02	1.360E-04	1.850E-01	**8.750E-01**
$F4$	**OB-C-ALO**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-LF-ALO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-ac-ALO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-SAC-ALO	1.160E-15	2.230E-21	3.270E-15	1.730E-14
	OB-L-ALO	4.090E-06	0.000E+00	9.620E-06	3.220E-05
	ALO	3.180E-03	1.020E-04	6.160E-03	**3.290E-02**
$F5$	**OB-C-ALO**	2.040E-04	1.370E-07	3.910E-04	1.830E-03
	OB-LF-ALO	6.2900E-02	2.9200E-05	1.0700E-01	4.0300E-01
	OB-ac-ALO	**2.270E-04**	**1.410E-03**	**3.680E-04**	**2.400E-07**
	OB-SAC-ALO	4.080E-04	7.610E-07	6.500E-04	3.040E-03
	OB-L-ALO	5.460E-04	1.250E-07	1.600E-03	8.630E-03
	ALO	6.810E+01	1.020E-04	1.990E+02	**1.070E+03**
$F6$	OB-C-ALO	4.580E-08	2.870E-09	5.100E-08	2.140E-07
	OB-LF-ALO	1.200E-03	5.900E-06	1.290E-03	5.100E-03
	OB-ac-ALO	2.920E-08	5.320E-09	4.920E-08	2.640E-07
	OB-SAC-ALO	1.250E-08	3.240E-09	1.610E-08	8.530E-08
	OB-L-ALO	3.320E-08	7.360E-09	3.000E-08	**1.160E-07**
	ALO	**8.390E-09**	**2.150E-09**	**5.220E-09**	2.220E-08
$F7$	**OB-C-ALO**	**2.660E-04**	**9.950E-06**	**2.200E-04**	**8.540E-04**
	OB-LF-ALO	3.1100E-04	1.6700E-05	4.1500E-04	1.9900E-03
	OB-ac-ALO	**2.620E-04**	**6.980E-04**	**1.920E-04**	**9.880E-06**
	OB-SAC-ALO	4.030E-04	1.200E-05	3.930E-04	1.710E-03
	OB-L-ALO	2.850E-04	3.890E-05	2.490E-04	1.330E-03
	ALO	2.210E-02	1.810E-03	1.230E-02	**5.630E-02**

Table 5.10: Comparison of results on unimodal functions (30D)

Function	***Methods***	***Ave.***	***Std. Dev.***	***Min***	***Max***
$F1$	**OB-C-ALO**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-LF-ALO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-ac-ALO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-SAC-ALO	2.280E-54	0.000E+00	1.250E-53	6.830E-53
	OB-L-ALO	1.010E-09	0.000E+00	4.630E-09	2.530E-08
	ALO	9.450E-06	1.720E-06	5.540E-06	2.110E-05
$F2$	**OB-C-ALO**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-LF-ALO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-ac-ALO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-SAC-ALO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-L-ALO	5.980E-05	0.000E+00	4.260E-05	1.090E-04
	ALO	3.650E+01	2.090E-02	5.170E+01	1.340E+02
$F3$	**OB-C-ALO**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-LF-ALO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-ac-ALO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-SAC-ALO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-L-ALO	1.160E-29	0.000E+00	5.950E-29	3.260E-28
	ALO	9.550E+02	2.470E+02	5.080E+02	2.160E+03
$F4$	**OB-C-ALO**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-LF-ALO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-ac-ALO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-SAC-ALO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-L-ALO	1.090E-05	0.000E+00	2.290E-05	7.130E-05
	ALO	1.210E+01	4.630E+00	3.650E+00	1.960E+01
$F5$	OB-C-ALO	3.070E-03	1.580E-06	4.720E-03	1.950E-02
	OB-LF-ALO	9.590E-03	2.310E-04	1.280E-02	4.740E-02
	OB-ac-ALO	**5.530E-04**	**2.950E-03**	**6.800E-04**	**1.160E-05**
	OB-SAC-ALO	2.090E-03	1.990E-04	1.930E-03	8.050E-03
	OB-L-ALO	5.460E-03	5.540E-06	7.170E-03	3.220E-02
	ALO	1.040E+02	1.640E+01	3.120E+02	1.740E+03
$F6$	OB-C-ALO	2.430E-05	2.860E-06	1.650E-05	6.450E-05
	OB-LF-ALO	5.860E-04	3.820E-05	5.890E-04	2.200E-03
	OB-ac-ALO	1.820E-05	7.350E-05	1.600E-05	1.140E-06
	OB-SAC-ALO	1.760E-05	3.160E-06	1.030E-05	4.360E-05
	OB-L-ALO	3.230E-05	7.980E-06	1.840E-05	7.620E-05
	ALO	**9.490E-06**	**1.320E-06**	**6.100E-06**	**2.920E-05**
$F7$	OB-C-ALO	2.160E-04	7.520E-06	1.880E-04	6.610E-04
	OB-LF-ALO	**9.730E-05**	**1.480E-06**	**8.690E-05**	**3.090E-04**
	OB-ac-ALO	1.670E-04	3.760E-04	1.020E-04	7.520E-06
	OB-SAC-ALO	2.710E-04	1.500E-05	2.670E-04	1.050E-03
	OB-L-ALO	2.780E-04	9.410E-06	2.530E-04	1.240E-03
	ALO	1.100E-01	5.700E-02	3.570E-02	2.250E-01

Table 5.11: Comparison of results on multimodal functions (10D)

Function	***Methods***	***Ave.***	***Std. Dev.***	***Min***	***Max***
$F8$	**OB-C-ALO**	**-2.970E+03**	**-3.970E+03**	**6.230E+02**	**-2.090E+03**
	OB-LF-ALO	-1.3400E+03	-1.3600E+03	8.6100E+00	-1.3300E+03
	OB-ac-ALO	-2.880E+03	-1.970E+03	6.160E+02	-3.970E+03
	OB-SAC-ALO	-2.430E+03	-3.040E+03	2.820E+02	-1.970E+03
	OB-L-ALO	-2.450E+03	-3.320E+03	4.230E+02	-1.920E+03
	ALO	-2.470E+03	-3.730E+03	4.260E+02	-1.920E+03
$F9$	**OB-C-ALO**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-LF-ALO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-ac-ALO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-SAC-ALO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-L-ALO	5.070E-10	0.000E+00	6.450E-10	2.240E-09
	ALO	2.480E+01	6.960E+00	1.080E+01	**5.070E+01**
$F10$	**OB-C-ALO**	**8.880E-16**	**8.880E-16**	**4.010E-31**	**8.880E-16**
	OB-LF-ALO	**8.8800E-16**	**8.8800E-16**	**4.0100E-31**	**8.8800E-16**
	OB-ac-ALO	**8.880E-16**	**8.880E-16**	**4.010E-31**	**8.880E-16**
	OB-SAC-ALO	1.100E-14	4.440E-15	1.110E-14	4.000E-14
	OB-L-ALO	1.540E-05	8.880E-16	8.100E-06	2.850E-05
	ALO	3.240E-01	2.070E-05	5.540E-01	**1.650E+00**
$F11$	**OB-C-ALO**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-LF-ALO	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	OB-ac-ALO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-SAC-ALO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-L-ALO	7.210E-10	0.000E+00	2.110E-09	9.600E-09
	ALO	2.140E-01	6.640E-02	8.720E-02	**4.060E-01**
$F12$	**OB-C-ALO**	2.91E-08	1.55E-09	3.48E-08	1.73E-07
	OB-LF-ALO	1.3000E-03	2.3000E-09	2.2500E-03	9.1300E-03
	OB-ac-ALO	2.460E-08	**2.100E-02**	2.300E-08	1.250E-09
	OB-SAC-ALO	**1.490E-08**	**7.020E-10**	**1.410E-08**	**5.600E-08**
	OB-L-ALO	1.630E-08	4.710E-32	2.410E-08	1.230E-07
	ALO	1.790E+00	8.230E-09	1.670E+00	**5.610E+00**
$F13$	OB-C-ALO	2.930E-03	3.240E-09	4.940E-03	1.100E-02
	OB-LF-ALO	5.7200E-03	1.5800E-06	1.2800E-02	6.3500E-02
	OB-ac-ALO	1.470E-03	1.100E-02	3.800E-03	1.070E-08
	OB-SAC-ALO	2.870E-03	1.340E-08	6.220E-03	2.100E-02
	OB-L-ALO	**6.820E-08**	**1.350E-32**	**9.610E-08**	3.800E-07
	ALO	1.800E-03	3.840E-09	4.940E-03	2.100E-02

Table 5.12: Comparison of results on multimodal functions (30D)

Function	***Methods***	***Ave.***	***Std. Dev.***	***Min***	***Max***
$F8$	OB-C-ALO	-1.180E+04	-1.260E+04	1.420E+03	-8.440E+03
	OB-LF-ALO	-1.080E+04	-1.2600E+04	3.7800E+03	-2.1300E+03
	OB-ac-ALO	**-1.250E+04**	**-1.240E+04**	**9.340E+01**	**-1.260E+04**
	OB-SAC-ALO	-6.490E+03	-6.540E+03	7.260E+01	-6.270E+03
	OB-*L*-ALO	-5.980E+03	-1.260E+04	1.710E+03	-5.420E+03
	ALO	-5.440E+03	-5.540E+03	4.190E+01	-5.420E+03
$F9$	**OB-C-ALO**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-LF-ALO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-ac-ALO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-SAC-ALO	1.890E-15	0.000E+00	1.040E-14	5.680E-14
	OB-*L*-ALO	3.010E-09	0.000E+00	4.820E-09	1.570E-08
	ALO	1.970E+00	5.890E-04	8.390E-01	3.520E+00
$F10$	**OB-C-ALO**	**8.880E-16**	**8.880E-16**	**4.010E-31**	**8.880E-16**
	OB-LF-ALO	**8.880E-16**	**8.8800E-16**	**4.0100E-31**	**8.8800E-16**
	OB-ac-ALO	**8.880E-16**	**8.880E-16**	**4.010E-31**	**8.880E-16**
	OB-SAC-ALO	4.440E-15	4.440E-15	4.010E-30	4.440E-15
	OB-*L*-ALO	2.100E-05	8.880E-16	1.460E-05	4.200E-05
	ALO	1.870E+00	1.020E-03	6.900E-01	3.090E+00
$F11$	**OB-C-ALO**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-LF-ALO	**0.000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	OB-ac-ALO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-SAC-ALO	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	OB-*L*-ALO	6.320E-09	0.000E+00	1.270E-08	5.220E-08
	ALO	1.190E-02	3.470E-04	1.260E-02	3.860E-02
$F12$	OB-C-ALO	1.090E-06	3.560E-07	6.340E-07	3.560E-06
	OB-LF-ALO	3.200E-05	1.4600E-08	8.4100E-05	4.5400E-04
	OB-ac-ALO	8.360E-07	3.370E-06	6.410E-07	1.780E-07
	OB-SAC-ALO	**6.650E-07**	**1.470E-07**	**3.070E-07**	**1.380E-06**
	OB-*L*-ALO	6.440E-07	7.190E-08	5.810E-07	2.100E-06
	ALO	1.000E+01	5.750E+00	4.340E+00	2.330E+01
$F13$	OB-C-ALO	2.550E-03	4.310E-06	5.430E-03	2.100E-02
	OB-LF-ALO	1.890E-04	1.7200E-07	3.9000E-04	1.9400E-03
	OB-ac-ALO	2.870E-03	3.090E-02	6.720E-03	3.830E-06
	OB-SAC-ALO	1.480E-03	2.100E-06	3.810E-03	1.100E-02
	OB-*L*-ALO	**6.980E-06**	**1.350E-32**	**5.330E-06**	**2.050E-05**
	ALO	1.320E+00	1.140E-05	3.150E+00	1.050E+01

Table 5.13: Comparison of results on fixed dimension multimodal benchmark functions (Continued…)

Function	***Methods***	***Ave.***	***Std. Dev.***	***Min***	***Max***
$F14$	OB-C-ALO	1.490E+00	9.980E-01	7.700E-01	3.970E+00
	OB-LF-ALO	**1.300E+00**	**9.9800E-01**	**5.3100E-01**	**2.9800E+00**
	OB-ac-ALO	1.860E+00	6.900E+00	1.310E+00	9.980E-01
	OB-SAC-ALO	1.490E+00	9.980E-01	7.700E-01	3.970E+00
	OB-L-ALO	1.820E+00	9.980E-01	1.080E+00	4.950E+00
	ALO	1.790E+00	1.950E-08	1.340E+00	4.950E+00
$F15$	OB-C-ALO	3.380E-03	3.080E-04	6.790E-03	2.050E-02
	OB-LF-ALO	1.510E-03	5.100E-04	3.570E-03	2.040E-02
	OB-ac-ALO	3.380E-03	2.050E-02	6.790E-03	5.980E-04
	OB-SAC-ALO	3.380E-03	3.080E-04	6.790E-03	2.050E-02
	OB-L-ALO	**8.210E-04**	**4.190E-04**	**2.050E-04**	**1.210E-03**
	ALO	4.180E-03	5.550E-04	7.370E-03	**2.040E-02**
$F16$	OB-C-ALO	-1.030E+00	-1.030E+00	6.780E-16	-1.030E+00
	OB-LF-ALO	-1.030E+00	-1.030E+00	6.780E-16	-1.030E+00
	OB-ac-ALO	-1.030E+00	-1.030E+00	6.780E-16	-1.030E+00
	OB-SAC-ALO	-1.030E+00	-1.030E+00	6.780E-16	-1.030E+00
	OB-L-ALO	-1.030E+00	-1.030E+00	6.780E-16	-1.030E+00
	ALO	-1.030E+00	-1.030E+00	6.780E-16	-1.030E+00
$F17$	OB-C-ALO	3.980E-01	3.980E-01	1.690E-16	3.980E-01
	OB-LF-ALO	3.980E-01	3.980E-01	1.690E-16	3.980E-01
	OB-ac-ALO	3.980E-01	3.980E-01	1.690E-16	3.980E-01
	OB-SAC-ALO	3.980E-01	3.980E-01	1.690E-16	3.980E-01
	OB-L-ALO	3.980E-01	3.980E-01	1.690E-16	3.980E-01
	ALO	3.980E-01	3.980E-01	1.690E-16	3.980E-01
$F18$	OB-C-ALO	3.000E+00	3.000E+00	0.000E+00	3.000E+00
	OB-LF-ALO	3.000E+00	3.000E+00	0.000E+00	3.000E+00
	OB-ac-ALO	3.000E+00	3.000E+00	0.000E+00	3.000E+00
	OB-SAC-ALO	3.000E+00	3.000E+00	0.000E+00	3.000E+00
	OB-L-ALO	3.000E+00	3.000E+00	0.000E+00	3.000E+00
	ALO	3.000E+00	3.000E+00	0.000E+00	3.000E+00
$F19$	**OB-C-ALO**	**-9.230E+00**	**-1.020E+01**	**2.140E+00**	**-2.630E+00**
	OB-LF-ALO	-7.790E+00	-1.0200E+01	2.8100E+00	-2.6300E+00
	OB-ac-ALO	-7.540E+00	-2.630E+00	2.690E+00	-1.020E+01
	OB-SAC-ALO	-8.970E+00	-1.020E+01	2.170E+00	-5.100E+00
	OB-L-ALO	**-9.900E+00**	**-1.020E+01**	**1.370E+00**	**-2.630E+00**
	ALO	-6.280E+00	-1.020E+01	2.930E+00	-2.630E+00
$F20$	OB-C-ALO	-8.720E+00	-1.040E+01	2.860E+00	-3.720E+00
	OB-LF-ALO	-8.520E+00	-1.040E+01	2.970E+00	-2.770E+00
	OB-ac-ALO	-7.900E+00	-2.770E+00	2.960E+00	-1.040E+01
	OB-SAC-ALO	-8.950E+00	-1.040E+01	2.710E+00	-2.770E+00
	OB-L-ALO	**-1.040E+01**	**-1.040E+01**	**7.230E-15**	**-1.040E+01**
	ALO	-6.670E+00	-1.040E+01	3.420E+00	**-1.840E+00**

Table 5.13: (Continued)

$F21$	OB-C-ALO	-9.190E+00	-1.050E+01	2.780E+00	-2.420E+00
	OB-LF-ALO	-7.100E+00	-1.050E+01	3.580E+00	-2.420E+00
	OB-ac-ALO	-7.470E+00	-2.420E+00	3.410E+00	-1.050E+01
	OB-SAC-ALO	-8.080E+00	-1.050E+01	3.350E+00	-2.420E+00
	OB-L-ALO	**-1.050E+01**	**-1.050E+01**	**9.030E-15**	**-1.050E+01**
	ALO	-6.270E+00	-1.050E+01	3.660E+00	-1.860E+00
$F22$	OB-C-ALO	**-3.863E+00**	**-3.863E+00**	**3.160E-15**	**-3.863E+00**
	OB-LF-ALO	-3.863E+00	-3.863E+00	3.160E-15	-3.863E+00
	OB-ac-ALO	**-3.863E+00**	**-3.863E+00**	**3.160E-15**	**-3.863E+00**
	OB-SAC-ALO	-3.863E+00	-3.863E+00	3.160E-15	-3.863E+00
	OB-L-ALO	**-3.863E+00**	**-3.863E+00**	**3.160E-15**	**-3.863E+00**
	ALO	-3.863E+00	-3.863E+00	3.160E-15	-3.863E+00
$F23$	OB-C-ALO	-3.322E+00	-3.322E+00	1.810E-15	-3.322E+00
	OB-LF-ALO	-3.322E+00	-3.322E+00	1.810E-15	-3.322E+00
	OB-ac-ALO	-3.322E+00	-3.322E+00	1.810E-15	-3.322E+00
	OB-SAC-ALO	-3.322E+00	-3.322E+00	1.810E-15	-3.322E+00
	OB-L-ALO	-3.322E+00	-3.322E+00	1.810E-15	-3.322E+00
	ALO	-3.322E+00	-3.322E+00	1.810E-15	-3.322E+00

Table 5.14: Comparison of results on composition functions

Function	***Methods***	***Ave.***	***Stan. Dev.***	***Min.***	***Max.***
$F24$	**OB-C-ALO**	**2.500E+03**	**0.000E+00**	**2.500E+03**	**2.500E+03**
	OB-LF-ALO	**2.500E+03**	**0.000E+00**	**2.500E+03**	**2.500E+03**
	OB-ac-ALO	**2.500E+03**	**0.000E+00**	**2.500E+03**	**2.500E+03**
	OB-SAC-ALO	2.630E+03	3.270E-04	2.630E+03	2.630E+03
	OB-*L*-ALO	2.546E+03	6.248E+01	2.500E+03	2.630E+03
	ALO	2.630E+03	1.275E-03	2.630E+03	2.630E+03
$F25$	OB-C-ALO	2.545E+03	2.516E+03	2.600E+03	2.416E+01
	OB-LF-ALO	2.600E+03	7.066E-01	2.596E+03	2.600E+03
	OB-ac-ALO	2.540E+03	2.017E+01	2.518E+03	2.600E+03
	OB-SAC-ALO	**2.536E+03**	**1.312E+01**	**2.518E+03**	**2.580E+03**
	OB-*L*-ALO	2.690E+03	**1.470E+01**	2.650E+03	2.700E+03
	ALO	2.538E+03	2.512E+03	2.577E+03	1.477E+01
$F26$	OB-C-ALO	2.693E+03	1.398E+01	2.647E+03	2.700E+03
	OB-LF-ALO	2.700E+03	4.349E-01	2.697E+03	2.700E+03
	OB-ac-ALO	2.690E+03	2.100E+01	2.620E+03	2.700E+03
	OB-SAC-ALO	**2.680E+03**	**2.320E+01**	**2.620E+03**	**2.700E+03**
	OB-*L*-ALO	2.690E+03	**1.470E+01**	2.650E+03	2.700E+03
	ALO	2.680E+03	2.430E+01	2.630E+03	2.700E+03
$F27$	**OB-C-ALO**	**2.700E+03**	**1.142E-01**	**2.700E+03**	**2.701E+03**
	OB-LF-ALO	2.705E+03	9.557E-02	2.705E+03	2.705E+03
	OB-ac-ALO	2.700E+03	1.100E-01	2.700E+03	2.700E+03
	OB-SAC-ALO	2.700E+03	1.040E-01	2.700E+03	2.700E+03
	OB-*L*-ALO	2.700E+03	**1.030E-01**	2.700E+03	2.700E+03
	ALO	2.700E+03	1.090E-01	2.700E+03	2.700E+03
$F28$	OB-C-ALO	2.865E+03	7.594E+01	2.702E+03	2.900E+03
	OB-LF-ALO	2.900E+03	0.000E+00	2.900E+03	2.900E+03
	OB-ac-ALO	**2.840E+03**	**9.170E+01**	**2.700E+03**	**2.900E+03**
	OB-SAC-ALO	2.970E+03	1.720E+02	2.700E+03	3.100E+03
	OB-*L*-ALO	2.940E+03	1.640E+02	2.700E+03	3.100E+03
	ALO	3.010E+03	1.440E+02	2.700E+03	**3.110E+03**
$F29$	**OB-C-ALO**	**3.000E+03**	**0.000E+00**	**3.000E+03**	**3.000E+03**
	OB-LF-ALO	**3.000E+03**	**0.000E+00**	**3.000E+03**	**3.000E+03**
	OB-ac-ALO	**3.000E+03**	**0.000E+00**	**3.000E+03**	**3.000E+03**
	OB-SAC-ALO	3.480E+03	2.950E+02	3.160E+03	3.890E+03
	OB-*L*-ALO	**3.220E+03**	**1.230E+02**	**3.000E+03**	3.710E+03
	ALO	3.490E+03	2.920E+02	3.170E+03	3.930E+03
$F30$	**OB-C-ALO**	**1.387E+05**	**4.681E+05**	**3.100E+03**	**4.681E+05**
	OB-LF-ALO	4.224E+06	1.406E+06	3.100E+03	4.755E+06
	OB-ac-ALO	7.220E+05	1.470E+06	3.100E+03	3.670E+06
	OB-SAC-ALO	7.220E+05	1.470E+06	3.100E+03	3.670E+06
	OB-*L*-ALO	1.050E+05	4.100E+05	3.100E+03	1.730E+06
	ALO	8.270E+05	1.520E+06	3.100E+03	3.670E+06
$F31$	**OB-C-ALO**	**3.823E+03**	**2.735E+02**	**3.500E+03**	**4.641E+03**
	OB-LF-ALO	5.623E+03	5.444E+02	5.406E+03	7.653E+03
	OB-ac-ALO	4.460E+03	3.980E+02	3.880E+03	6.190E+03
	OB-SAC-ALO	4.460E+03	3.980E+02	3.880E+03	6.190E+03
	OB-*L*-ALO	3.790E+03	2.530E+02	3.500E+03	4.470E+03
	ALO	4.420E+03	4.140E+02	3.880E+03	6.170E+03

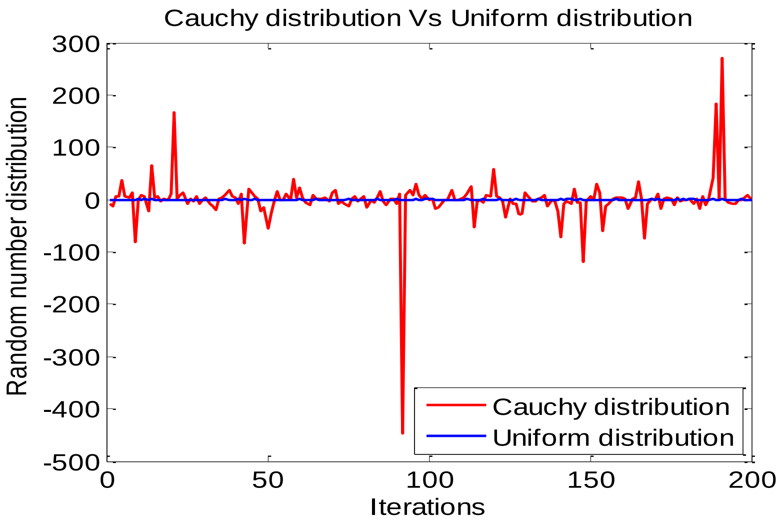


Figure 5.1: Generation of random numbers using Uniform and Cauchy distribution

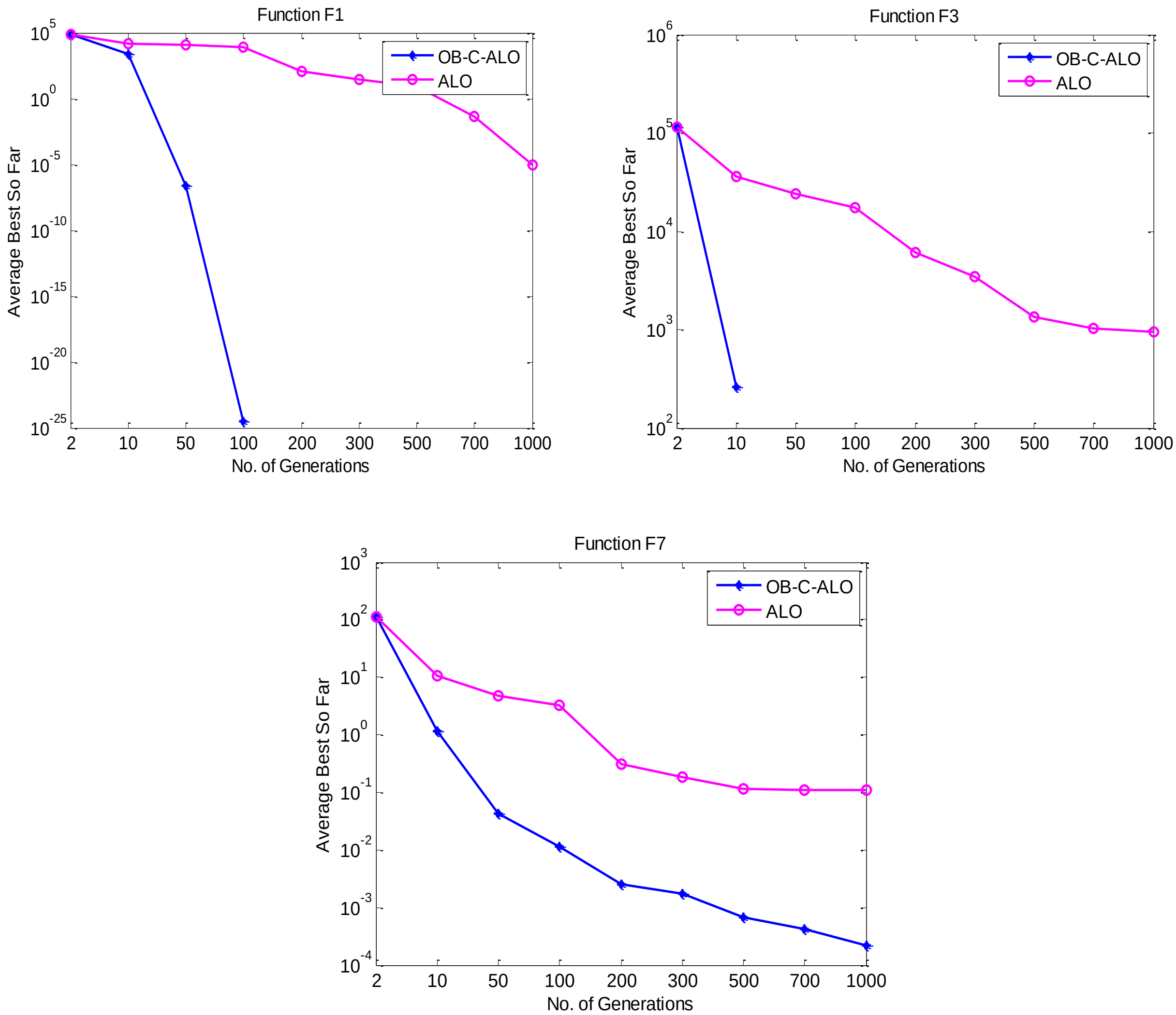


Figure 5.2**:** Convergence behaviour on three of the unimodal functions $F1, F3$ and $F7$

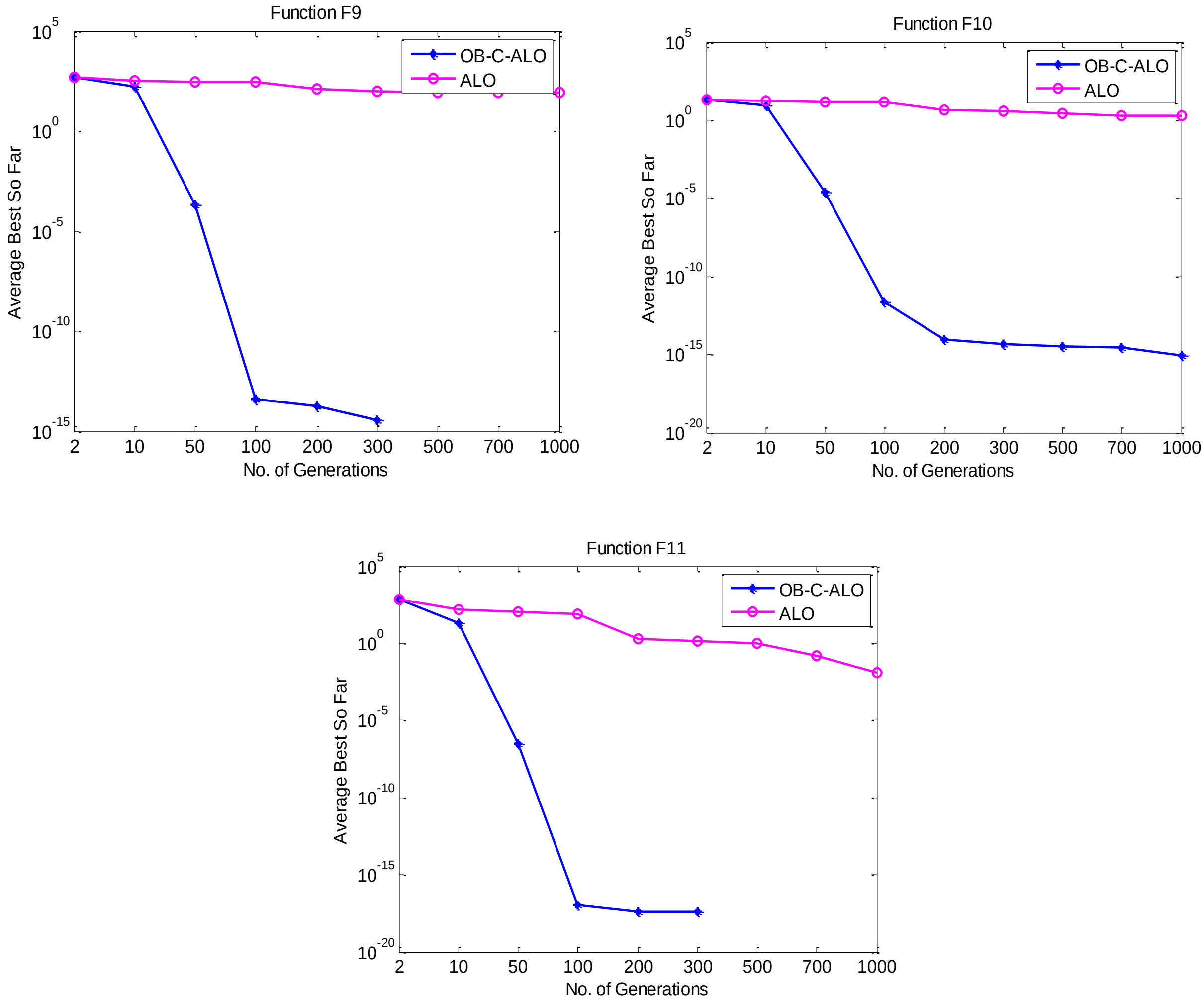


Figure 5.3: Convergence behaviour on three multimodal functions $F9, F10$ and $F11$

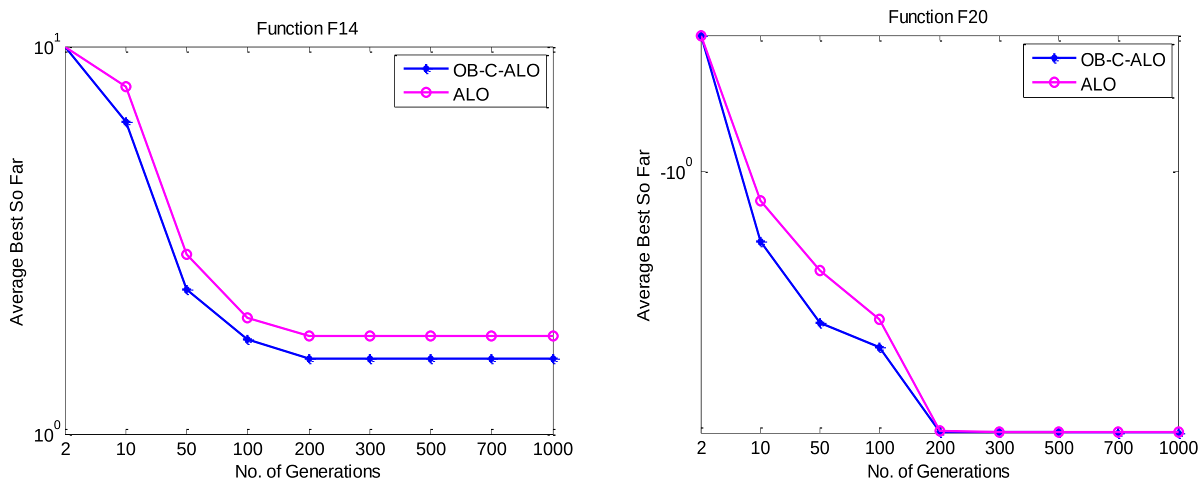


Figure 5.4: Convergence behaviour on two fixed dimension multimodal functions $F14$ and $F20$

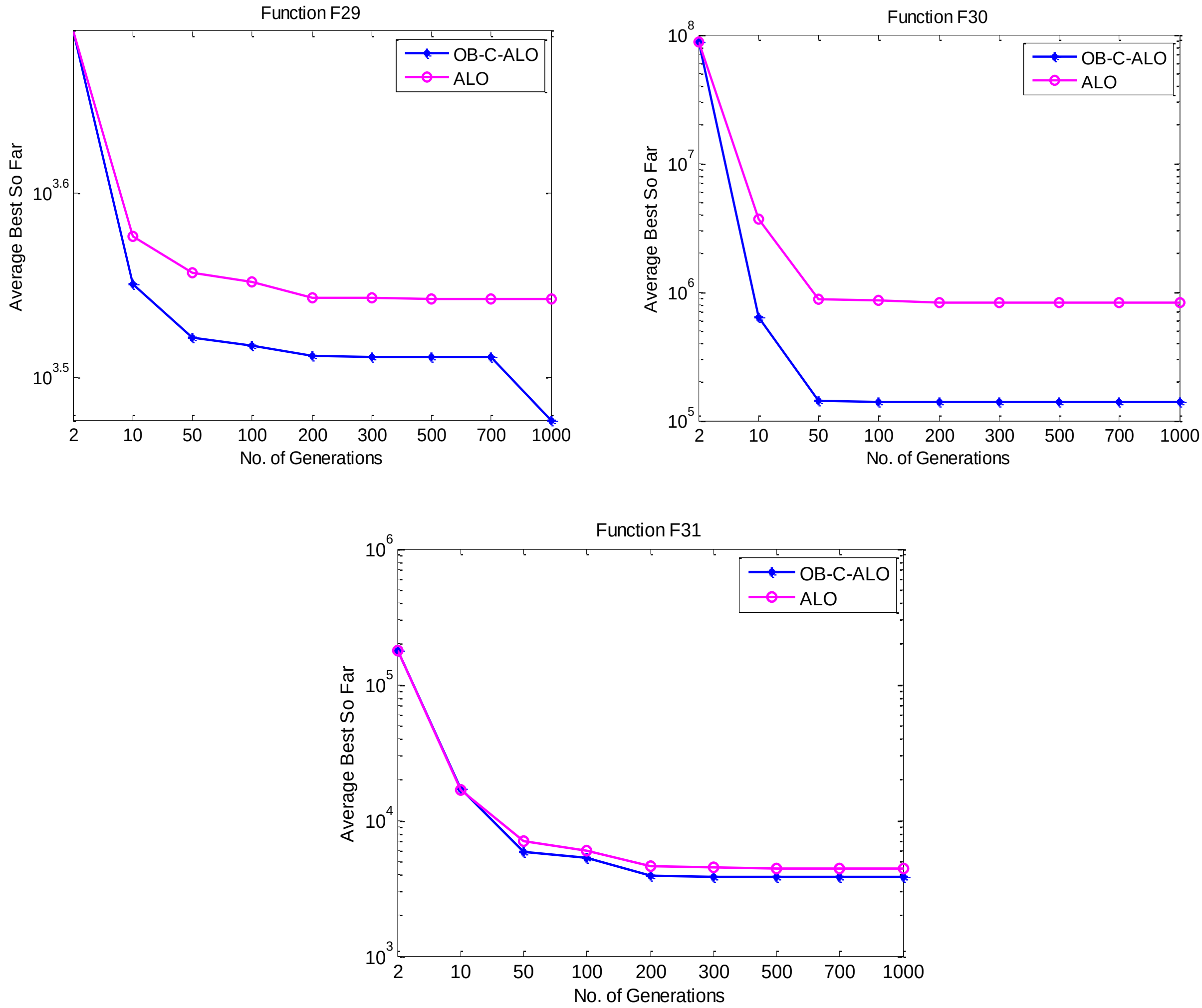


Figure 5.5: Convergence behaviour on three composition functions $F29$,$F30$ and $F31$

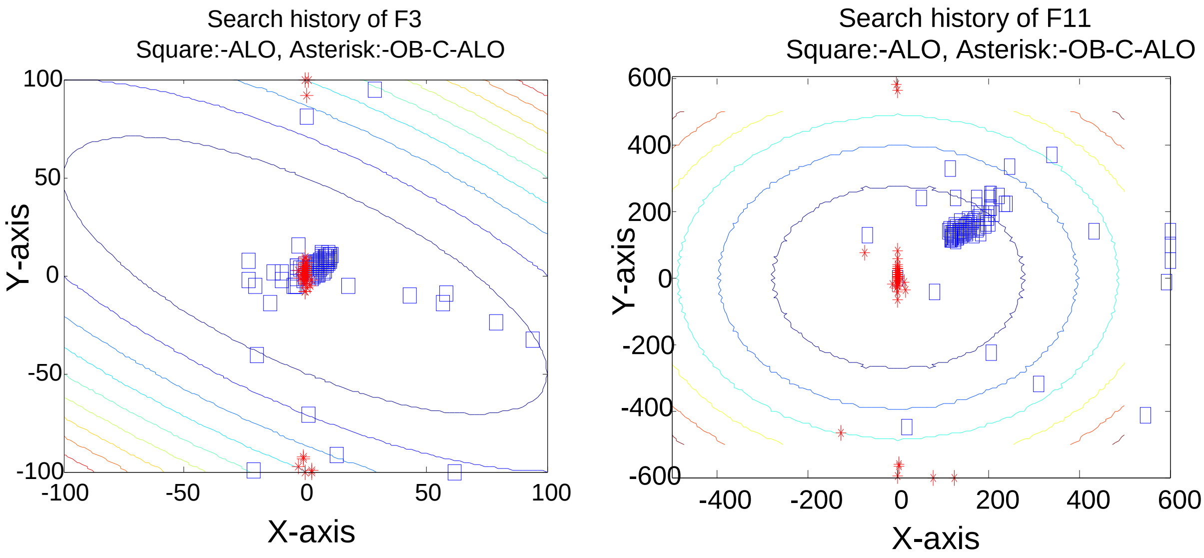


Figure 5.6: Search history analysis of functions $F3$ and $F11$

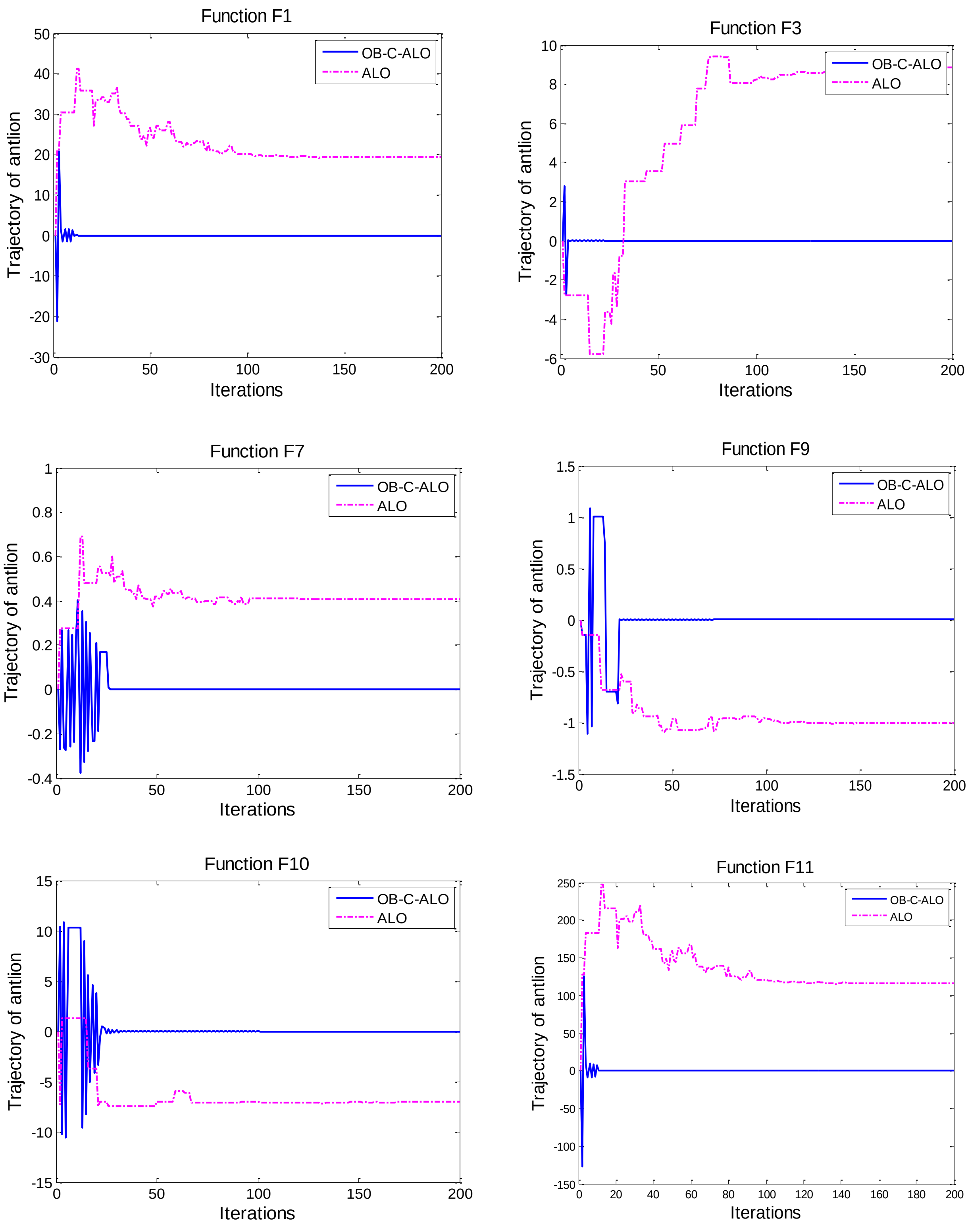


Figure 5.7: Trajectories of elite antlion

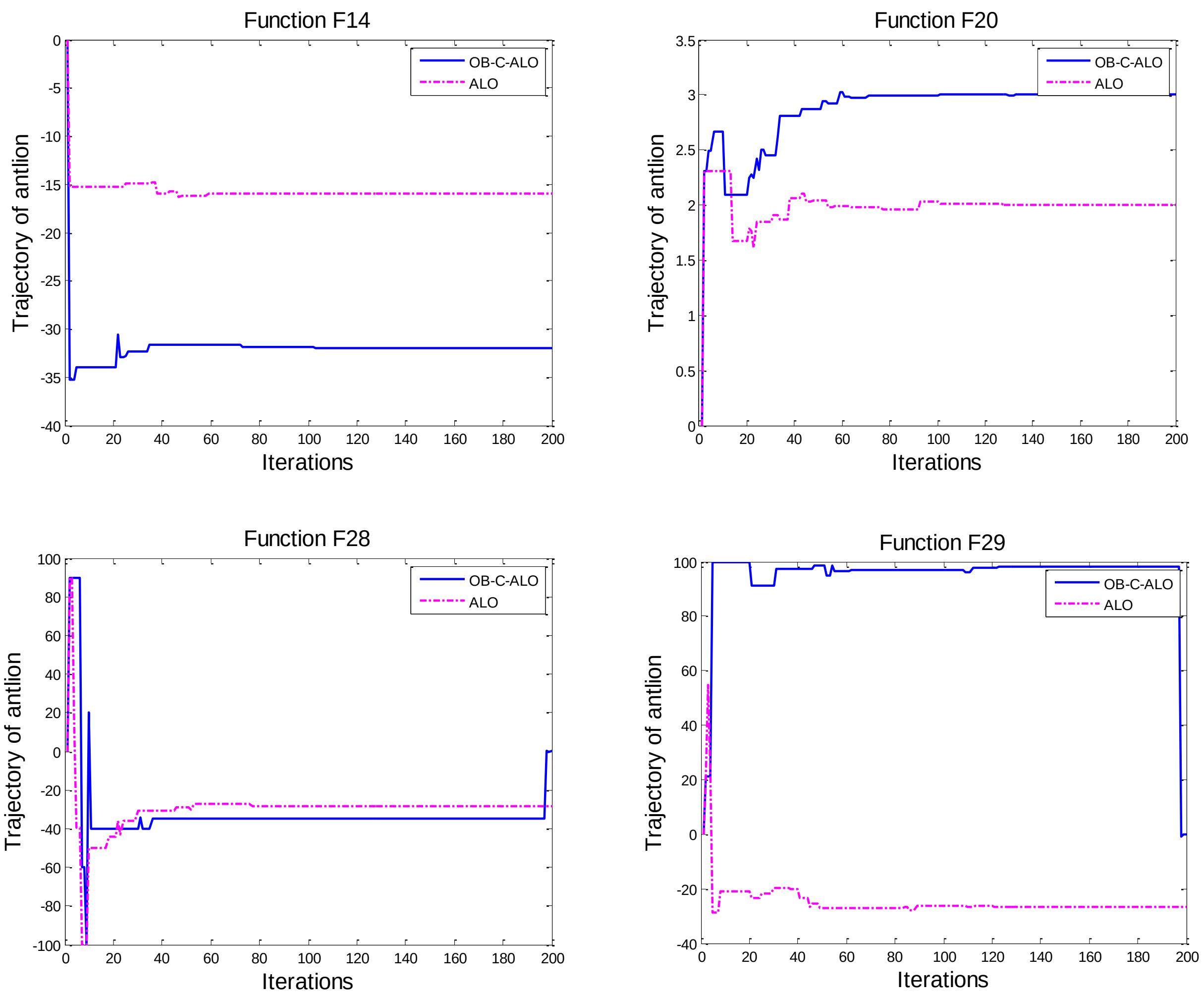


Figure 5.7: (Continued)

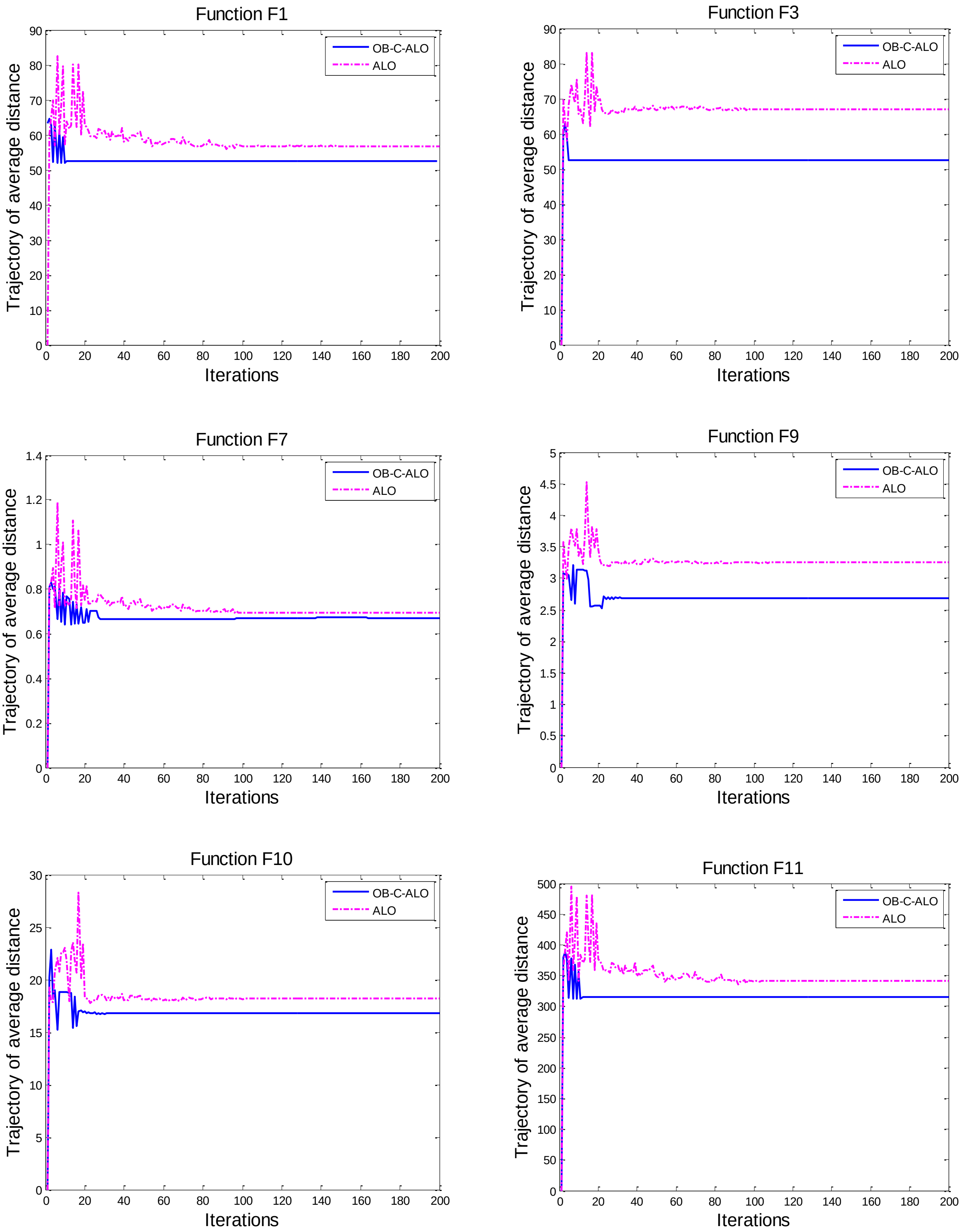


Figure 5.8: Trajectory analysis of average distance between search agents

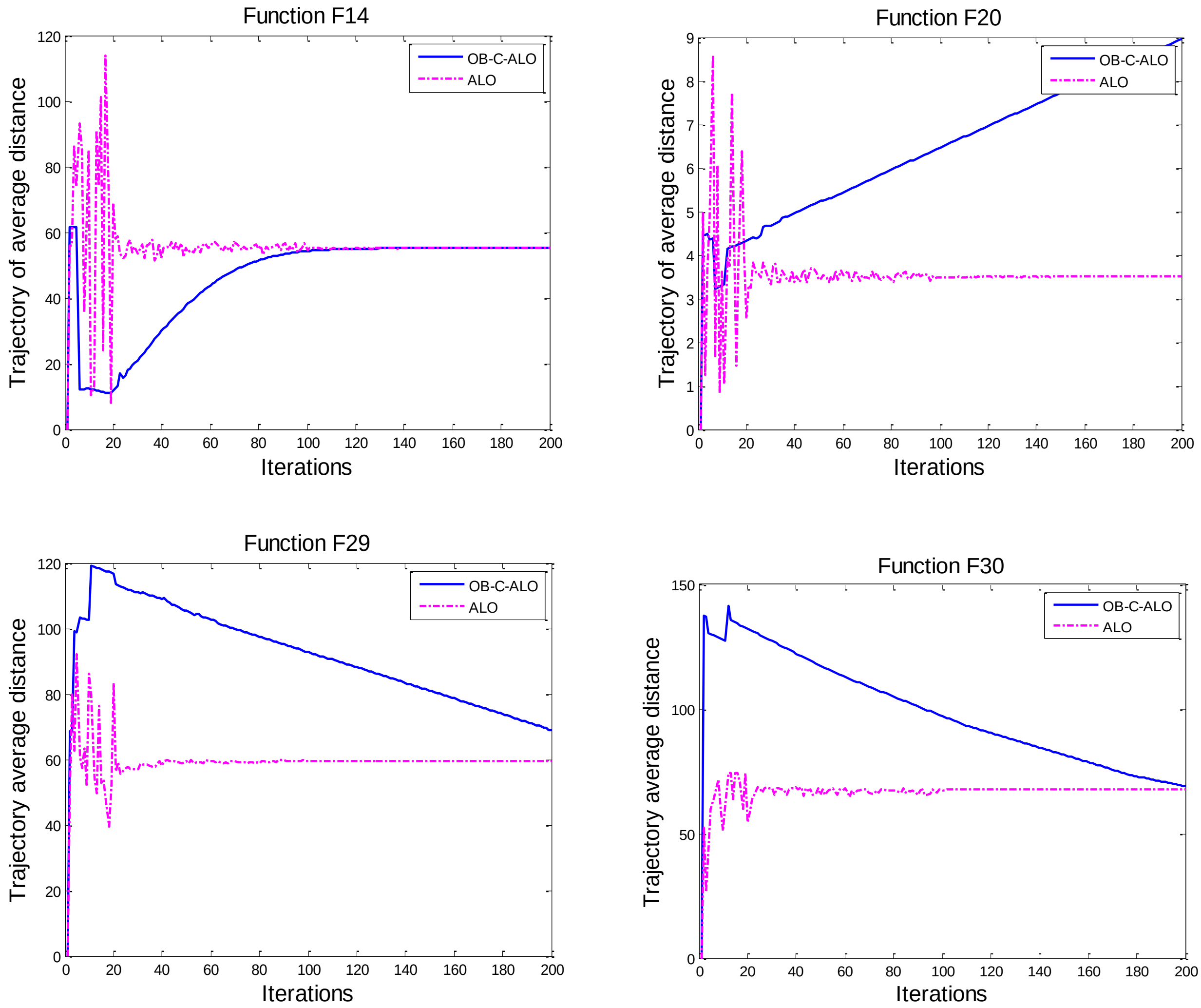


Figure 5.8: (Continued)

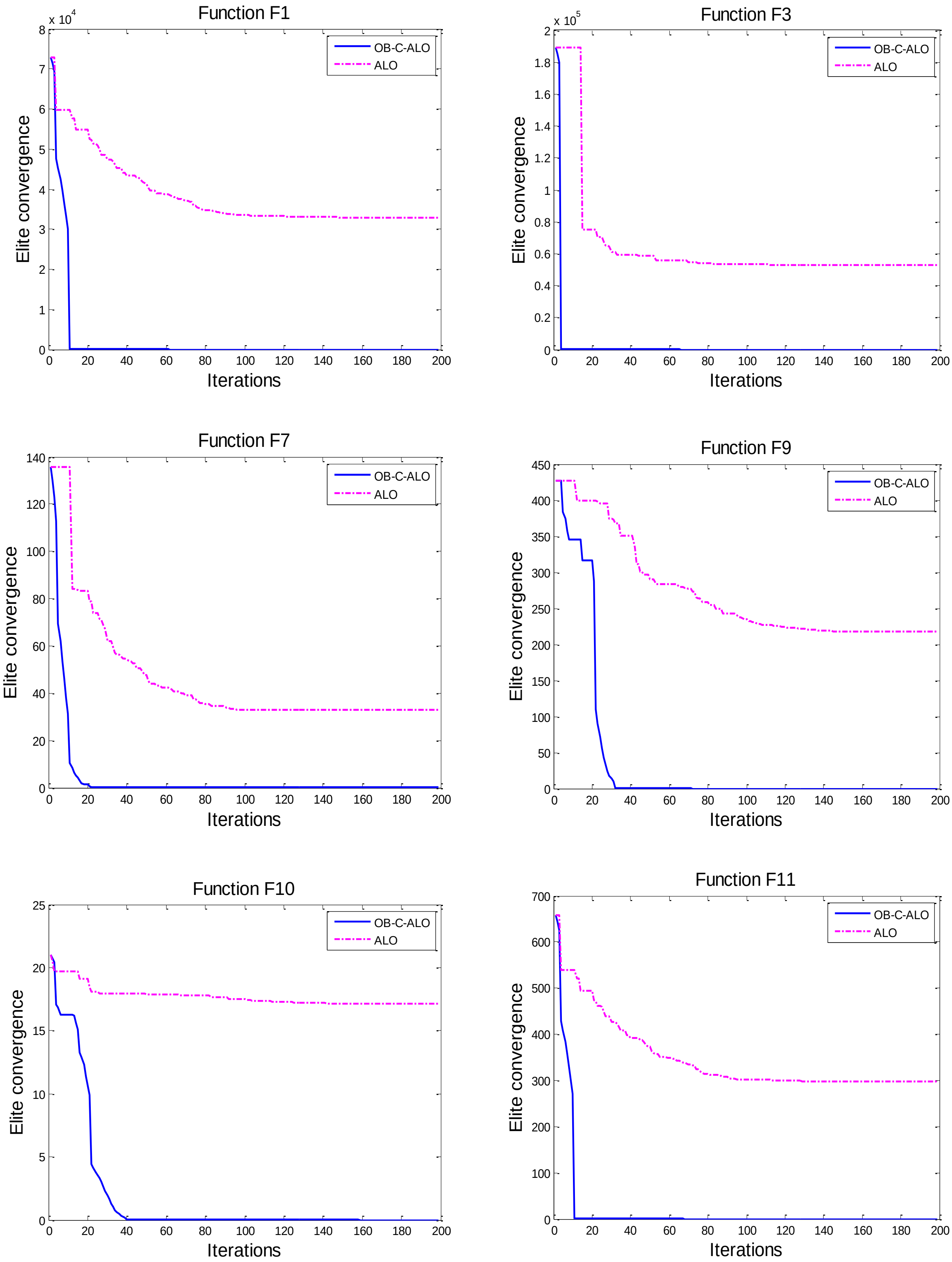


Figure 5.9: Elite convergence curves

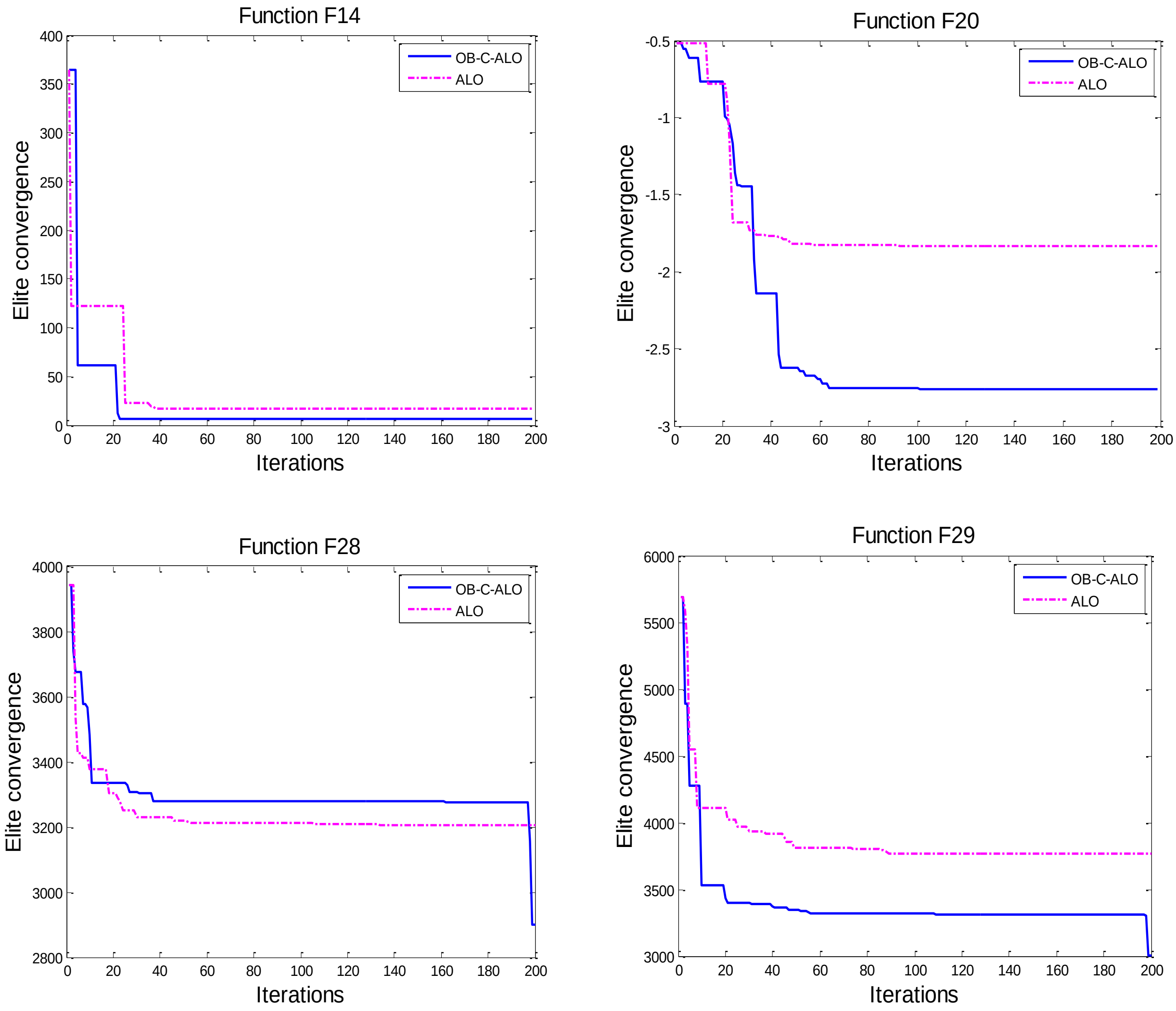


Figure 5.9: (Continued)

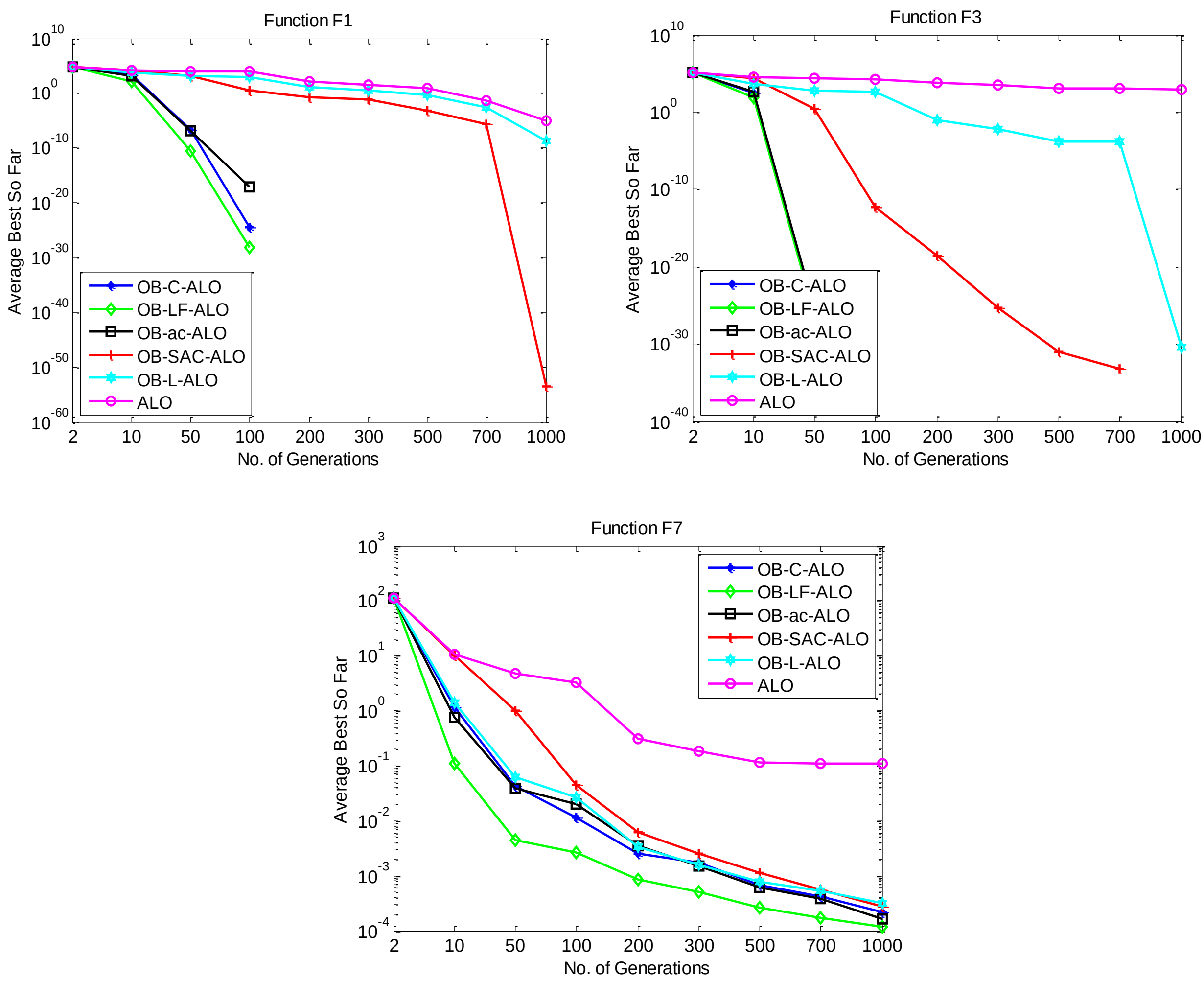


Figure 5.10: Convergence behaviour on three of the unimodal functions $F1, F3$ and $F7$

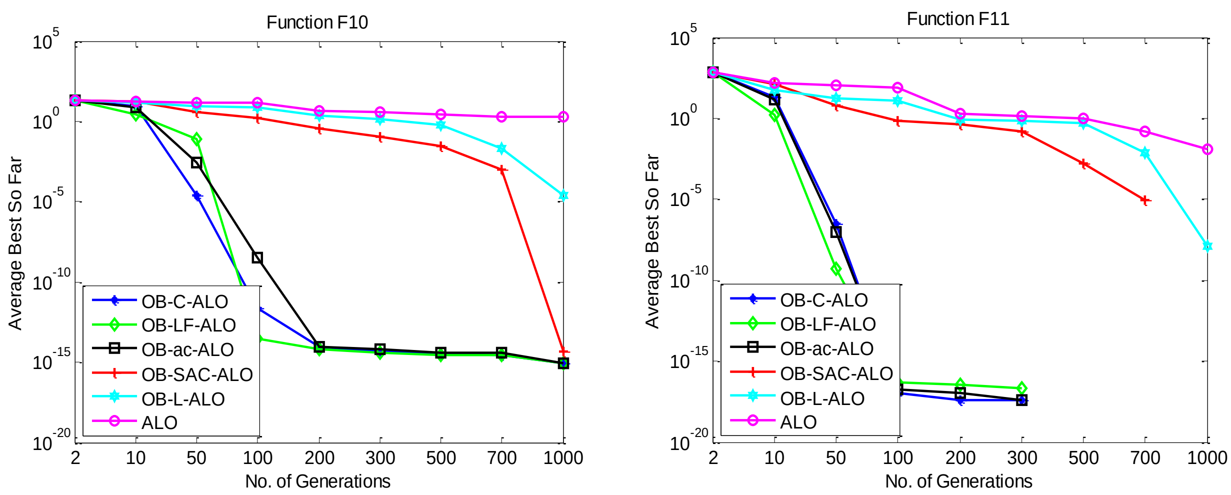


Figure 5.11 : Convergence behaviour on two of the multimodal functions $F10$ and $F11$

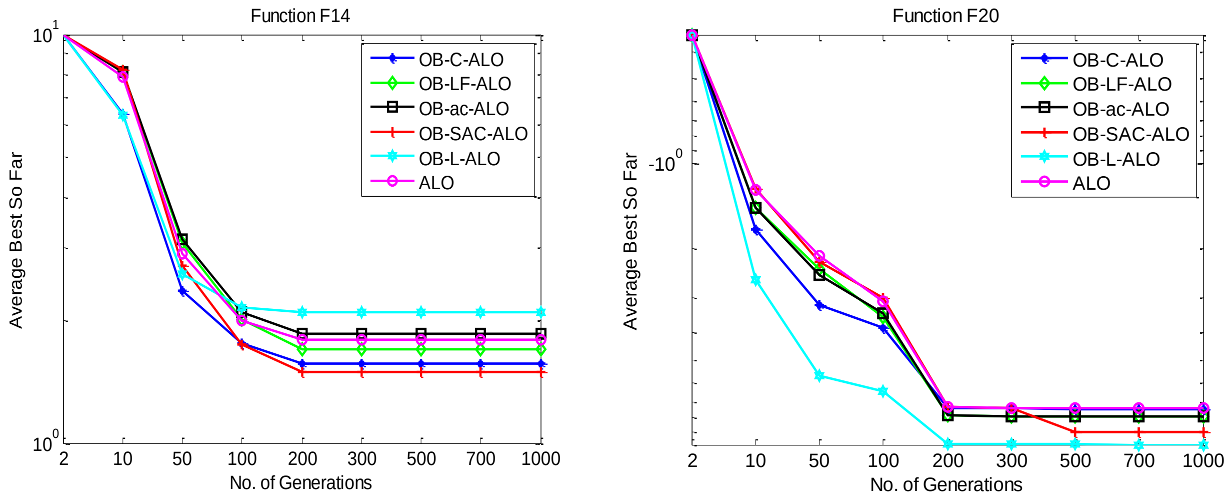


Figure 5.12: Convergence behaviour on two of the fixed dimensional multimodal functions $F14$ and $F20$

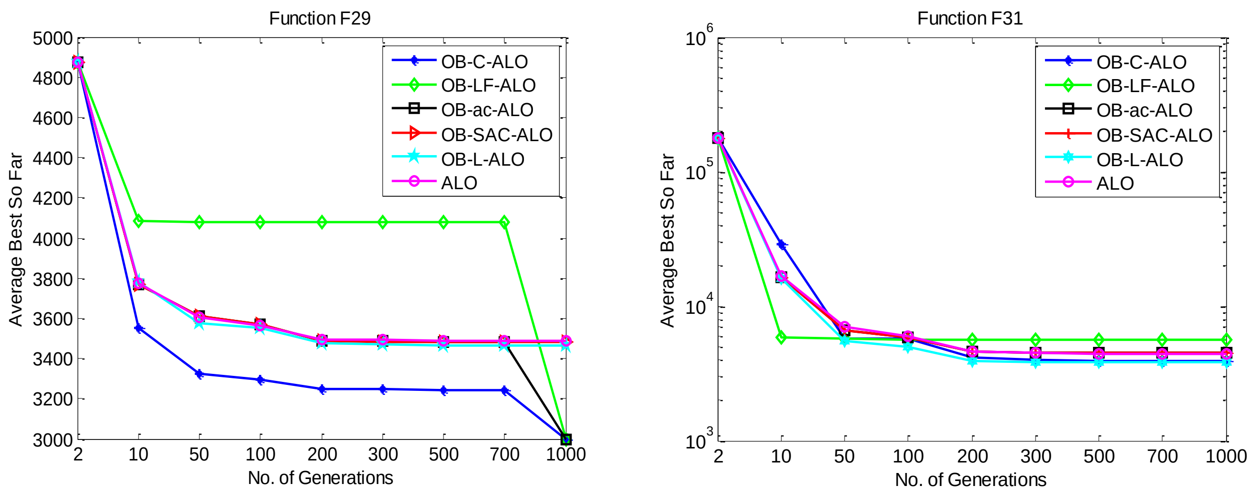


Figure 5.13: Convergence behaviour on two of the composition functions $F29$ and $F31$

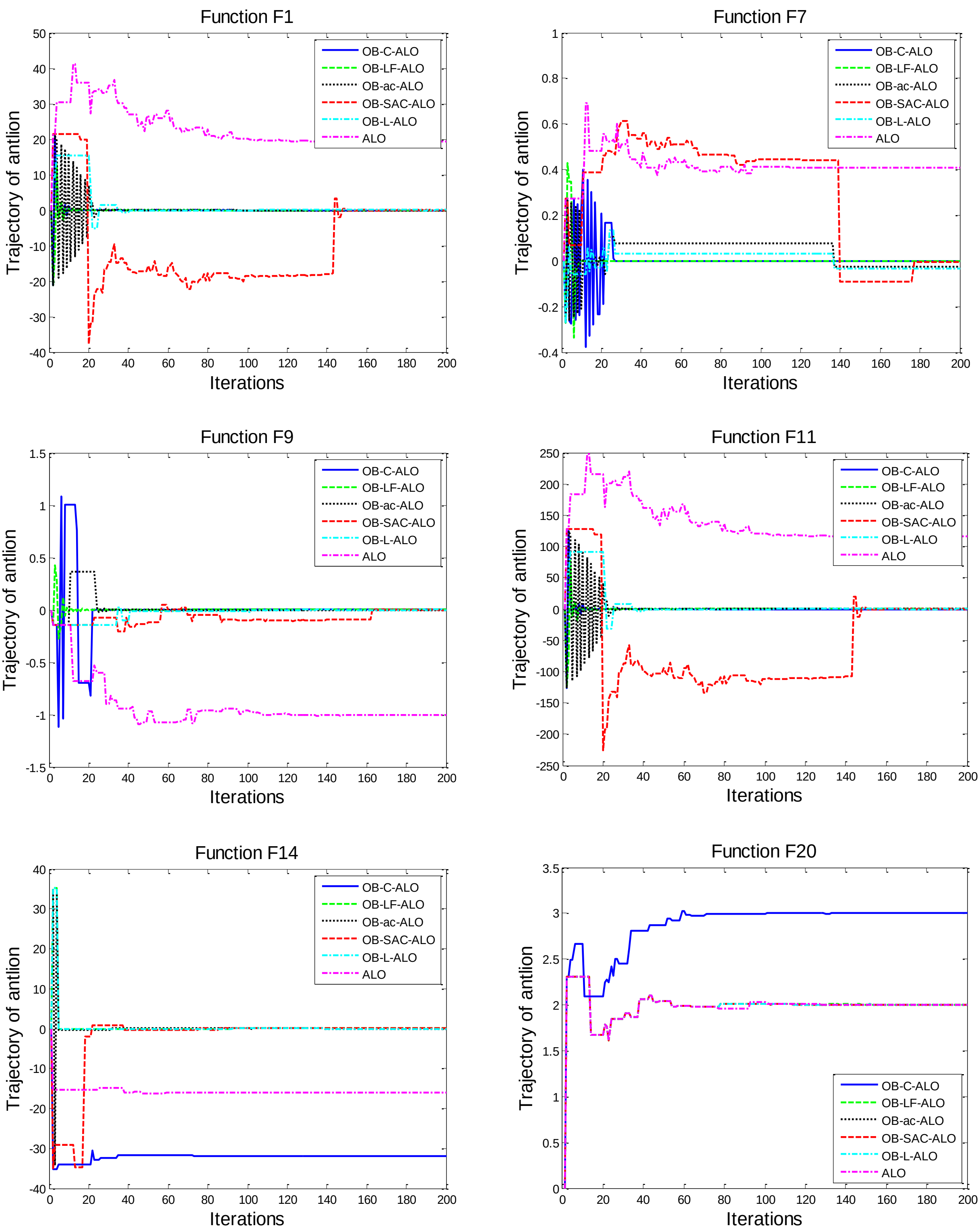
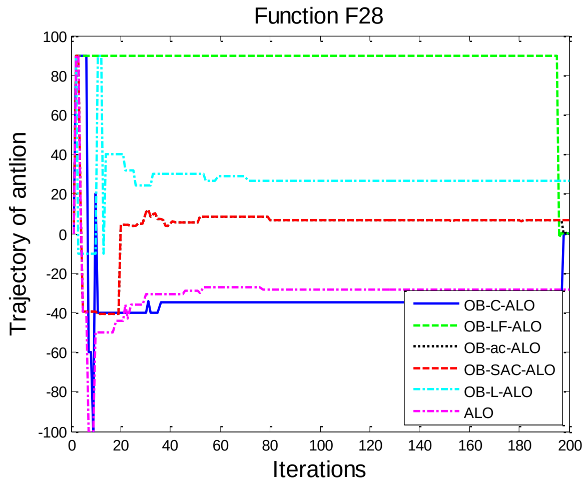


Figure 5.14: Trajectories of elite antlion



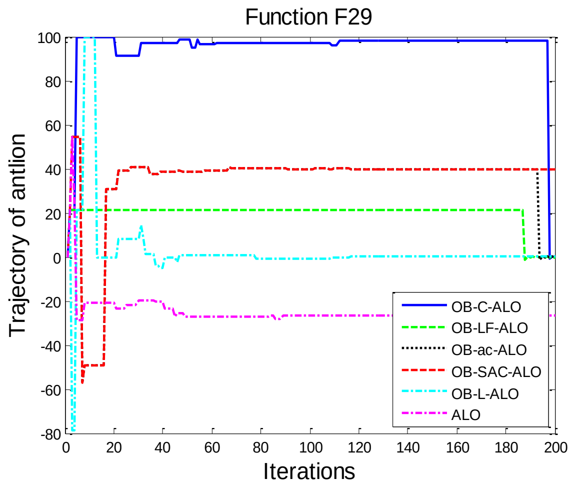


Figure 5.14: (Continued)

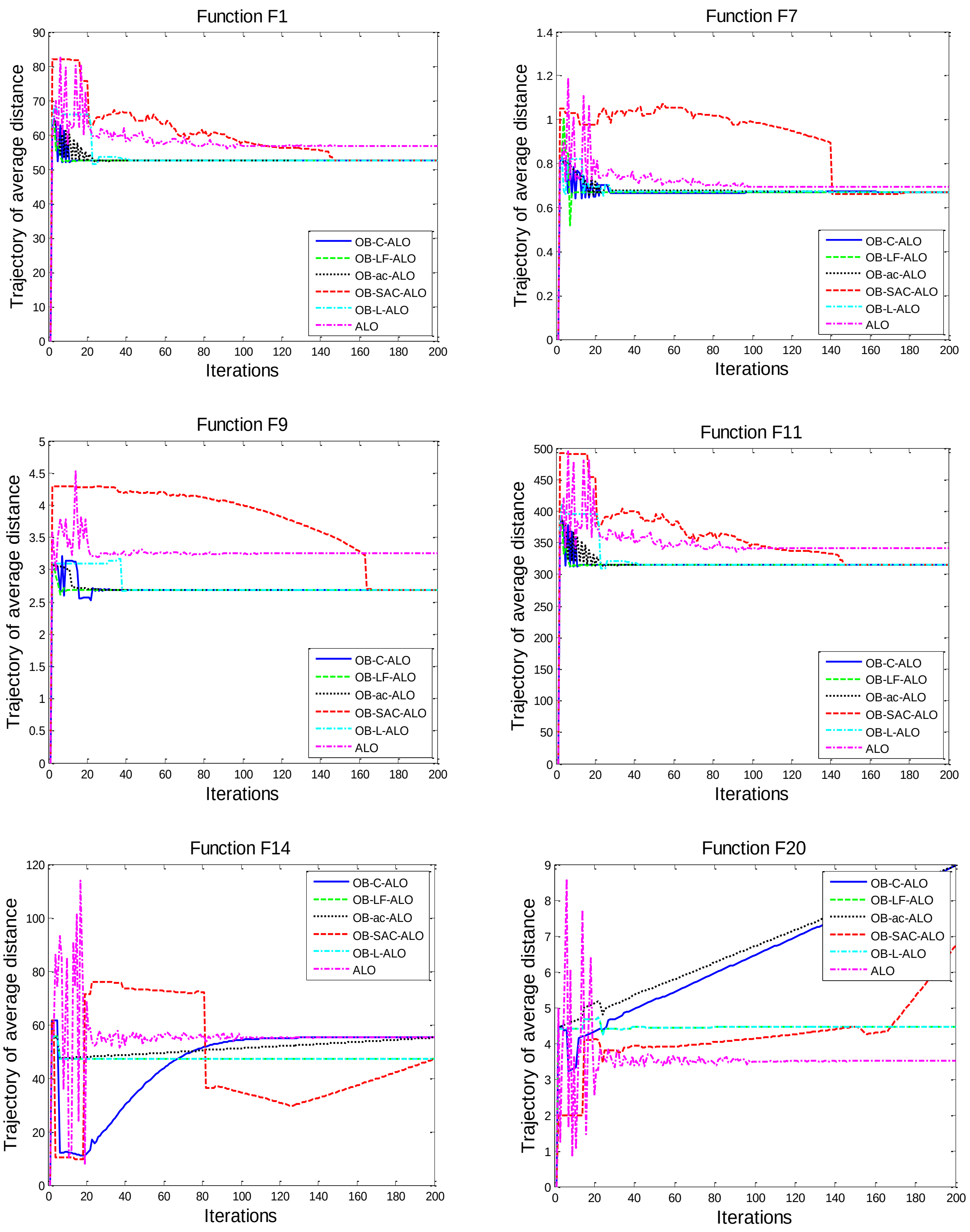
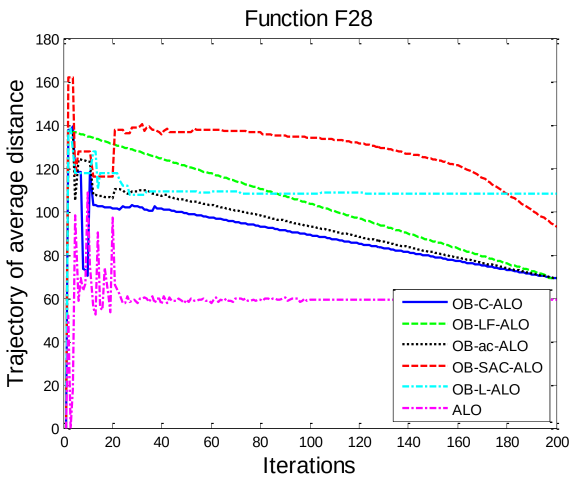


Figure 5.15: Trajectory analysis of average distance between search agents



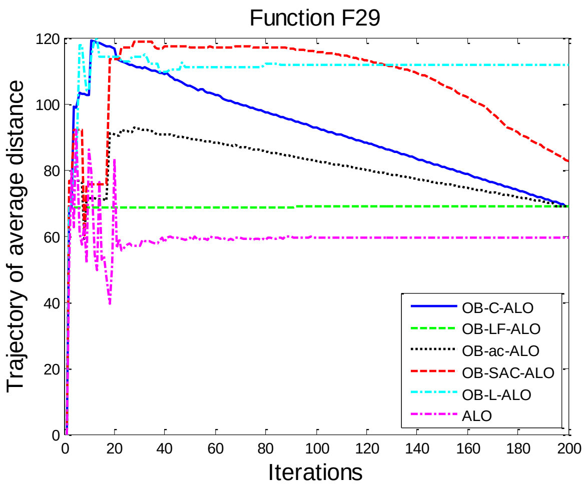


Figure 5.15: (Continued)

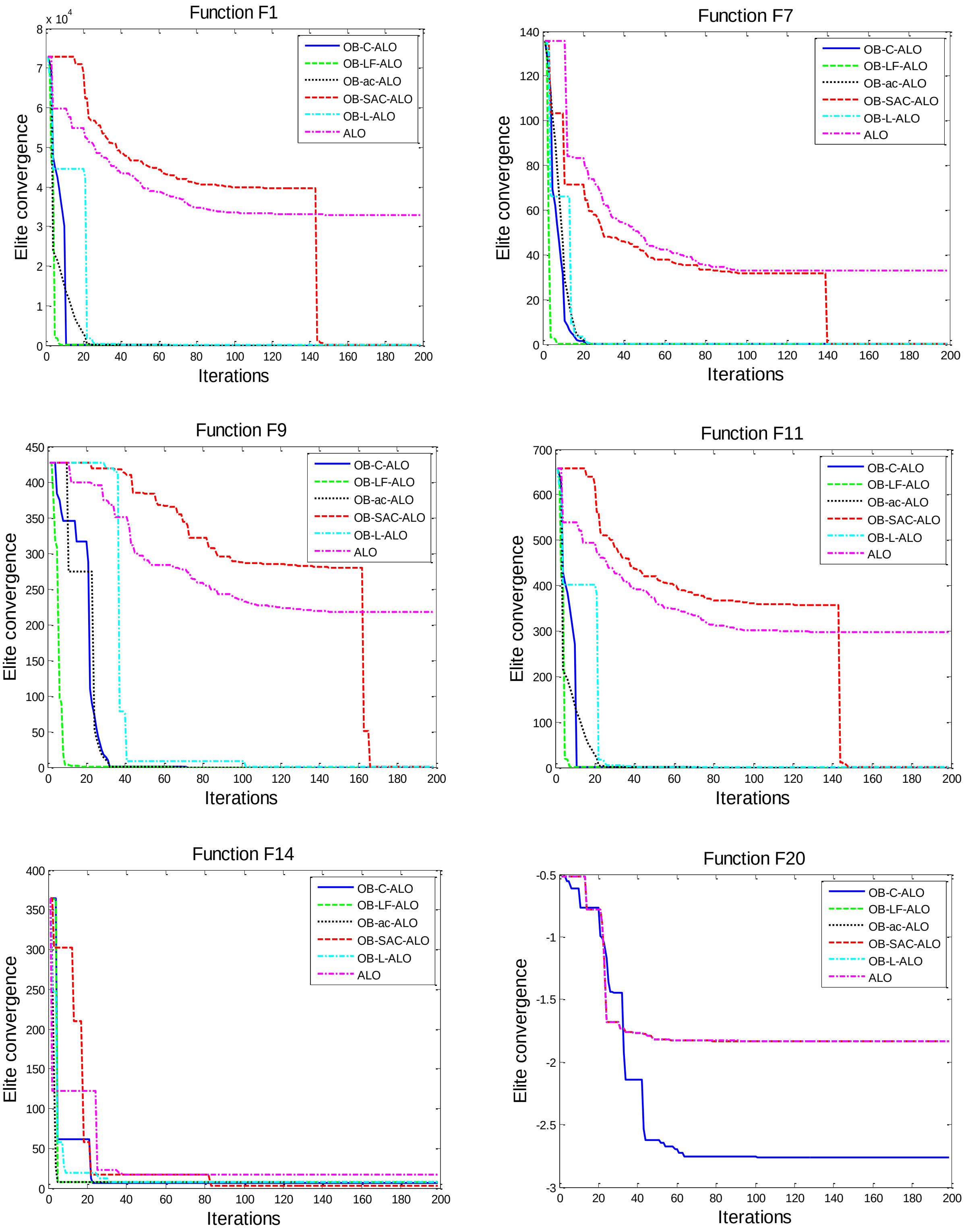


Figure 5.16: Elite convergence curves

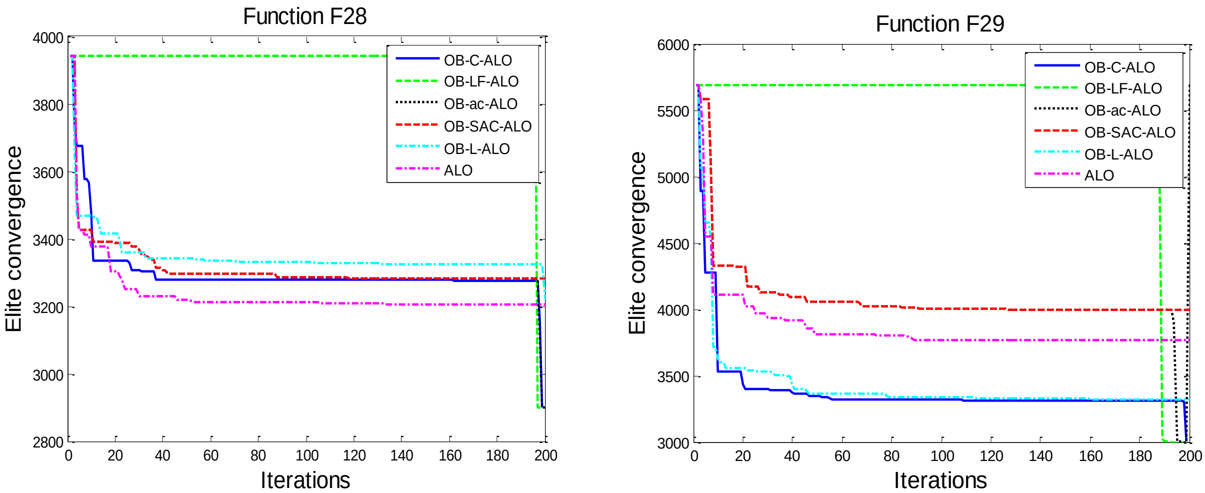


Figure 5.16: (Continued)

CHAPTER 6

Order Reduction of High Order Continuous Linear Time Invariant System using Antlion Optimizer and its Modified Variants

The Objective of this chapter is to investigate the capability of the Antlion optimizer and its proposed variants to solve a real life problem in the field of Electrical engineering. The problem is to reduce the order of complex continuous liner time invariant system into lower order keeping system as stable while it is in transient state**.**

6.1 Introduction

In the field of power system and control engineering, it is necessary to minimize the complexities of high order continuous linear time invariant systems. This process can be accomplished by applying order reduction techniques to obtain steady state while the system is in transient or dynamic condition. This problem can be seen as an optimization problem of minimizing the error between original high order and reduced order system. In this chapter, this problem is dealt using classical ALO and its proposed variants in this thesis to reduce the order of complex system. For this purpose, integral square error (ISE) is chosen to be minimized as an objective function to verify the performance. Three well known single input single output (SISO) problems of varying orders are considered in this chapter and the results are obtained using classical ALO and its variants which are further compared with the existing techniques of order reduction. The obtained results using proposed algorithms are found to be better as compared to other order reduction techniques available in literature.

The chapter is organized as follows: Section 6.2 describes the model order reduction, related literature and basic definitions. Section 6.3 formalizes the order reduction problem as an optimization problem including pseudo code and computational complexity. Section 6.4 illustrates the various SISO problems of varying order including obtained results discussion and analysis. Section 6.5 represents conclusive remarks.

Partial contents of this chapter has been published as:

- **Dinkar, S. K., & Deep, K. (2018). Accelerated Opposition-Based Antlion Optimizer with Application to Order Reduction of Linear Time-Invariant Systems. Arabian Journal for Science and Engineering, 44(3), 2213-2241. (SCIE, IF-1.092).**

6.2 Model Order Reduction Problem

Large systems are represented in the form of complex high order differential equations in the area of control system engineering. There are lot of difficulties while dealing and analyzing these systems [133]. This situation originates the spirit of approximating high order continuous linear time invariant complex system into corresponding low order systems [134]. In reality, reduction of order is very important as the attained system after transformation must be stable [135]. Davison [136] in 1966 proposed the very first technique of model order reduction and later updated by Chidambara[137-139] in 1967. Few techniques were proposed based on minimal realization [140-143]. Some other popular methods such as Aggregation method [144], Padé approximation [145], Routh approximation [146], Moment matching technique [147] and Routh stability technique [148] etc. which describe the effective employment of high order reduction in to efficient low order reduced system.

On the contrary, it is observed that sometimes the model may be restricted in non-minimal phase [149]. This inspired researcher and practitioners to use optimization techniques to remove this restriction. In reduction of high order system, the optimization techniques are utilized to minimize the integral square errors between the transient parts of high and low order reduced systems [150-151].

The increasing popularity and efficiency to solve highly complex real-life applications using nature inspired optimization algorithms and effective implementation to this problem is found to be a good practice [134]. Minimization of error is one method to be utilized while reducing higher order system into corresponding lower order system. Integral square error (ISE) is one of the measures to optimize during this process as objective function to optimization technique [152-153]. Though the other performance metrics such as integral absolute error(IAE),integral time squared error(ITSE) etc. are also available but in this chapter integral square error(ISE) is utilized due to its ability in diminishing huge error in stable and dynamic(transient) state.

With the above reasons, the proposed variants of antlion optimizer (ALO)are employed for reduction of high order complex system into low order system while using ISE as objective(fitness)function. The numerator and denominator of transfer function of reduced order system are determined using the same variant of ALO. Number of search agents are fixed as 30 with the termination criteria as maximum number of iteration which is chosen as 200. The lower and upper bound for each problem are chosen 0.005 and 4 respectively.

6.2.1 Basic Definitions

Model order reduction [134]

Model order reduction can be defined as follows:

Definition : Assume that $P(s): x \rightarrow y$ be transfer function depicting a system of high order n,then the order reduction method determines the numerator and denominator coefficient values of a reduced order transfer function $P(s): x \rightarrow \tilde{y}$ with order $\tilde{n}$ such that $\tilde{n} < n$ implying for the same input $x(t)$,the obtained output is $\tilde{y}(t) \approx y(t)$.

Single Input Single Output (SISO) System [134]

The transfer function form of nth original higher order continuous linear time invariant system can be depicted as:

$$P(s)=\frac{\sum_{z=0}^{m} a_z s^z}{\sum_{z=0}^{n} b_z s^z}=\frac{a_0+a_1 s+a_2 s^2+a_3 s^3+\cdots+a_m s^m}{b_0+b_1 s+b_2 s^2+b_3 s^3+\cdots+b_n s^n} \tag{6.1}$$

where $m \leq n$ and $z, m, n \in I$ and the system is depicted in eq. (6.1)

The system shown in eq. (6.1) is a “bounded input bounded output(BIBO) stable continuous” high order system of order n in actual form; a_z and b_z are represented as constant coefficient of variable s as numerator and denominator polynomial. Here, s is a complex variable. The proposed variants of antlions optimizer (ALO) estimate the actual nth order system $P(s)$ into the $\widetilde{P(s)}$ of reduced form of rth order in such a way that [134]

$$\widetilde{P(s)} = \frac{\sum_{z=0}^{m_r} \widetilde{a_z} s^z}{\sum_{z=0}^{n_r} \widetilde{b_z} s^z} \tag{6.2}$$

here $m_r \leq n_r$ and $z, m_r, n_r \in I$. $\widetilde{a_p}$ and $\widetilde{b_p}$ are coefficient of s in numerator and denominator respectively in reduced order system as shown in eq.(6.2).

6.3 Formulation of Model order reduction as an optimization problem

In this section, the MOR problem is modelled as optimization problem. Let $P(s)$ be the actual high order system to be reduced in to $\widetilde{P(s)}$ as low order system.

Formulation as optimization problem can be represented as:

$$min\ ISE=\int_0^{\infty}[e(t)-\widetilde{e(t)}]^2 dt \tag{6.3}$$

where ISE=Integral square error; $e(t)$ and $\widetilde{e(t)}$ are step responses of actual $P(s)$ and reduced $\widetilde{P(s)}$ LTI systems respectively.

After formulating the problem, the proposed techniques of ALO are applied to determine the coefficient $[a_0, \ldots, a_{r-1}]$ and $[b_0, \ldots, b_{r-1}]$ of numerator and denominator polynomial of reduced system $\widetilde{P(s)}$ while minimizing the ISE. Here r denotes the order of reduced system. The determined coefficients must be able to transform the high order system into low order system that the system is stable in transient as well as in steady state.

6.3.1 Computational Complexity

Computational complexity highly relies on size of population, number of iterations, number of function evaluation and internal loops within the steps to determine the output. As far as the complexity of proposed ALO algorithms is concerned, it comes out to be $O(it_{max} * D * N) * O(f(X))$ as described in computational complexity section of earlier chapters. Here, it_{max}, D and N represent maximum number of iterations, dimensions and size of population or candidate solutions. $f(X)$ depicts as objective function to determine the integral square error (ISE).

It is evident from computational steps that the instruction to determine ISE is executed for N times. Thus the computational complexity of objective function $f(X)$ is $O\ (ISE)$ where $ISE = \int_0^{\infty} [e(t) - \widetilde{e(t)}]^2 dt$.

6.4 Illustration of single input single output (SISO) problems

6.4.1 First Problem [135]

Let us consider a problem of fourth order in transfer function form as

$$P(s) = \frac{s^3+7s^2+24s+24}{s^4+10s^3+35s^2+50s+24} \tag{6.4}$$

Performance evaluation and discussion of results (Problem 1)

The comparison of obtained ISE value is performed with the original system, classical ALO and its proposed variants and other reduced systems available in literature as shown in Table 6.1. The best ISE value obtained out of 30 independent runs is taken. It is evident from the table that the obtained ISE using OB-SAC-ALO is better (minimum) than the other techniques. The coefficient of numerator and denominator polynomial determined using OB-SAC-ALO are $a_0 =$

2.4736, $a_1 = 1.0124$ and $b_0 = 2.4737$,$b_1 = 3.6908$,$b_2 = 1.3422$ respectively. The obtained reduced second order system is $\widetilde{P(s)}$=$\frac{1.0124s+2.4736}{1.3422s^2+3.6908s+2.4737}$ and the obtained integral square error (ISE) **is** $\mathbf{7.5788 \times 10^{-4}}$.

The obtained values of a_0 in numerator and b_0 in denominator guarantees the steady state after reducing higher order to low order. The comparison among unit step responses in time domain and frequency domain is depicted in Figure 6.1 and Figure 6.2 respectively. It is evident from the figures that the reduced order system obtained using proposed OB-SAC-ALO is better approximating the original higher order system as compared to other reduced models. The performance order of classical ALO and its proposed variants can be given as OB-SAC-ALO $>$ OB-ac-ALO $>$ OB-L-ALO $>$ ALO $>$ OB-C-ALO $>$ OB-LF-ALO.

The analysis of reduced system in transient state is also performed using step information as shown in Table 6.2 for problem 1. Three parameters namely Rise time(T_r),Settling time(T_s) and Peak overshoot(M_p) are chosen to show the transient response. The parameter T_r ensures whether the proposed reduced model takes similar time for response to rise. The parameters T_s and M_p also approximate the original system to ensure whether the proposed system conforms the steady state during transient response. It is visible from the Table 6.3 that the rise time of OB-SAC-ALO is better as compared to other techniques. Though the settling time is slightly higher but peak overshoot is better approximated by OB-SAC-ALO.

6.4.2 Second Problem[135]

Considering eighth order problem in terms of transfer function as

$$P(s) = \frac{18s^7+514s^6+598s^5+36380s^4+122664s^3+222088s^2+185760s+40320}{s^8+36s^7+546s^6+4536s^5+22449s^4+67284s^3+118124s^2+109584s+40320} \quad (6.5)$$

Performance evaluation and discussion of results(Problem 2)

The best solution attained out of 30 runs using classical ALO and its variants to evaluate their performance is exhibited in Table 6.3. The obtained coefficient of numerator and denominator using OB-SAC-ALO are shown as $a_0 = 0.5609$, $a_1 = 1.8172$ and $b_0 = 0.5608$,$b_1 = 0.7341$,$b_2 = 0.1079$ .The reduced second order system obtained is $\widetilde{P(s)}$=$\frac{1.8172\,s+0.5609}{0.1079\,s^2+0.7341s+0.5608}$ and the obtained integral square error (ISE) is $\mathbf{6.5414 \times 10^{-3}}$.

It is observed from the Table 6.3 that the OB-SAC-ALO determines the minimum ISE value as compared to other methods. It is also evident from the table that the numerator and denominator polynomial values are well capable of approximating the original higher order system and remains steady throughout the transient phase. The performance order is given as OB-SAC-ALO $>$ OB-ac-ALO $>$ ALO $>$ OB-L-ALO $>$ OB-C-ALO $>$ OB-LF-ALO.

The time response and frequency response diagrams in time and frequency domain are shown in Figure 6.3 and Figure 6.4 respectively. It is clearly observed from both the figures that the OB-SAC-ALO model is better representing the original system as compared to other existing systems.

The step information analysis of transient response analysis of problem 2 is depicted in Table 6.4. The chosen parameter for analysis Rise time (T_r) ensures that reduced model takes approximately similar time for response to rise. Though the Settling time (T_s) and Peak overshoot (M_p) of reduced model is slightly different with the original system yet the reduced model is capable of approximating the original system as evident from the unit step and frequency response diagrams.

6.4.3 **Third Problem** [156]

The transfer function form of problem 3 is shown in eq. (6.6)

$$P(s) = \frac{s+4}{s^4+19s^3+113s^2+245s+150} \tag{6.6}$$

Performance evaluation and discussion of results (Problem 3)

While comparing the obtained ISE using classical ALO and its proposed variants as shown in Table 6.5, it can be concluded that the ISE of second order reduced system for OB-C-ALO is better than the ISE of original system as determined using various variants of ALO and other methods. The numerator coefficients using OB-C-ALO are $a_0 = 0.0050$, $a_1 = 0.0050$ and denominator coefficients are $b_0 = 0.1877$, $b_1 = 0.4268$ and $b_2 = 0.3442$. The second order reduced system is $\widetilde{P(s)}$=$\frac{0.0050\ s+0.0050}{0.3442\ s^2+0.4268s+0.1877}$ and the integral square error (ISE) value is $\mathbf{4.8687 \times 10^{-5}}$.

It is evident from the obtained results that the reduced system using proposed OB-C-ALO is better approximating the high order system as compared to other systems. The values of numerator and denominator also reveal that the reduced system is steady as well during the transient phase.

The step response comparison and frequency response comparison are shown in Figure 6.5 and Figure 6.6 respectively.

To analyze the transient response, step information comparison for problem 3 is performed and depicted in Table 6.6. The obtained values of the Rise time (T_r) and Settling time (T_s) are slightly different from the original system but Peak overshoot(M_p) of reduced model is exactly the same as of original system. It shows the steady state of the reduced model which is also evident from the unit step response diagram.

6.5 Conclusion

In this chapter, the performance of proposed variants of classical ALO described in previous chapters is investigated to solve a real world complex application of model order reduction of linear time invariant system in the field of control system. The performance of these algorithms are verified on three single input single output (SISO) systems of different orders. The performance metric is chosen so as to minimize the objective function in terms of integral square error (ISE) while reducing the order of input system. The analysis of step information is performed using Rise time (T_r), Settling time (T_s) and Peak overshoot (M_p) for investigating the performance of proposed variants over model order reduction problem. The reduced models using all ALO algorithms and other techniques available in literature are compared to devise the performance order. The obtained results are analyzed using step response in terms of time and frequency diagrams.

Table 6.1: Comparison of order reduction Techniques with respect to ISE (Problem 1)

Model order reduction techniques	Reduced Model	ISE
Classical ALO	$\frac{1.2253s+2.2541}{1.5689s^2+3.6569s+2.2543}$	8.0415×10^{-4}
OB-L-ALO	$\frac{1.1806s+2.3043}{1.2104s^2+3.4623s+2.3048}$	7.8362×10^{-4}
OB-SAC-ALO	$\mathbf{\frac{1.0124s+2.4736}{1.3422s^2+3.6908s+2.4737}}$	$\mathbf{7.5788 \times 10^{-4}}$
OB-ac-ALO	$\frac{1.2806s+2.5048}{1.6504s^2+3.6848s+2.5050}$	7.6574×10^{-4}
OB-C-ALO	$\frac{1.8793s+1.7185}{2.2557s^2+3.7137s+1.7189}$	2.2247×10^{-3}
OB-LF-ALO	$\frac{1.7316s+1.7559}{0.7997s^2+1.9986s+1.7678}$	297.95×10^{-2}
Birader and Saxena[135]	$\frac{0.2838s+1.00043}{0.3986s^2+1.3744s+1}$	1.1478×10^{-3}
Sikander & Prasad[134]	$\frac{0.6997s+0.6997}{s^2+1.45771s+0.6997}$	27.7989×10^{-3}
Desai & Prasad[154]	$\frac{0.8058s+0.7944}{s^2+1.65s+0.7944}$	2.8358×10^{-3}
Truncation Method[155]	$\frac{7s^2+24s+24}{35s^2+50s+24}$	70.138×10^{-3}
Routh Hurwitz[148]	$\frac{0.2057s+24}{30s^2+42s+24}$	97.41283×10^{-3}

Table 6.2: Step information comparison in terms of transient response (Problem1)

Model order reduction techniques	Rise time T_r(s)	Settling time T_s(s)	Peak overshoot M_p(%)
Original system	2.260	3.931	0
Classical ALO	2.265	3.803	0
OB-L-ALO	2.264	3.8172	0
OB-SAC-ALO	2.259	3.8658	0
OB-ac-ALO	2.264	3.817	0
OB-C-ALO	2.3035	3.7403	0.1640
OB-LF-ALO	0.6446	3.3878	11.2556
Birader and Saxena[135]	2.268	3.958	0
Sikander and Prasad[134]	2.301	3.410	1.0722
Desai and Prasad[154]	2.278	3.619	0.274
Truncation method[155]	2.737	4.080	0.564
Routh Hurwitz[148]	1.926	5.587	3.595

Table 6.3: Comparison of order reduction Techniques with respect to ISE (Problem 2)

Model order reduction techniques	Reduced Model	ISE
Classical ALO	$\frac{1.4764\ s + 0.4585}{0.0873\ s^2 + 0.5996s + 0.4587}$	6.6766×10^{-3}
OB-L-ALO	$\frac{1.8071\ s + 0.5609}{0.1069\ s^2 + 0.7342s + 0.5611}$	6.6814×10^{-3}
OB-SAC-ALO	$\mathbf{\frac{1.8172\ s + 0.5609}{0.1079\ s^2 + 0.7341s + 0.5608}}$	$\mathbf{6.5414 \times 10^{-3}}$
OB-ac-ALO	$\frac{1.8071\ s + 0.5609}{0.1069\ s^2 + 0.7342s + 0.5611}$	6.5814×10^{-3}
OB-C-ALO	$\frac{3.6545\ s + 1.1904}{0.2206\ s^2 + 1.4623s + 1.1895}$	2.0723×10^{-2}
OB-LF-ALO	$\frac{2.0955\ s + 2.2615}{0.1996\ s^2 + 1.2777s + 2.2435}$	734.05×10^{-2}
Birader and Saxena[135]	$\frac{3.1084s + 1.0005}{0.2075s^2 + 1.2434s + 1}$	38.1244×10^{-3}
Sikander & Prasad[134]	$\frac{16.92s + 5.263}{s^2 + 6.893s + 5.263}$	19.2386
Desai & Prasad[154]	$\frac{2.06774s + 0.43184}{s^2 + 1.17368s + 0.43184}$	69.569×10^{-3}

Table 6.4: Step information comparison in terms of transient response (Problem 2)

Model order reduction techniques	Rise time T_r(s)	Settling time T_s(s)	Peak overshoot M_p(%)
Original System	0.0569	4.8201	120.3504
Classical ALO	0.0596	5.0921	122.8296
OB-L-ALO	0.0596	5.0975	122.7421
OB-SAC-ALO	0.0597	5.0972	123.6659
OB-ac-ALO	0.0596	5.0975	122.7421
OB-C-ALO	0.0605	4.7676	124.3630
OB-LF-ALO	0.1078	1.8407	53.0155
Birader and Saxena[135]	0.0669	4.7990	123.3062
Sikander & Prasad[134]	0.0596	5.1033	122.1894
Desai & Prasad[154]	0.5244	8.7786	61.5459

Table 6.5: Comparison of order reduction Techniques with respect to ISE (Problem 3)

Model order reduction techniques	Reduced Model	ISE
Classical ALO	$\frac{0.1851\,s + 0.0051}{0.0135\,s^2 + 1.1377s + 0.1864}$	2.5649×10^{-3}
OB-L-ALO	$\frac{0.0127\,s + 0.0317}{1.4904\,s^2 + 1.8999s + 1.1927}$	8.9×10^{-4}
OB-SAC-ALO	$\frac{0.0050\,s + 0.0050}{0.1866\,s^2 + 0.4519s + 0.1871}$	2.6095×10^{-4}
OB-ac-ALO	$\frac{0.0050\,s + 0.0050}{0.1866\,s^2 + 0.4519s + 0.1871}$	2.6095×10^{-4}
OB-C-ALO	$\boldsymbol{\frac{0.0050\,s + 0.0050}{0.3442\,s^2 + 0.4268s + 0.1877}}$	$\boldsymbol{4.8687 \times 10^{-5}}$
OB-LF-ALO	$\frac{0.0050\,s + 0.0059}{0.0050\,s^2 + 0.0053s + 0.2332}$	11.705×10^{-2}
Singh[157]	$\frac{-494.596s + 405.48}{150s^2 + 2487s + 15205.5}$	2.856×10^{-3}

Table 6.6: Step information comparison in terms of transient response (Problem 3)

Model order reduction techniques	Rise time T_r(s)	Settling time T_s(s)	Peak overshoot M_p(%)
Original system	2.3691	4.3583	0
Classical ALO	0.0018	23.9191	489.6116
OB-L-ALO	2.2418	6.2403	4.5040
OB-SAC-ALO	3.4402	6.5765	0
OB-ac-ALO	3.4402	6.5765	0
OB-C-ALO	2.6605	3.8407	1.5105
OB-LF-ALO	0.0205	7.6379	506.5704
Singh[157]	2.301	3.410	1.0722

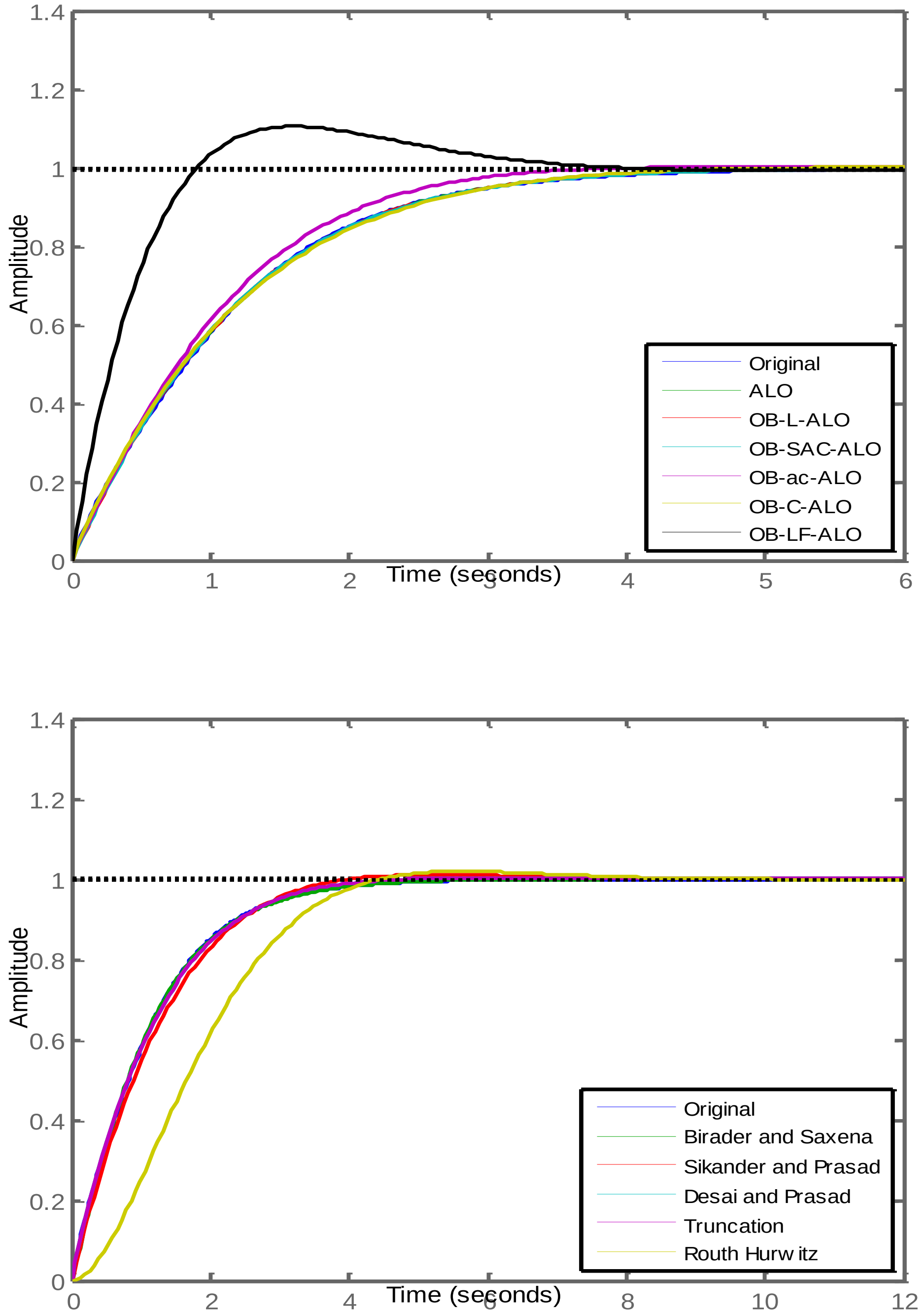


Figure 6.1: Comparison of step response for problem 1

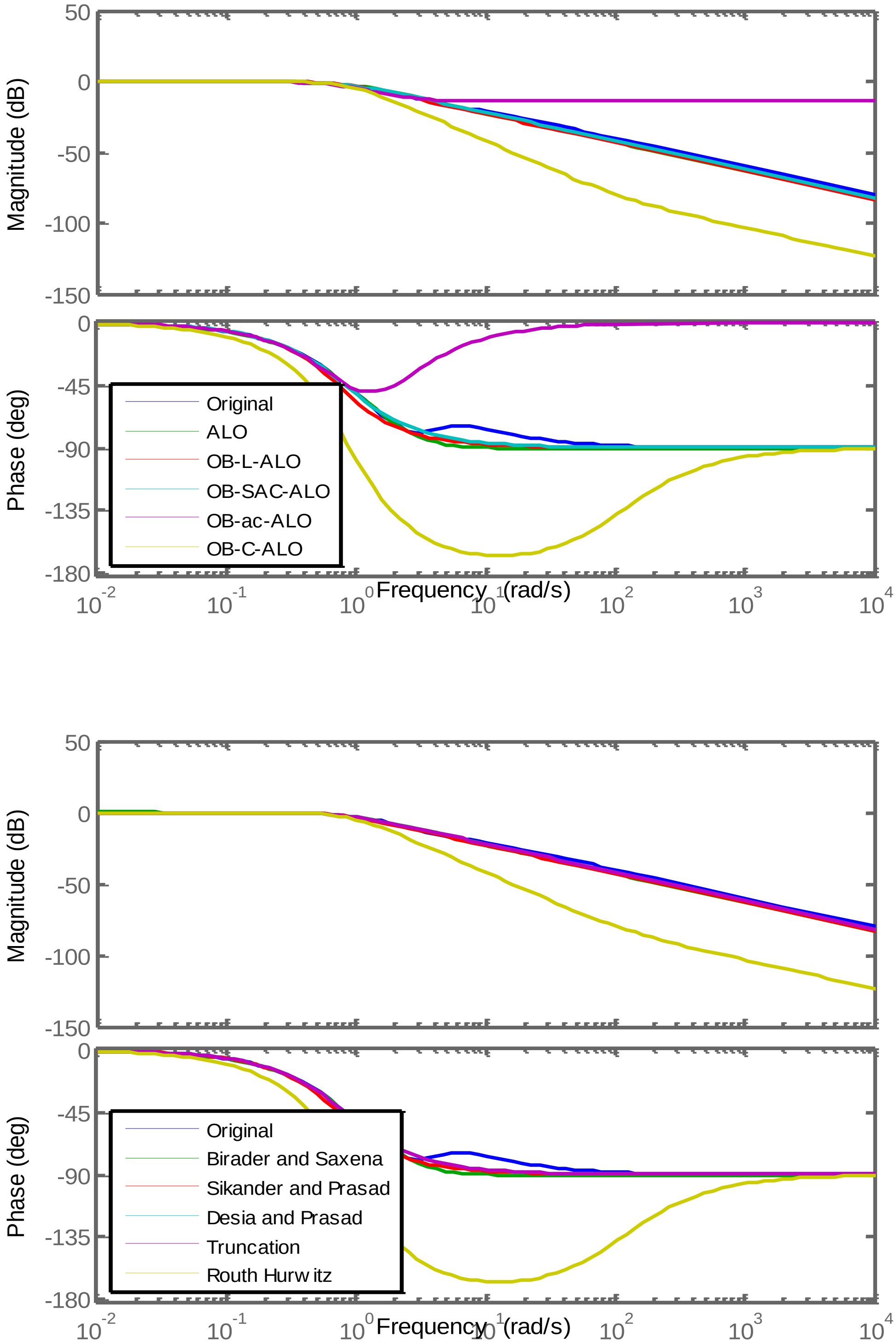


Figure 6.2: Comparison of frequency response for problem 1

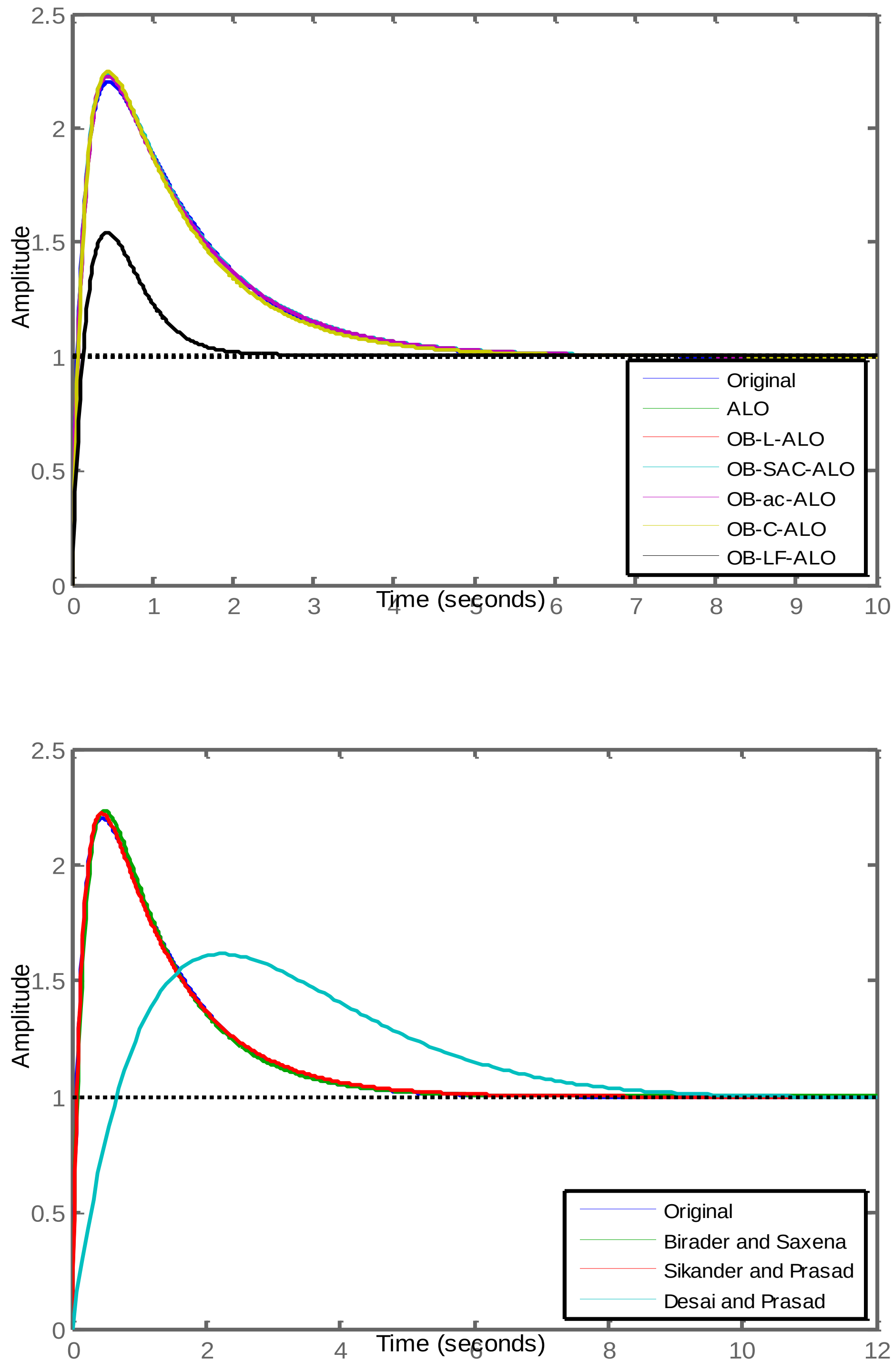


Figure 6.3: Comparison of step response for problem 2

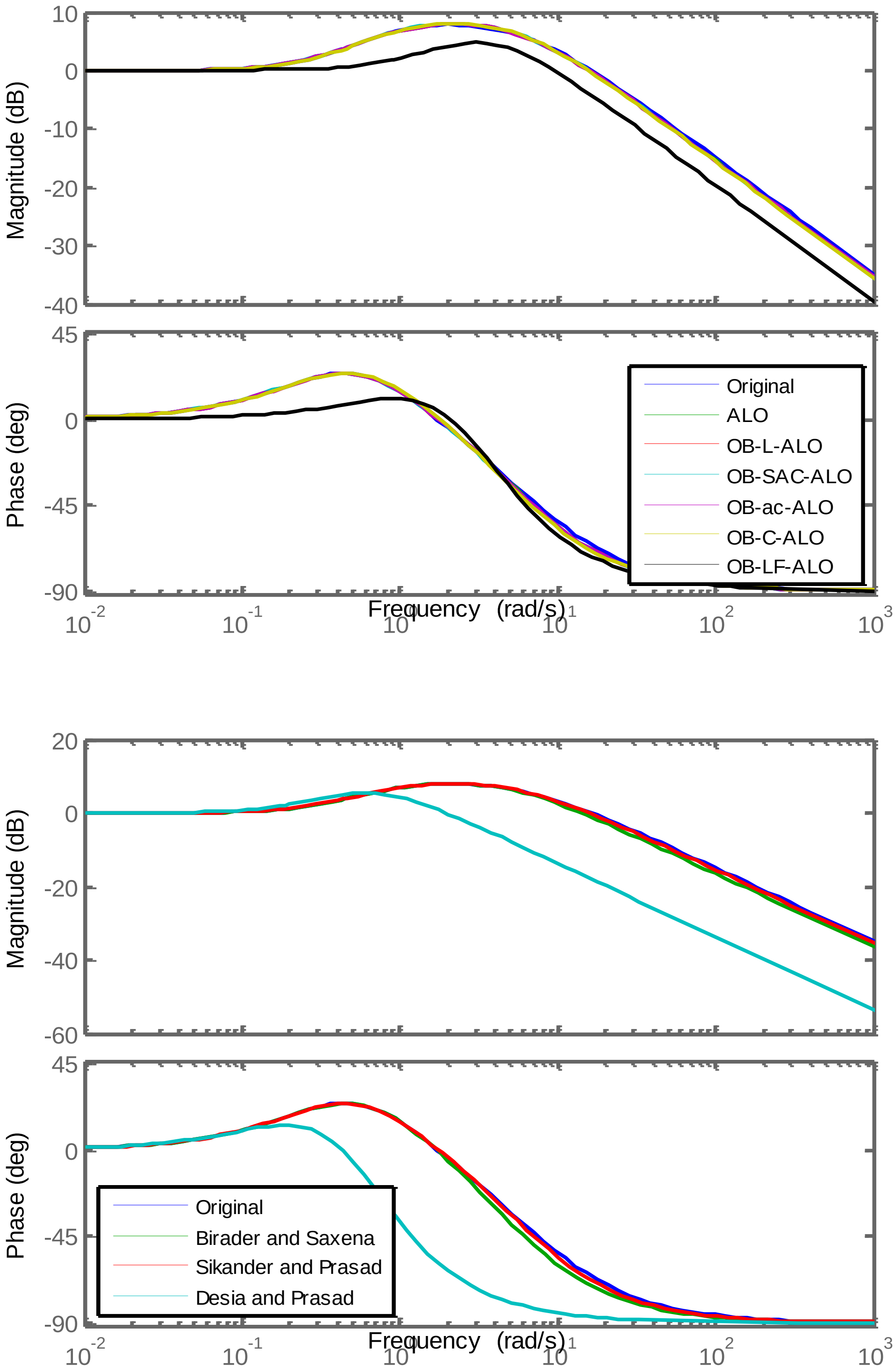


Figure 6.4: Comparison of frequency response for problem 2

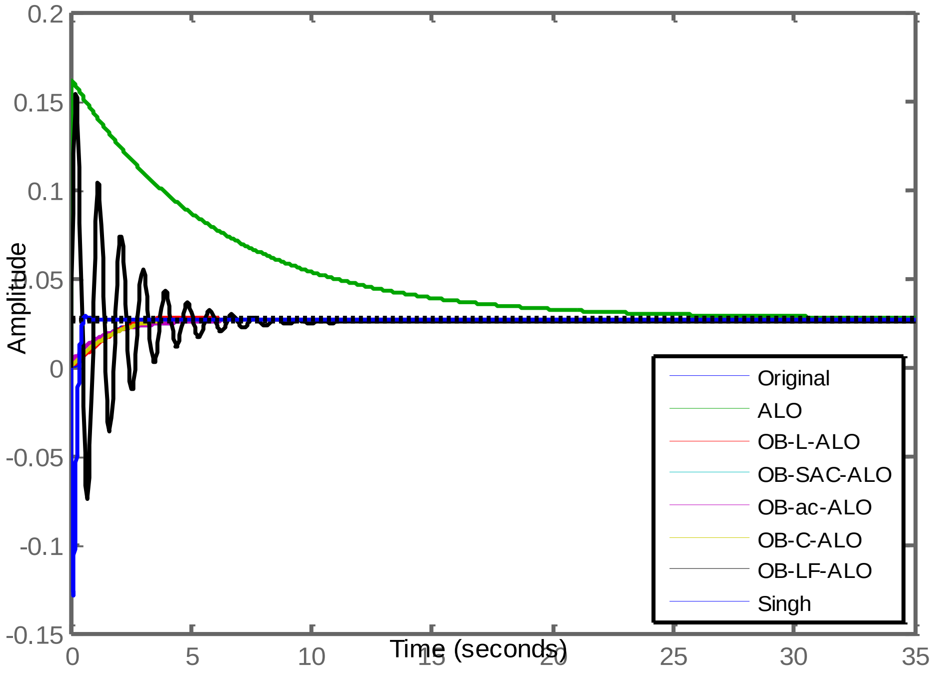


Figure 6.5: Comparison of step response for problem 3

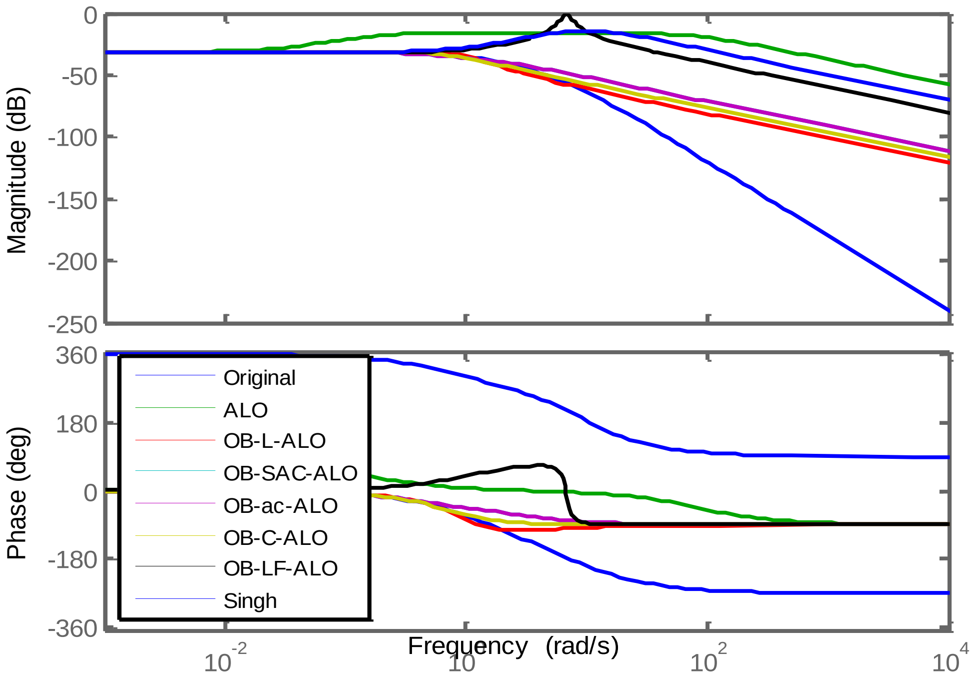


Figure 6.6: Comparison of frequency response for problem 3

CHAPTER 7

Single and Multi-objective Optimization of Nanofluid flow in flat tube to enhance Heat Transfer using Antlion Optimizer and its Variants

The objective of this chapter is to investigate the performance of classical Antlion optimizer and its variants to solve a real world problem from the field of Computational Fluid dynamics. This problems is to optimize two conflicting objective functions: maximizing heat transfer coefficient and minimizing pressure drop value using Nanofluid flow in flat tube.

7.1 Introduction

This optimization problem focuses to determine the five independent design variables to find out the optimal values of heat transfer coefficient ($\widetilde{H}$) and pressure drop(ΔP) parameters. This problem consist of two conflicting objective functions: first to maximize heat transfer coefficient and second to minimize pressure drop value. In this chapter, the problem is optimized using two approaches: First, single objective approach for both the objective functions separately to determine the optimal values of design variables using classical ALO and its proposed modified variants namely OB-C-ALO,OB-LF-ALO, OB-ac-ALO,OB-SAC-ALO and OB-L-ALO in this thesis. Secondly, multi-objective approach in which both the objective functions are optimized simultaneously to determine objective function values of heat transfer coefficient and pressure drop while optimizing design variables. This purpose is achieved using two different methods of multi-objective optimization: (i) Utilizing weighted sum approach of multi-objective optimization using classical ALO and its proposed variants (ii) Pareto based multi-objective optimization using multi-objective antlion optimizer (MOALO). The model used in this work is developed using Al_2O_3 –water nanofluid using horizontal flat tube with the help of computational fluid dynamics (CFD) and response surface methodology (RSM). The obtained results show superiority of ALO and its modified variants approach and also compared with RSM.

The content of this chapter is communicated as:

Dinkar, S. K., & Deep, K. (2018). Single and Multi-objective optimization of Nanofluid flow in flat tube to enhance Heat Transfer using Antlion Optimizer Algorithms. Arabian Journal for Science and Engineering, Springer.

7.2 Motivation and Literature review

The development of various engineering equipment require adequate heat transfer from one component to other [158]. There are three modes of heat transfer extensively used: convection, conduction and radiation. Out of these modes, convection is most popular and easy to use among all. The functioning of convection process relies on the properties of fluid including surface area and thermal properties. Now a days, nanofluid particles are efficiently used in the mechanism of increasing heat transfer in tube [159]. The cross section of these tubes may be of various shapes such as circular, rectangular, flat etc. Nonofluid can be termed as a mixture prepared with mixing nanoparticles such as Al_2O_3 or CuO with basic fluid which is responsible in increasing the thermal conductivity and heat transfer of the surface tube. In [160], the behaviour of thermal conductivity of nanoparticles is studied by analyzing various parameters such as size of nanoparticles, Brownian motion, nanolayer and temperature.

There are various mixture methods to accomplish heat transfer through various surfaces: single phase, two phase and Eulerian-Eulerian method. These methods are investigated using flow of nanofluids particles in circular tubes [161]. In [162], Al_2O_3 –water nanofluid flow is simulated using two phase mixture method in straight elliptic tubes. In similar way, flat tubes are used to increase the heat transfer instead of circular tubes due to the higher surface area. This is helpful to enhance the compactness and consequently able to increase the heat transfer of heat exchangers [163]. Using nanofluids in flat tubes, very few studies have been noticed till now [163-164].

The use of nanofluids within flat tubes result in enhancing heat transfer and minimizing pressure drop value. These two coefficient are different and conflicting in nature and can be solved as multi-objective problem as well as single objective problem. This optimization using nanofluid with flat tubes is investigated with modelling and optimization process as a combination of computational fluid dynamics (CFD) and response surface methodology [159]. Safikhani et al. [165] also implemented multi-objective approach to optimize Al_2O_3-water nanofluid to determine heat transfer coefficient and pressure drop using CFD with artificial neural network(ANN) and non-dominated sorted genetic algorithm(NSGA II). The responses received from various design variables are collected, statistical analyzed and mathematical modelled using RSM. Thus RSM is used to model these problem having various design variable and the responses are then optimized. In this chapter, the model which is used is developed by utilizing the CFD data and then using RSM technique for optimization of Al_2O_3-water nanofluid in horizontal flat tube [159].

Model thus obtained using RSM is optimized using classical ALO and its proposed variants in this thesis for single objective optimization. Since there are two conflicting objectives to be optimized hence the problem is handled using two different methods: (1) Firstly, by determining the optimal design variables for optimizing two different single objective functions separately: first one is to maximize the heat transfer coefficient and second problem is to minimize the pressure drop. (2) Secondly, by applying multi-objective optimization approach in which both the objective functions are utilized at the same time to determine optimal values of design variables. The metaheuristic techniques are very popular as there is no need of priori knowledge about search domain or any other information about the objective function to be utilized. The determination of heat transfer coefficient is quite popular using the evolutionary algorithm and it has been determined using GA [166].

The multi-objective approach is applied to optimize such problems due to the presence of more than one objective functions. These functions are solved by means of evolutionary algorithms as these algorithms are population based algorithm and able to provide parallel solution [167]. Pareto optimal solutions can also be determined in multi objective environment [168]. The set of these solutions represents non-dominated solutions where no solution is best but superior to the rest of the solutions [169]. Another popular approach is to scalarlize the objectives into single objective by using some weighting factor [170]. The use of GA to solve multi-objective problems paid much attention than other techniques [171]. Non-dominated sorted GA(NSGAII) has big impact on solving multi-objective problems since last decades[172-173]. In this work, both the methods including weighted sum approach of scalerizing the objectives into single objective using some weighting factors and pareto front based multi-objective approach are utilized to determine the optimum values of design variables and to obtain the heat transfer coefficient and pressure drop values.

7.3 Problem Formulation

In this work, there are five independent design variables : tube flattening variable (x_1), wall heat flux(x_2),fraction of nanoparticle volume(x_3),nanoparticle diameter(x_4) and inlet volumetric flow rate(x_5) which are used to estimate the optimum values of heat transfer and pressure drop. The boundary ranges of these design variables are shown in Table 7.1. The values of these design variables are utilized for optimization of Al_2O_3-water nanofluid flow using ALO and its proposed variants.

This problem is formulated as nonlinear optimization problem. In [159], RSM is used to optimize the five independent variables. The independent variables and data pertaining to these variables to produce a response variable can be given in the form of regression equation as an output using RSM. Two nonlinear polynomial equations are produced as a result: one defining heat transfer coefficient and second defining pressure drop as shown in eq.(7.1) and eq.(7.2):

$$\widetilde{H(X)} = 705.78 - 55.13x_1 + 173.62x_2 + 4.63x_4 + 125.13x_3 - 0.06x_4^2 - 14.98x_3^2 - 8.89x_1x_2 \quad (7.1)$$

$$\Delta P(X) = 52.07 - 21.40x_1 + 17810.05x_5 + 4.47x_3 + 1.84x_1^2 - 816.65x_1x_5 - 0.47x_1x_3 \quad (7.2)$$

To determine the optimal values of design variables, two approaches are utilized in this work: firstly, maximize the heat transfer coefficient $\widetilde{H}$ using eq.(7.1) and minimizing pressure drop ΔP using eq.(7.2) separately as single objective function optimization. Secondly, utilizing weighted sum and pareto based approach of multi-objective optimization by using both the objective function simultaneously.

7.3.1 First Approach: Single objective optimization

In this approach, the nonlinear optimization problem shown in eq.(7.1) and eq.(7.2) are optimized as two distinctive objective functions of conflicting nature. The heat transfer coefficient must be enhanced through the tube so that maximum heat can be transferred from one component of the system to another while keeping the pressure in flat tube low. These eq.(7.1) and (7.2) represent as two different optimization problem as follows:

Maximize $\widetilde{H} = f_1(X) = 705.78 - 55.13x_1 + 173.62x_2 + 4.63x_4 + 125.13x_3 - 0.06x_4^2 - 14.98x_3^2 - 8.89x_1x_2$ (7.3)

Minimize $\Delta P = f_2(X) = 52.07 - 21.40x_1 + 17810.05x_5 + 4.47x_3 + 1.84x_1^2 - 816.65x_1x_5 - 0.47x_1x_3$ (7.4)

7.3.2 Second Approach: Multi-objective optimization

This optimization problem can be dealt using multi-objective approach also. There may be two formulation of this problem using multi-objective approach: Firstly, single objective scalarization in which weighted sum approach using some weight factor is used to convert both the objective functions into a single objective function. Secondly, using Pareto dominance by determining the number of non-dominated solutions. These solutions may not present the best solutions but provide a set of Pareto optimal solution which are better than other solutions. In this work, both the approach of multi-objective optimization is utilized to determine the optimal value of variable while increasing the heat transfer coefficient ($\widetilde{H}$) and reducing the pressure drop (ΔP).

Weighted sum single objective formulation

In this approach, the basic concept is to reduce or scalarize [170] the unconstrained multi-objective functions $f: R^m \rightarrow R^n$ into a single objective function $f: R^m \rightarrow R$. Generally weighted sum method is more popular for single objective optimization. The weights chosen for this purpose is a $n-$dimensional vector $W = \{w_1, w_1, \dots, w_n\}$ where each of $w_i \in [0,1]$. It can be selected optionally provided that $\sum_{i=1}^{n} w_i = 1$. The values of the weights are assigned to each of the objective function f_i and represents the importance given to it. The reduction or scalarization of the objective functions is done by taking dot product of vector w with corresponding function f_i. Mathematically, the process of single objective optimization can be defined as:

$$F(X) = W.f(X) = \sum_{i=1}^{n} w_i f_i(x) \tag{7.5}$$

In this work, the weight vector can be defined as $W = \{w_1, w_2\}$ as there are two objective functions to be considered as represented by eqs.(7.3) and (7.4) in the form of optimization problems depicted in section 7.3.1.

The objective functions defined in eq.(7.3) and (7.4) are of conflicting nature. The minimization problem shown in eq.(7.4) is converted into maximization problem by changing the sign of negative of pressure drop formulation. By analyzing the importance of the problem, the weight w_1 assigned to function f_1 is 0.6 and w_2 assigned to function f_2 is 0.4. After scalrization of these two functions, the multi-objective problem can be converted into single objective optimization as per eq.(7.6) as follows:

Maximize $F(X) = w_1.f_1(X) + w_2.(-f_2(X))$ (7.6)

Where $f_1(X)$ and $f_2(X)$ are shown in eq.(7.3) and (7.4) respectively such that

$$2 \le x_1 \le 10 \tag{7.7}$$

$$1 \le x_2 \le 5 \tag{7.8}$$

$$0 \le x_3 \le 5 \tag{7.9}$$

$$20 \le x_4 \le 100 \tag{7.10}$$

$$0.002826 \leq x_5 \leq 0.014130 \tag{7.11}$$

Pareto based multi-objective optimization

To determine the optimal values of five design variables, the model obtained using RSM method is optimized as multi-objective optimization problem using multi-objective antlion optimizer (MOALO) [174]. The formulation of multi-objective problem is performed by taking two conflicting objective functions as depicted in eq. (7.3) and (7.4) for maximization of heat transfer value and minimization of pressure drop value. The boundary values of five design variable are utilized as shown in eqs. (7.7)-(7.11).

7.4 Method of Solution: Antlion Optimizer and its variants

In this work, the ants and antlions are initialized randomly. For first approach, two distinct objective function, one for maximization of heat transfer($\widetilde{H}$) and second for minimization of pressure drop(ΔP) as shown in eqs.(7.3) and (7.4) respectively are considered. These objective functions are dealt as two independent optimization problems. The fitness of antlions is determined using the objective functions. The antlions are sorted on the basis of their fitness and best (maximum for heat transfer and minimum for pressure drop) antlion called elite is determined. Then the positions of ants are updated in each generation by taking mean of random walks around elite antlion and around selected antlion using roulette wheel method of GA. After calculating the fitness of each ant, it is compared with the elite antlion and better antlion is kept as elite and carried to the next generation until predefined stopping criteria is satisfied. The optimal values of design variables for best candidate solutions are reported.

For second approach, the problem is utilized as Multi-objective problem which is transformed into single objective problem using weighted sum approach by converting both the problems as maximization problems. The remaining steps are same as described above to determine the fitness of best candidate solution against the optimal values of design variables. Another approach of multi-objective optimization is based on pareto optimal set obtained after applying MOALO [174]. The obtained pareto front investigates the obtained non-dominated solutions for both the objectives.

7.5 Numerical Simulation

RSM is used as a method of modelling and regression analysis for both heat transfer coefficient ($\widetilde{H}$) and pressure drop (ΔP). The accuracy obtained from ANOVA of the model shown in eq. (7.1) for heat transfer $\widetilde{H}$ is defined by a parameter R^2 whose value is 90.03% [159]. This value indicates the high accuracy of the heat transfer model. Similarly, the accuracy for the pressure drop (ΔP) model in eq. (7.2) is 98.39% [159] which also indicates the high accuracy of this model also.

The parameters used to optimize the design variables for determining the optimum value of heat transfer coefficient and pressure drop for both the approaches are chosen as follows: The size of the population is 30 for both ant and antlions. Termination criteria is fixed as 200 maximum number of iterations. Experiments are performed for 30 independent runs out of which the result of best run is reported. The antlions are initialized randomly in such a way that each candidate solution is defined using a five dimensional vector x_1, x_2, x_3, x_4 and x_5. The boundary values of all these parameters are given in Eqs. (7.7)- (7.11). All the experiments have been performed on MATLAB 8.3.0(R2014a) on Intel(R) Core(TM) i5-7200 CPU @ 2.50GHz- 2.71 Ghz with 8GB RAM.

7.6 Discussion of Results

7.6.1 First method: single objective function optimization approach

In this section, the obtained results to maximize the heat transfer coefficient $(\widetilde{H})$ and to minimize the pressure drop(ΔP) are presented. Optimization of objective function for heat transfer coefficient $(\widetilde{H})$ determines the optimal values of five design variables as shown in Table 7.2 which represent the optimized value of pressure drop (ΔP) corresponding to determined design variables for heat transfer. In similar way, Table 7.3 represents the optimized fitness value and design variables for pressure drop. The heat transfer coefficients$(\widetilde{H})$ are also represented in this table corresponding to optimal design variable determined for pressure drop.

Table 7.2 of results exhibit the results obtained to maximize the heat transfer coefficient$(\widetilde{H})$. The results analysis clearly show that the value of heat transfer coefficient is maximized significantly as compared to RSM. The performance order in terms of heat transfer coefficient is OB-L-ALO > OB-C-ALO > OB-ac-ALO > OB-SAC-ALO > ALO > OB-LF-ALO > RSM. However, the pressure drop values corresponding to optimum values obtained using the single objective function to optimize heat transfer coefficient$(\widetilde{H})$ is not significant as compared to RSM. Similarly, Table 7.3 represents the results after minimizing the pressure drop (ΔP). It can be clearly observed from the table that reduction in pressure drop values are acceptable as compared to RSM but the corresponding optimum value is not enhanced as much in comparison to RSM. The performance order of obtained pressure drop values OB-C-ALO < OB-ac-ALO < OB-L-ALO < OB-SAC-ALO <ALO < OB-LF-ALO.

This analysis establishes that the optimization using single objective function is capable of attaining good results for both the coefficient separately but not able to determine good results for other coefficient. All the heat exchangers using flat tubes are having heat transfer and pressure drop parameters. These parameters are conflicting in nature and while one improves the other deteriorates. This implies that this mechanism is acceptable to find the results for individual coefficient (either $\widetilde{H}$ or ΔP)but not for determining both the objectives simultaneously using this approach.

Figure 7.1(a) and (b) represent objective function values and exhibit gradual increment in heat transfer coefficient and reduction in pressure drop parameter respectively as the number of iterations increased. Figure 7.2 and Figure 7.3 represent bar diagram to exhibit the heat transfer coefficient and pressure drop value for proposed ALO algorithms in this thesis.

7.6.2 Second method: Multi-objective optimization

Weighted sum approach

The results obtained using second approach are shown in Table 7.4 in which both the objective functions are taken simultaneously and converted into single objective function using weighted sum approach of multi-objective optimization. The optimum values of five design variables are determined for which transfer coefficient value($\widetilde{H}$) and pressure drop value (ΔP) are determined. The obtained results are compared among all the proposed ALO algorithms in this thesis and RSM method.

The results are exhibited in Table 7.4. It is visible that the optimal values of tube flattening variable (x_1) and inlet volumetric flow rate (x_5) have the lowest values and wall heat flux (x_2) has the highest value of range for all the techniques except OB-LF-ALO. However, other design variable values lie between the given ranges. The heat transfer ($\widetilde{H}$) determined using OB-C-ALO is maximum whereas OB-L-ALO and ALO obtain similar value of heat transfer. The performance order can be visualized for heat transfer coefficient ($\widetilde{H}$) as OB-C-ALO> OB-L-ALO > ALO > OB-SAC-ALO > OB-ac-ALO > OB-LF-ALO >RSM. It can be concluded that all the proposed ALO algorithms outperform the RSM in terms of obtained heat transfer value. Though the pressure drop (ΔP) is minimum in case of RSM however the performance order of obtained pressure drop value can be given as OB-SAC-ALO < OB-L-ALO < OB-C-ALO < OB-ac-ALO

< OB-LF-ALO < ALO . However, OB-C-ALO is able to provide good design values of variables with hear transfer value as 1725.3 and pressure drop value as 76.8106. Figure 7.2 and Figure 7.3 show bar diagram of heat transfer coefficient $(\widetilde{H})$ and pressure drop value (ΔP).

Pareto based multi-objective optimization approach

To determine the pareto front, size of population is fixed at 100 with maximum iteration 100 using multi-objective antlion optimizer MOALO. Two conflicting objective functions, one to maximize heat transfer coefficient $(\widetilde{H})$ and second to minimize pressure drop value (ΔP) are taken simultaneously to determine the optimal value of five design variables.

Figure 7.4 depict pareto optimal front of both the objective functions obtained by applying multi-objective antlion optimizer(MOALO). It can be clearly observed from the figure that no point have dominancy over another point i.e. there are no two points having same objective function value. In other words, it can be stated that while moving from one point to another point, the value of one objective function gets better and other value gets worse. This reflects the conflicting behaviour of both the objective functions.

Figure 7.4 depicts five optimal points designated as A, B, C, D and E. The corresponding design variables to these points are represented in Table 7.5. These points depict unique properties with respect to objective function values of both the functions. Points A and points E show minimum pressure drop (ΔP) and maximum heat transfer $(\widetilde{H})$ values respectively. Point B can be represented as break point which show that up to point B, the value of heat transfer increases significantly (36.1%) while pressure drop value increases at low pace (1.77%). Similarly, point D represents the break point up to which the heat transfer values enhances at much faster rate as compared to pressure drop value but after it, pressure drop value increases at much faster rate(27.61%) than heat transfer value(7.43%). Point C is ideal point which reflects the optimal values of five design variable to determine both the objective function values. This point is obtained as optimal candidate solution using MOALO.

7.7 Conclusion

In this chapter, a multi-objective real life application to optimize nanofluid flow in flat tube to maximize heat transfer $(\widetilde{H})$ and to minimize pressure drop (ΔP) value is solved. The model is taken from the literature and is performed using RSM method while using Al_2O_3-water nanofluid flowing in horizontal flat tube.

The objective functions for both these parameters are of conflicting nature. A variety of method are applied to solve this optimization problem. First, single objective approach is utilized in which both the functions are solved separately using classical ALO ant its proposed modified variants in thesis for maximizing heat transfer and minimizing pressure drop value. The five design variables: tube flattening variable (x_1), wall heat flux (x_2), fraction of nanoparticle volume (x_3), nanoparticle diameter (x_4) and inlet volumetric flow rate (x_5) are determined from each of the functions and used to obtain both the values as shown in tables of results. The convergence curves and bar diagrams for ($\widetilde{H}$) and (ΔP) are drawn and analyzed.

Then the two multi-objective approaches are used to optimize the problem: first method is weighted sum approach in which both the functions are converted into single objective function by applying appropriate weight to each of function. The results are obtained and compared using classical ALO and its proposed variants in thesis. In second method, pareto based approach of multi-objective optimization is utilized to optimize both the objective function. Five unique points A, B, C, D and E over the pareto front are determined which consist important design information about the optimization of this problem and able to give the ideal solution using MOALO.

Table 7.1: Design variables

Design variables	Lower bound	Upper bound
Flattening variable x_1(mm)	2	10
Wall heat flux x_2(kWm^{-2})	1	5
Nanoparticle volume x_3(%)	0	5
Nanoparticle diameter x_4(nm)	20	100
Inlet volumetric flow rate x_5(m^{3}h^{-1})	0.002826	0.014130

Table 7.2: Heat transfer coefficient and pressure drop using first single objective function for heat transfer coefficient

Method	x_1(mm)	x_2(kWm^{-2})	x_3(%)	x_4(nm)	x_5(m^{3}h^{-1})	$\widetilde{H}$(W(m^{2}k)$^{-1}$)	ΔP(Pa)
OB-C-ALO	2	5	4.17928	38.5983	0.013294	1725.3473	246.4366
OB-LF-ALO	2.4252	4.5816	2.61889	41.4024	0.00782144	1582.5605	143.5237
OB-ac-ALO	2	5	4.06	38.13	0.0028	1725.1	76.2567
OB-SAC-ALO	2	5	4.37	38.18	0.0030	1724.8	80.5863
OB-L-ALO	2	5	4.17653	38.5828	0.0106454	**1725.3474**	203.5811
ALO	2	5	3.43747	45.482	0.00975376	1714.3088	186.5484
RSM	5.5	5	3.63	36.16	0.0028	1361.5	**6.01**

Table 7.3: Heat transfer coefficient and pressure drop using second single objective function for pressure drop

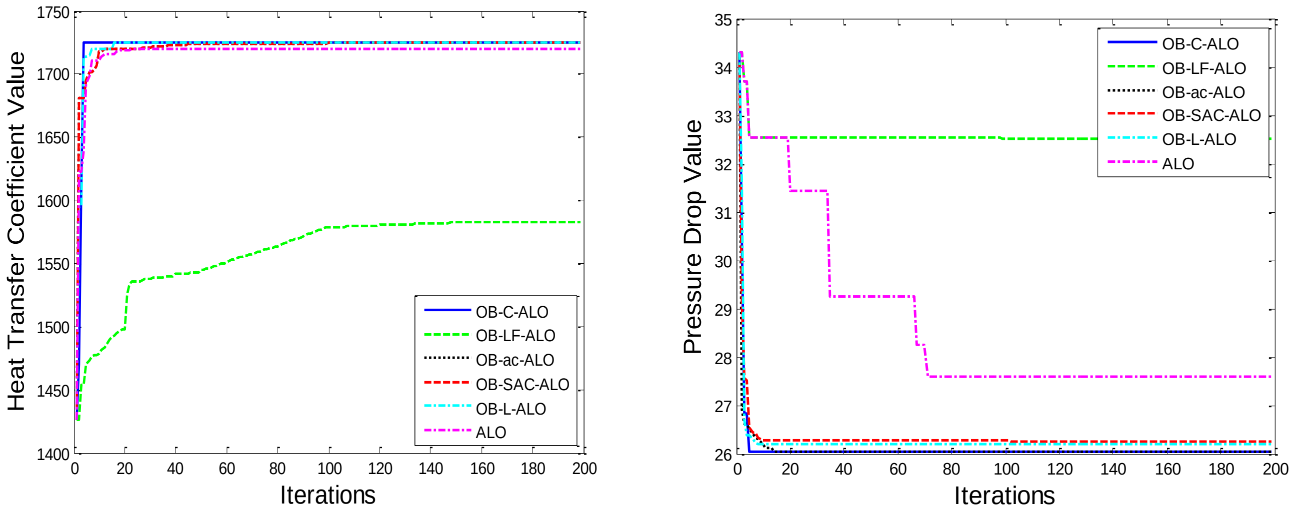
Method	x_1(mm)	x_2(kWm^{-2})	x_3(%)	x_4(nm)	x_5(m^{3}h^{-1})	$\widetilde{H}$(W(m^{2}k^{-1}))	ΔP(Pa)
OB-C-ALO	6.44235	2.0330	0	23.3362	0.002826	662.5255	**26.034**
OB-LF-ALO	5.81223	4.56759	3.19062	35.5005	0.002826	1277.9	32.3107
OB-ac-ALO	6.34096	3.78811	0	21.3019	0.002826	871.7560	26.053
OB-SAC-ALO	6.19886	2.23985	0.0731429	21.3165	0.002826	709.9903	26.257
OB-L-ALO	6.13659	2.37779	.000007556	21.1945	0.002826	721.7621	26.2061
ALO	6.3398	3.56317	1.03422	20	21.1945	956.0704	27.5947
RSM	5.5	5	3.63	36.16	0.0028	1361.5	**6.01**

Table 7.4: Heat transfer coefficient and pressure drop using weighted sum approach of multi-objective optimization

Method	x_1(mm)	x_2(kWm^{-2})	x_3(%)	x_4(nm)	x_5(m^{3}h^{-1})	$\widetilde{H}$(W(m^{2}k^{-1}))	ΔP(Pa)
OB-C-ALO	2	5	4.09775	38.5839	0.002826	1725.3	76.8106
OB-LF-ALO	2.67394	4.52164	3.20429	31.4211	0.00414678	1569.3	83.0989
OB-ac-ALO	2	5	4.36375	33.523	0.00282601	1723.3	77.7497
OB-SAC-ALO	2	5	43.87525	38.4891	0.002826	1724.0	76.0251
OB-L-ALO	2	5	4.08667	38.4703	0.002826	1725.2	76.7714
ALO	2	5	4.09225	38.6863	0.002826	1725.2	231.4265
RSM	5.5	5	3.63	36.16	0.0028	1361.5	6.01

Table 7.5: Objective function and design variables values of pareto front

Points	x_1(mm)	x_2(kWm^{-2})	x_3(%)	x_4(nm)	x_5(m^{3}h^{-1})	$\widetilde{H}$(W(m^{2}k^{-1}))	ΔP(Pa)
A	6.4552	2	0	20	0.002826	650.9680	26.0343
B	6.1150	4.5922	0.1714	27.0491	0.002826	1018.7	26.5048
C	**4.0208**	**5**	**3.2717**	**39.3917**	**0.002826**	**1511.8**	**45.2654**
D	3.2558	5	3.6129	35.9827	0.002826	1595.1	55.3383
E	2	5	3.8614	42.0044	0.002856	1723.2	76.4549



(a) Heat transfer coefficient value (b) Pressure drop value

Figure 7.1: Curves showing predicted values w.r.t no. of iterations

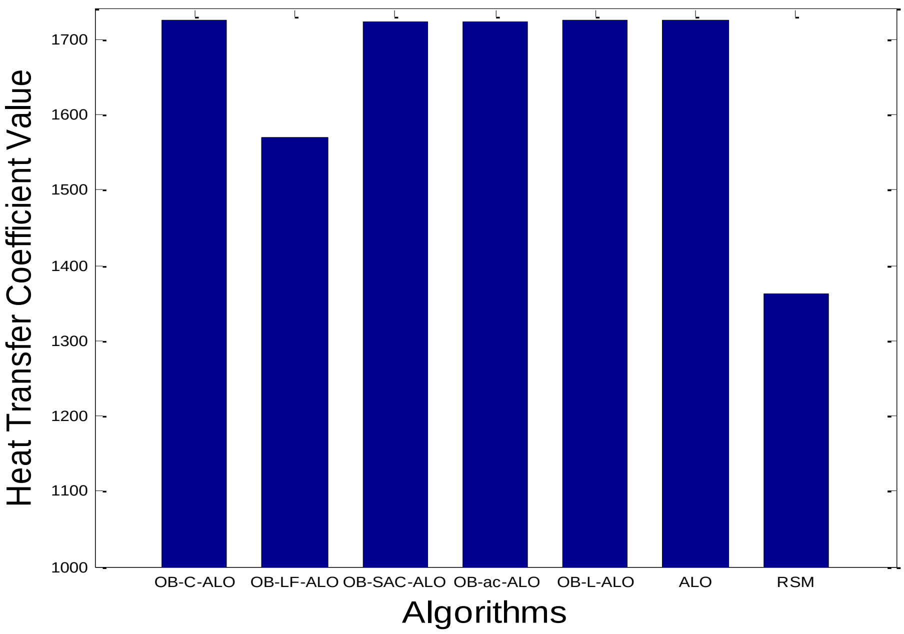


Figure 7.2: Bar diagram of heat transfer value ($\widetilde{H}$)

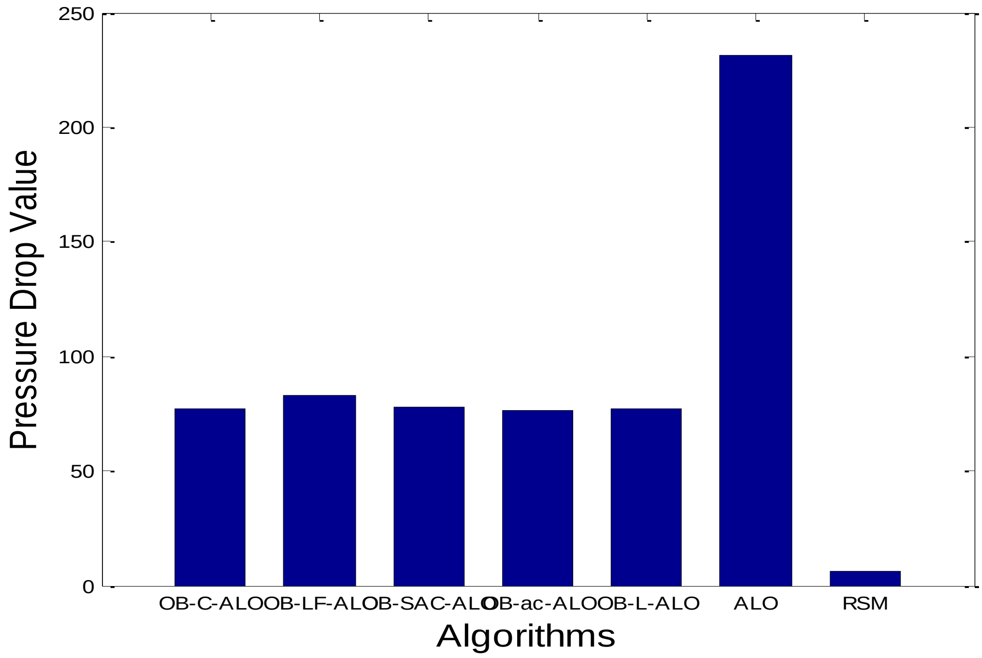


Figure 7.3: Bar diagram of pressure drop (ΔP)

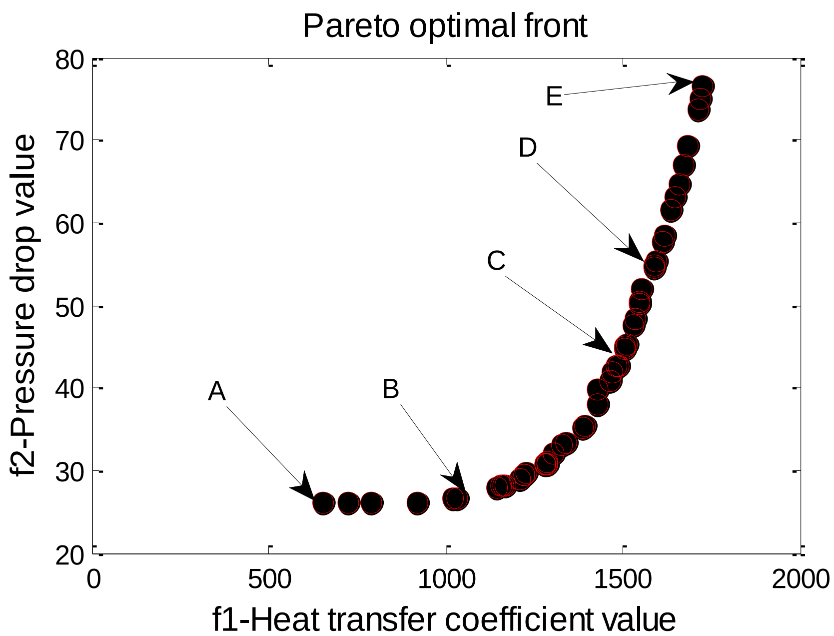


Figure 7.4: Pareto optimal points for heat transfer($\widetilde{H}$) and pressure drop(ΔP)

CHAPTER 8

Process Optimization of Biodiesel Production using Antlion Optimizer and it's Variants

The objective of this chapter is to evaluate the performance of ALO and its proposed variants for solving the real life problem to optimize the process of biodiesel production.

8.1 Introduction

The continuous consumption of energy, particularly fuel obtained from petroleum products is becoming uncontrollable and contributing to the extreme build-up of greenhouse gases in the environment. The production of biodiesel from vegetable oils is definitely an alternative source to reduce the ill effects of greenhouse gases. Usually the production of biodiesel (alkyl ester) is performed with transesterification process using response surface methodology. In this chapter, the problem of optimizing the production of biodiesel (methyl ester) is performed using modified versions proposed in this thesis namely OB-C-ALO, OB-LF-ALO,OB-ac-ALO,OB-SAC-ALO and OB-L-ALO including classical ALO. The optimization problem is modelled into a single objective optimization problem and a combination of three parameters called temperature (x_1), methanol/oil ratio (x_2) and dolomite catalyst concentration (x_3) has been optimized to maximize the production of biodiesel (methyl ester). The results indicate significant improvement in production of methyl ester (biodiesel) yield using classical ALO and its modified versions.
The organization of chapter is as follows:

Section 8.2 states the motivation and literature review. Section 8.3 depicts the formulation of the problem. Section 8.4 establishes the problem as optimization model and states the objective function of the problem. Section 8.5 demonstrates method of solution. Section 8.6 depicts the computational study and discussion of results using RSM and variants of classical ALO. Section 8.7 concludes the chapter.

The content of this chapter is published as:

- **Dinkar, S. K., & Deep, K. (2019). Process optimization of biodiesel production using antlion optimizer. Journal of information and optimization sciences (Taylor & Francis). DOI: 10.1080/02522667.2018.1491821. (ESCI).**

8.2 Motivation and related literature study

The biggest concern of the world is to address the problem of global warming which is affecting the environment critically [175-176]. A lot of research is going on to reduce the malefic effect of global warming which is majorly contributed by pollution due to industries and consumption of fuel. In recent years, great emphasis is on production of biodiesel fuel which is beneficial to environment in comparison to other available fuels. As the fossil fuel resources are declining very rapidly in the environment so there is high attraction towards the production and use of renewable energy resources [176].

Usually biodiesel is produced through a chemical reaction called transesterification process which is a process of substituting the organic group of ester with the organic group of an alcohol. This reaction is then catalyzed using homogeneous alkaline catalysts as this process produces high amount of alkyl ester in a short reaction time [177]. The catalyst performance is majorly dependent over the various operating conditions such as reaction temperature, amount of catalyst, methanol/oil ratio molar ratio as well as the temperature heat treatment used in natural dolomitic rock [178].

Sometimes the heterogeneous catalysts can be useful as synthesis process of these catalysts as it may be beneficial to add to the supplementary cost to the ultimate product. If heterogeneous catalyst is used safely for production of biodiesel then it can be effective for various industrial applications [179-180].

In transesterification process, original ester is reacted with alcohol which is called alcoholises (Figure 8.1) [177]. Numerous parameters are included like type of catalyst, vegetable oil/alcohol molar ratio, reactants purity, temperature etc. having impact on transesterification by reaction.

As given in Santos et al. [179], the objective is to maximize the production of methyl ester (biodiesel) yield by transesterification of soybean oil with ethanol using several parameters which includes alcohol/vegetable oil molar ratio, catalyst and temperature which affect the transesterification process and then to develop an approach which establishes relationship among the variables (ethanol-to-oil ratio, catalyst concentration and temperature) and the response (methyl esters).

In this chapter, the experimental values are being taken from [179] where dolomite as a heterogeneous catalyst is being used for transesterification process. It is ecologically suitable substantial with high basicity and low cost and capable substitute catalyst for biodiesel production. To obtain the optimal conditions for biodiesel production central composite rotable design (CCRD) and response surface methodology (RSM) is used [179].

Energy sources are in high demand since as long as man's need of energy even from increasing industrialization commencing in early 19th century. This is because of the development of various machines used to utilize the energy kept in a variety of power [181].The spark ignition and compression ignition (diesel) engines are main study of the combustion engines. In spite of fossil sources, biogenic resources also found useful mainly ethanol in case of spark-ignition engine [182] and vegetable oils for the compression ignition engine [183]. The Belgian patent 422877 perhaps is the first report where the term biodiesel is being used [184-185]. The alkyl esters extracted from the vegetable oils and animal fats are termed as biodiesel and documented for the first time [181, 186]. This process was again reinvented after forty years later [187].

The recent development for producing esters (biodiesel) is based on use of catalytic reaction termed as hydro deoxygenation from biogenic resources [188-189]. These catalysis may be homogeneous or heterogeneous and the reactions are termed as transesterification process. Extensive research is employed recently on various catalysts for transesterification process. The various new heterogeneous catalysts and their chemistry are being reviewed in [190-191]. Enzymatic catalysis [192-194] , whole-cell biocatalysts [195], ionic liquids [196] and dolomite [179] etc. are examples of heterogeneous catalysts. The optimization of alky ester production is usually functioned using Response Surface Methodology (RSM) [177, 179]. However, Fayyazi et al. [197] used genetic algorithm approach in ultrasonic system for optimizing biodiesel production which expresses significance of the use of nature inspired optimization algorithm for optimizing this process.

8.3 Problem Formulation: Design and Optimization of Parameters

For the optimization [179] ,the problem is modelled as nonlinear optimization problem in which decision variables are temperature of reaction, catalyst concentration and methanol/oil ratio (molar ratio) as shown in Table 8.1. The first column depicts the independent variables used in optimization process, second column shows the coding of range of values of independent variables and third and fourth columns indicates the lower and upper bound for variables.

A set of 17 experiments is considered by Santosh et. al [179] for full factorial central composite design as depicted in Table 8.2. For the given range of independent variables, low order polynomial is employed for modelling.

The data pertaining to three independent variables and one response variable is modelled to produce a second order polynomial regression equation as the function of independent variables to yield methyl ester as biodiesel fuel. The equation can be defined as:

$$y_{me} = b_0 + \sum_{i=1}^{3} b_i x_i + \sum_{i=1}^{3} b_{ii} x_i^2 + \sum_{i\neq j=1}^{3} b_{ij} x_i x_j \quad (8.1)$$

where y_{me} represents methyl ester yield, b_0 is a constant, b_i, b_{ii} and b_{ij} are linear, quadratic and interactive coefficients.

The efficiency of the biodiesel production is majorly influenced by three independent parameters namely temperature (x_1), methanol to oil ratio (x_2) and concentration of catalyst (x_3). The combined effect of these three parameters is also responsible for efficiency of biodiesel production called interaction factors. As per Santos et al. [179], after performing experiments with different physical parameters as shown in Table 8.2, multiple regression analysis was performed and fitted to polynomial Eq. (8.2). It gives the regression equation as the function of three independent parameters to form methyl ester (biodiesel) which is defined as:

$$y_{me}(\%) = 37.45 + 0.42x_1 + 4.37x_2 - 0.16x_2^2 + 24.26x_3 - 1.53x_3^2 + 0.001x_1x_2 - 0.24x_1x_3 - 0.44x_2x_3 \quad (8.2)$$

8.4 Optimization of Biodiesel (Methyl ester) production

The maximization of biodiesel (Methyl ester) production is performed using proposed variants of classical ALO in this thesis in place of transesterification process of soybean oil with ethanol. The optimal values of three independent variables are determined in such a way that the production of biodiesel gets maximized. The objective function is given in Eq. (8.3) which can be defined as optimization problem [179]:

$$max\ f(x_1x_2x_3) = 37.45 + 0.42x_1 + 4.37x_2 - 0.16x_2^2 + 24.26x_3 - 1.53x_3^2 + 0.001x_1x_2 - 0.24x_1x_3 - 0.44x_2x_3 \quad (8.3)$$

Such that

$$55 \leq x_1 \leq 65, \quad (8.4)$$

$$6 \leq x_2 \leq 15, \quad (8.5)$$

$$0.6 \leq x_3 \leq 2.0 \quad (8.6)$$

8.5 Method of Solution: Antlion Optimizer and its modified variants

In this chapter, the classical ALO [56] and its modified variants proposed in this thesis are applied to the application of maximization of biodiesel (methyl ester) production. Three dimensional population of ants and antlions are randomly generated where first, second and third dimension represents the independent variables namely temperature (x_1), methanol to oil ratio (x_2) and concentration of catalyst (x_3) respectively. Then the fitness of each antlion is determined using the objective function shown in eq. (8.3). The fitness values for all antlions are sorted to determine the best candidate solution having maximum fitness. Then two random walks are performed: (1) around selected antlion using roulette wheel selection method and (2) around elite antlion. In classical ALO, the position of ant is updated by taking the average of both of these random walks. In modified versions of classical ALO, the updated strategies are applied as per corresponding version of classical ALO to determine the updated positions of each ant. The fitness of each ant is determined and compared with the fitness of best candidate solution as per eq. (1.13), then the fitter solution is carried forward to next generation.

8.6 Computational Results and Discussions

The observed values depicted in Table 8.2 are being taken from [179] and predicted values shown in column 6 of Table 8.2 are computed using ALO having same values chosen for three independent variables as in [179]. Fitness of the model for predicted values computed with ALO algorithm is verified with the coefficient of determination R^2 which comes to be 0.9929 i.e. 99.2%. This model is statistically significant as per the F-test with 95% of significance level. The value of F is 108.03 and the p-value is 0.0001 which is significantly low. Figure 8.2 shows the linear correlation plot between observed and predicted values computed using ALO.

This optimization problem having objective function shown in eq. (8.3) is solved by employing ALO and its modified versions proposed in previous chapters taking 30 independent runs with population size as 30 and the maximum number of iteration as 500. The antlions (population) is initialized in such a way that each individual is of three dimensional vector corresponding to three independent parameters x_1, x_2 and x_3. The boundary values of all these three parameters are given in eqs. (8.4), (8.5) and (8.6) respectively. All the experiments have been performed on MATLAB R2014 on Intel(R) Core(TM) i5-7200 CPU @ 2.50GHz 2.71Ghz with 8GB RAM.

The optimum values obtained using classical ALO and its proposed modified variants are shown in Table 8.3 for three independent variables. After performing 30 independent runs, the values of three independent variable, the reported values in Table 8.3 are taken from the best run. It can be observed from the table of results that three modified versions of classical ALO namely OB-ac-ALO, OB-SAC-ALO an OB-L-ALO are able to produce the maximized methyl ester (biodiesel) yield including classical ALO. The optimum value is same for all four algorithm i.e. 97.4% with values of three independent variables as: temperature 65^0 , Methanol/oil ratio 12.62 and catalyst (dolomite) concentration is 1.01 %(w/v). The obtained value is significantly better than the yield produced by response surface methodology. However, OB-C-ALO produces 97.46% yield whereas OB-LF-ALO produces 97.27%. So the performance order for all the proposed algorithms can be devised as OB-ac-ALO = OB-SAC-ALO = OB-L-ALO = ALO $>$ OB-C-ALO $>$ RSM $>$ OB-LF-ALO.

Also Figure 8.3 is drawn to demonstrate the methyl ester(biodiesel) yield as the number of iterations increased. The values on horizontal axis represent number of iterations and values on vertical axis denote the methyl ester yield. It is visible from the figure that production of yield increases up to 200 iterations and then it becomes stable till the last iteration. So,it indicates that the maximum value of yield can be achieved up to 200 iteration which takes only 6000 function evaluations for the production of biodiesel. Figure 8.4 exhibits the comparison of obtained methyl ester (biodiesel) yield for all modified versions of classical ALO and RSM method. It can be concluded that the four algorithms namely ALO, OB-ac-ALO, OB-SAC-ALO and OB-L-ALO are able to produce same amount of biodiesel.

8.7 Conclusion

In this chapter, the regression equation demonstrating the relationship among three independent variables namely temperature (x_1), methanol to oil ratio (x_2) and concentration of catalyst (x_3) is successfully optimized for maximizing biodiesel production using antlion optimizer (ALO) and its proposed modified versions in this thesis namely OB-L-ALO,OB-ac-ALO,OB-SAC-ALO,OB-LF-ALO and OB-C-ALO. The optimal yield of biodiesel comes out to be **97.47%** at parameter values as **65**°C temperature,**12.62:1** methanol to oil ratio and **1.01 %(w/v)** catalyst concentration which is significantly high than the biodiesel yield using RSM. Though the response surface methodology (RSM) is a popular optimization techniques for such kind of process but the success of ALO and its variants indicate the diverse capability of evolutionary algorithms for solving such types of real optimization problems. In literature from where the model is chosen, the fitness of the model is also verified statistically with the coefficient of determination R^2 and statistical significance with F-test which shows the predictability of the model. As ALO and its proposed variants has successfully obtained the optimum yield of methyl ester, it shows that the modified versions of classical ALO are proficient in search and can be employed to more complex real world applications having compound multimodal functions. The performance order of the proposed variants comes out to be as OB-L-ALO $>$ OB-SAC-ALO $>$ OB-C-ALO $>$ ALO $>$ OB-ac-ALO $>$ RSM $>$ OB-LF-ALO. The novelty in this work is optimizing the process to obtain biodiesel from renewable energy sources in place of using the fossil fuels by employing evolutionary algorithm.

Table 8.1: Range and levels coding of design variables for biodiesel production

Independent Variables	Range and Level			Lower Bound	Upper Bound
	-1	0	1		
Temperature(x_1)	55	60	65	55	65
Methanol/oil ratio(x_2)	6:1	10.5:1	15:1	6:1	15:1
Catalyst concentration(x_3)	0.6	1.3	2.0	0.6	2.0

Table 8.2: Full factorial central composite design and results for transesterification of soybean oil using dolomite [179]

Run	Independent Variables			Observed Values of Methyl Ester Yield (wt %)	Predicted Values of Methyl Ester Yield (wt %)
	x_1	x_2	x_3		
1	55	6	0.6	85.5	86.1052
2	65	6	0.6	89.9	88.9732
3	55	15	0.6	93.6	93.7102
4	65	15	0.6	96.1	96.7402
5	55	6	2.0	92.6	92.324
6	65	6	2.0	91.6	91.832
7	55	15	2.0	93.2	94.385
8	65	15	2.0	94.3	94.055
9	60	10.5	0.6	93.7	94.6222
10	60	10.5	2.0	95	97.389
11	60	15	1.3	94.4	95.4723
12	60	6	1.3	89.3	90.5583
13	65	10.5	1.3	94.6	96.8898
14	55	10.5	1.3	95.6	95.6208
15	60	10.5	1.3	97.3	96.2553
16	60	10.5	1.3	96.5	96.2553
17	60	10.5	1.3	97.1	96.2553

Table 8.3: Results Comparison of biodiesel yield using ALO algorithms

Algorithm	Independent Variables			Optimum Value of Methyl Ester (biodiesel) Yield (wt %)
	x_1	x_2	x_3	
OB-C-ALO	65	12.84	1.00	97.46
OB-LF-ALO	64	12.54	1.01	97.27
OB-ac-ALO	65	12.32	1.02	97.32
OB-SAC-ALO	**65**	**12.62**	**1.01**	**97.47**
OB-L-ALO	**65**	**12.62**	**1.01**	**97.47**
ALO	65	12.71	1.2	97.33
RSM[179]	60	10.5	1.3	97.3

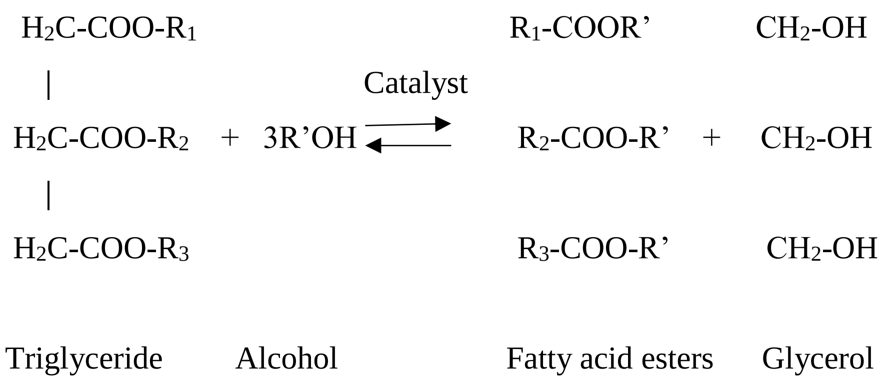


Figure 8.1: General Transesterification reaction

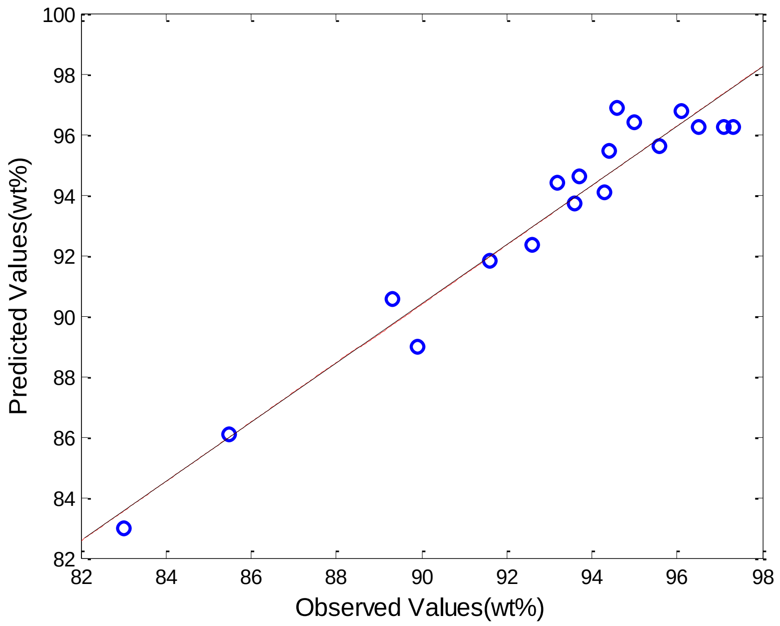


Figure 8.2: Linear correlation plot between observed and predicted values

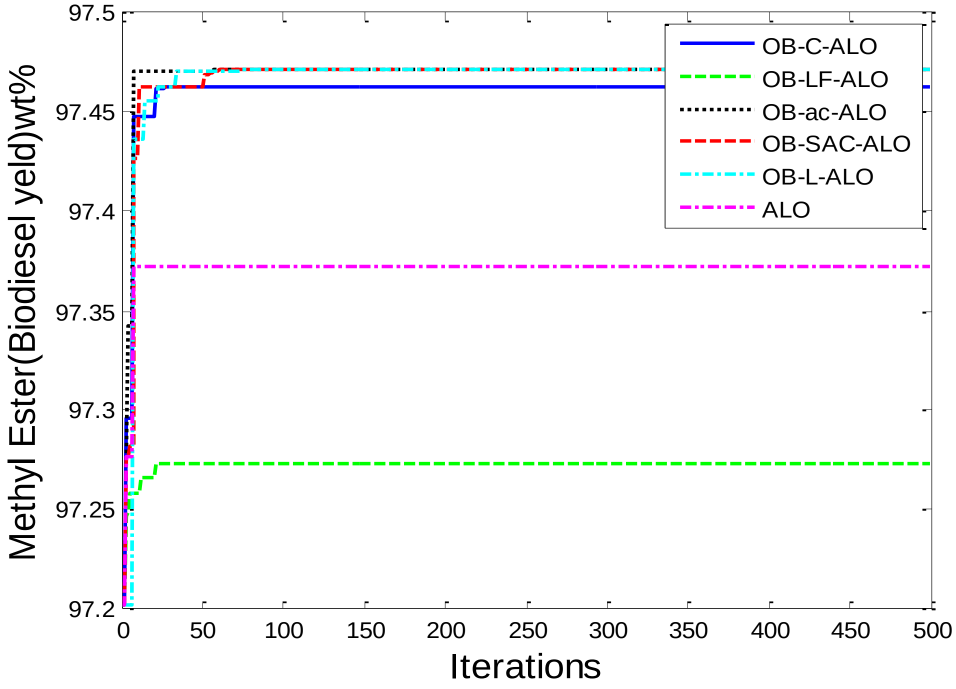


Figure 8.3: Methyl Ester (Biodiesel) yield

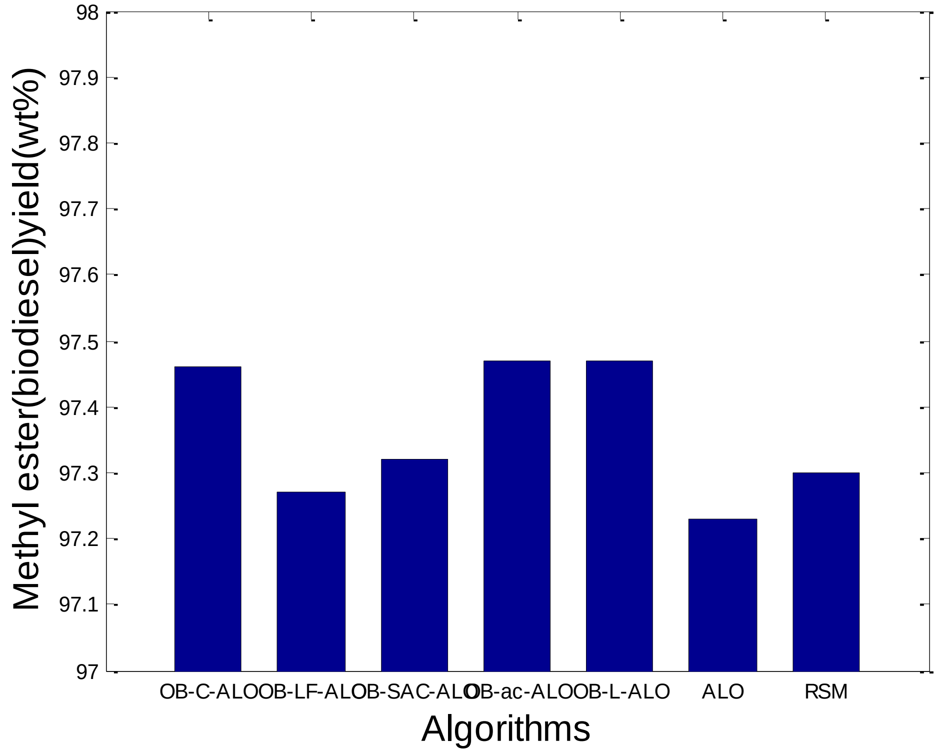


Figure 8.4**:** Comparative bar chart diagram of obtained Methyl Ester (Biodiesel) yield

CHAPTER 9

Conclusions and Future scope

This chapter contains the concluding observations about the thesis. The organization of this chapter is as follows: Section 9.1 demonstrates the conclusive remarks to outline the studies performed in this thesis. Section 9.2 represents some futuristic directions of research related to this study.

9.1 Conclusions

The performance of any optimization algorithm depends on its efficient searching capability in such a way that the search region is explored as much as possible during initial phase and then converges towards optima during later phase of searching. The overall purpose of this thesis is to design and modify the Antlion optimizer algorithm (ALO) to propose new variants with an objective to acquire appropriate balance between diversity at early phase and exploitation at later phase during evolutionary process. These proposed variants are then applied over some real world applications. To accomplish this goal, five different variants are proposed: OB-*L*-ALO to enhance the exploration and convergence, OB-ac-ALO and OB-SAC-ALO to accelerate the convergence using acceleration parameters, OB-LF-ALO and OB-C-ALO by improving exploration, exploitation and acceleration parameters. The performance of the proposed variants are evaluated over a set of 31 benchmark problems having different difficulty level characteristics. These problems include 23 state-of-the-art problems (7 unimodal, 6 multimodal, 10 fixed dimension multimodal) and a set of 8 complex composition functions taken from IEEE CEC 2014 benchmark problem set.

In the second part of the thesis, the performance of proposed variants is investigated on three real world applications of different fields. In first real world application, from the field of Electrical Engineering three single input single output problems of order reduction of complex linear time invariant (LTI) systems are used to evaluate the performance. The second real world application from the field of Computational Fluid Dynamics (CFD) is to optimize two conflicting objective functions: maximizing heat transfer coefficient and minimizing pressure drop value using Nanofluid flow in flat tube. The third problem from the field of Mechanical Engineering is to optimize the process of biodiesel production to address the energy conservation and recycling.

The chapter wise concluding observations of the thesis are as follows:

Chapter 1 introduces the field of Nature Inspired Optimization and presents the conceptual details, operators, modelling and computational steps of classical ALO. Then the recent developments in designing modified ALO variants and it's applications in different areas and related literature are discussed. At the end of the chapter, the organization of thesis is discussed in brief.

Chapter 2 proposes a novel Opposition Based Laplacian Antlion Optimizer (OB-L-ALO) for unconstraint global optimization. This version of classical ALO is useful to address the drawback of premature convergence and inability to avoid entrapment into local optima. The performance of proposed OB-L-ALO is verified over a set of 31 benchmark problems of varying difficulties containing 23 state-of-the-art problems (a set of unimodal, multimodal and fixed dimension multimodal functions) and 8 IEEE CEC 2014 composition functions. This set is reproduced in Appendix I of this thesis. The performance of proposed variant is compared with classical ALO. The obtained numerical results and analysis establish that the proposed OB-L-ALO outperforms classical ALO over 6 out of 7 unimodal functions, all 6 multimodal functions,9 out of 10 fixed dimensional multimodal functions and 6 out of 8 composition functions. It can be concluded that the proposed OB-L-ALO improves the performance of classical ALO over all categories of problems.

Chapter 3 proposes two variants namely Opposition based ALO using varying Acceleration Coefficient (OB-ac-ALO) and Opposition based ALO using Sine Acceleration Coefficient (OB-SAC-ALO) to accelerate the convergence of opposition based ALO. The performance of both the proposed variants is investigated over same set of benchmark functions as applied in chapter 2 and reproduced in Appendix I of this thesis. Both the variants are compared and analyzed with the classical ALO. The performance order of proposed variants is given for different categories of problems as: for unimodal functions, OB-ac-ALO $>$ OB-SAC-ALO $>$ ALO, for multimodal functions OB-ac-ALO $>$ OB-SAC-ALO $>$ ALO, for fixed dimensional multimodal functions OB-SAC-ALO $>$ OB-ac-ALO $>$ ALO and for composition functions OB-ac-ALO $>$ OB-SAC-ALO $>$ ALO. However both the proposed variants outperform classical ALO in most of the problem but it can be concluded that OB-ac-ALO performs better than OB-SAC-ALO and classical ALO.

Chapter 4 proposes another extended variant namely Opposition based ALO using Lévy flight distribution (OB-LF-ALO). The performance of proposed OB-LF-ALO is tested on same set of benchmark problems as utilized in chapter 2 and 3 and compared with classical ALO which is reproduced in Appendix I of this thesis. The evaluation of numerical results and analysis authorize that the proposed OB-LF-ALO outperforms classical ALO over 6 out of 7 unimodal functions, 5 out of 6 multimodal functions, 9 out of 10 fixed dimensional multimodal functions and 3 out of 8 composition functions. It can be concluded that the proposed OB-LF-ALO improves the performance of classical ALO in majority of the problems except composition functions.

Chapter 5 is divided into two parts. The first part introduces an extended modified variant to overcome the limitation of premature convergence and entrapment in local optima. In this part, the modification is proposed as opposition based antlion optimizer using Cauchy distribution (OB-C-ALO) after applying three strategies to classical ALO. OB-C-ALO is verified using the same set of benchmark problems as used in previous chapters and compared with classical ALO. The numerical results and analysis show that the proposed OB-C-ALO performs better than classical ALO over 6 out of 7 unimodal functions, all multimodal and fixed dimensional multimodal functions, and 6 out of 8 composition functions. It can be established that the proposed OB-C-ALO improves the performance of classical ALO in all categories of problems.

The second part of the chapter presents the performance comparison in terms of obtained results and analysis among all the five proposed variants of classical ALO i.e. OB-*L*-ALO, OB-ac-ALO, OB-SAC-ALO, OB-LF-ALO and OB-C-ALO. Three variants namely OB-C-ALO, OB-LF-ALO and OB-ac-ALO obtain exact global optima for eleven benchmark functions. It can be concluded that OB-C-ALO and OB-ac-ALO performs better for unimodal functions, OB-C-ALO, OB-ac-ALO and OB-LF-ALO show enhanced performance for multimodal functions than other proposed variants. For fixed dimension multimodal functions, OB-*L*-ALO performs better than other variants whereas OB-C-ALO outperforms other variants for compositions functions. The overall performance order can be concluded as OB-C-ALO > OB-ac-ALO > OB-LF-ALO > OB-SAC-ALO > OB-*L*-ALO > ALO.

In Chapter 6, the performance of proposed variants of classical ALO is investigated over a real world complex application of model order reduction of linear time invariant system in the field of control system. The performance of these algorithms are investigated by applying on three single input single output (SISO) systems including two four and one eight order problem of different characteristics. The obtained results are analysed using step response in terms of time and frequency diagrams. For first problem, the performance order is concluded as OB-SAC-ALO $>$ OB-ac-ALO $>$ OB-L-ALO $>$ ALO $>$ OB-C-ALO $>$ OB-LF-ALO. For second and third problems, the performance order of the proposed variants are OB-SAC-ALO $>$ OB-ac-ALO $>$ ALO $>$ OB-L-ALO $>$ OB-C-ALO $>$ OB-LF-ALO and OB-C-ALO $>$ OB-ac-ALO $>$ OB-SAC-ALO $>$ OB-L-ALO $>$ ALO $>$ OB-LF-ALO respectively.

Chapter 7 attempts to determine optimal values of two conflicting objective functions i.e. heat transfer coefficient $(\widetilde{H})$ and pressure drop(ΔP) parameters. Two approaches are utilized: First, single objective approach for both the objective functions separately and secondly, multi-objective approach in which both the objective functions are optimized simultaneously to determine objective function values of heat transfer coefficient and pressure drop values: (i) using weighted sum approach of multi-objective optimization using classical ALO and its proposed variants (ii) Pareto based multi-objective optimization using multi-objective antlion optimizer (MOALO). The performance orders in terms of heat transfer coefficient and pressure drop values are OB-L-ALO $>$ OB-C-ALO $>$ OB-ac-ALO $>$ OB-SAC-ALO $>$ ALO $>$ OB-LF-ALO and OB-C-ALO $<$ OB-ac-ALO $<$ OB-L-ALO $<$ OB-SAC-ALO $<$ ALO $<$ OB-LF-ALO respectively. For weighted sum of multi-objective optimization approach, the performance order can be concluded for heat transfer coefficient $(\widetilde{H})$ as OB-C-ALO> OB-L-ALO > ALO > OB-SAC-ALO > OB-ac-ALO > OB-LF-ALO and for pressure drop value as OB-SAC-ALO < OB-L-ALO < OB-C-ALO < OB-ac-ALO < OB-LF-ALO < ALO. However, it can be concluded that OB-C-ALO is able to provide good design values of variables with hear transfer value as 1725.3 and pressure drop value as 76.8106.

In Chapter 8, maximization of biodiesel production using Antlion Optimizer (ALO) and its proposed modified is performed. For this purpose, the regression equation demonstrating the relationship among three independent variables namely temperature (X_1), methanol to oil ratio (X_2) and concentration of catalyst (X_3) is successfully optimized. The optimal yield of biodiesel comes out to be 97.47% for OB-L-ALO and OB-SAC-ALO at parameter values as 65℃ temperature,12.62:1 methanol to oil ratio and 1.01 %(w/v) catalyst concentration which is significantly high than the biodiesel yield using RSM. The performance order of the proposed variants can be concluded as OB-ac-ALO $=$ OB-SAC-ALO $=$ OB-L-ALO $=$ ALO $>$ OB-C-ALO $>$ RSM $>$ OB-LF-ALO.

9.2 Future Work

Though the five modified versions of classical ALO are designed and verified over a wide variety of benchmark problems as well as real life applications, but research is an ongoing process. Still there are some suggested directions for future research in this area which are as follows:

1. The proposed algorithms can be extended as per the different constraint handling techniques available in literature so as to solve the constrained optimization problems.

2. The proposed algorithms can be further enhanced to solve multi-objective and many-objective optimization problems.

3. Techniques suggested in thesis can be applied to multi-objective algorithms such as MOALO, MOPSO, NSGA-II, NSGA-III etc.

4. The performance of these algorithms can be stretched over large scale optimization problems.

5. These algorithms are computationally extensive, thus the parallelized version of these algorithm can be developed to reduce the computational cost.

6. The proposed algorithms can be applied to solve more complex real life applications.

Bibliography

[1] Himmelblau, D. M. (1972). Applied nonlinear programming. McGraw-Hill Companies.

[2] Boyd, S., & Vandenberghe, L. (2004). Convex optimization. Cambridge University Press.

[3] Rao, S. S. (2009). Engineering optimization: theory and practice. John Wiley & Sons.

[4] Chandra, S., & Jayadeva, M. (2009). Numerical optimization with applications, Alpha Science International.

[5] Mohan, C., & Deep, K. (2009).Optimization techniques. New Age, New Delhi.

[6] Jha, P. C., & Hoda, M. N. (2011). Mathematical Modelling, Optimization and their applications. Narosa, New Delhi.

[7] Kapur, P. K., Pham, H., Gupta, A., & Jha, P. C. (2011). Software reliability assessment with OR applications. Springer, London.

[8] Bertsekas, D. P. (2014). Constrained optimization and Lagrange multiplier methods. Academic Press.

[9] Deb, K. (2001). Multi-objective optimization using evolutionary algorithms. John Wiley & Sons.

[10] Taha, A. H. (2017).Operations Research: Introduction. Pearson.

[11] Knowles, J., Corne, D., & Deb, K. (Eds.). (2007). Multiobjective problem solving from nature: from concepts to applications. Springer Science & Business Media.

[12] Coello, C. A. C., Lamont, G. B., & Van Veldhuizen, D. A. (2007). Evolutionary algorithms for solving multi-objective problems (Vol. 5). New York, Springer.

[13] Dahiya, K., Suneja, S. K., & Verma, V. (2007). Convex programming with single separable constraint and bounded variables. Computational Optimization and Applications, 36(1), 67-82.

[14] Dahiya, K., & Verma, V. (2007). Positive sensitivity analysis in linear programming with bounded variables. ASOR BULLETIN, 26(4), 2.

[15] Verma, V., & Puri, M. C. (1988). Constrained transportation problem with upper and lower bounds on row-availabilities and destination requirements. Cahiers Du Centre D'études De Recherche Opérationnelle, 30(2-3), 177-189.

[16] Omran, M. G. (2004). Particle swarm optimization methods for pattern recognition and image processing (Doctoral dissertation, University of Pretoria).

[17] Holland J.H. (1975). Adaptation in natural and artificial system, Ann Arbor, The University of Michigan Press.

[18] Storn, R., & Price, K. (1995). Differential evolution-A simple and efficient adaptive scheme for global optimization over continuous spaces [R]. Berkeley: ICSI.

[19] Ali, M. M., & Zhu, W. X. (2013). A penalty function-based differential evolution algorithm for constrained global optimization. Computational Optimization and Applications, 54(3), 707-739.

[20] Mezura-Montes, E., Velázquez-Reyes, J., & Coello, C. C. (2006, July). Modified differential evolution for constrained optimization. In Evolutionary Computation, 2006. CEC 2006. IEEE Congress on (pp. 25-32). IEEE.

[21] Mezura-Montes, E., Miranda-Varela, M. E., & Del Carmen Gómez-Ramón, R. (2010). Differential evolution in constrained numerical optimization: an empirical study. Information Sciences,180(22),4223-4262.

[22] Sabat, S. L., Kumar, K. S., & Udgata, S. K. (2009, December). Differential evolution and swarm intelligence techniques for analog circuit synthesis. In Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on (pp. 469-474).IEEE.

[23] Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization, Proceedings IEEE International Conference Neural Networks, 4, 1942-1948.

[24] Ali, M. M., & Kaelo, P. (2008). Improved particle swarm algorithms for global optimization. Applied Mathematics and Computation, 196(2), 578-593.

[25] Sabat, S. L., Ali, L., & Udgata, S. K. (2011). Integrated learning particle swarm optimizer for global optimization. Applied Soft Computing, 11(1), 574-584.

[26] Ali, L., Sabat, S. L., & Udgata, S. K. (2012). Particle swarm optimisation with stochastic ranking for constrained numerical and engineering benchmark problems. International Journal of Bio-Inspired Computation, 4(3), 155-166.

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

[28] Bansal, R., Sehgal, P., & Bedi, P. (2012). Securing fingerprint images using PSO- based wavelet domain watermarking. Information Security Journal: A Global Perspective, 21(2), 88-101.

[29] Chander, A., Chatterjee, A., & Siarry, P. (2011). A new social and momentum component adaptive PSO algorithm for image segmentation. Expert Systems with Applications, 38(5), 4998-5004.

[30] Krishnanand, K. N., & Ghose, D. (2006). Glowworm swarm based optimization algorithm for multimodal functions with collective robotics applications. Multiagent and Grid Systems, 2(3), 209-222.

[31] Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization (Vol. 200). Technical Report-tr06, Erciyes University, Engineering Faculty, Computer Engineering Department.

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

[33] Dorigo, M., & Di Caro, G. (1999). Ant colony optimization: a new meta-heuristic. In Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on (Vol. 2, pp. 1470-1477). IEEE.

[34] Passino, K. M. (2002). Biomimicry of bacterial foraging for distributed optimization and control. IEEE Control Systems, 22(3), 52-67.

[35] Mezura-Montes, E., & Hernández-Ocaña, B. (2009). Modified bacterial foraging optimization for engineering design. In Intelligent engineering systems through artificial neural networks. ASME Press.

[36] Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. Advances in Engineering Software, 69, 46-61.

[37] Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2009). GSA: A gravitational search algorithm. Information Sciences, 179(13), 2232-2248.

[38] Formato, R. A. (2007). Central force optimization: a new metaheuristic with applications in applied electromagnetics. Progress In Electromagnetics Research, 77, 425-491.

[39] Geem Z.W., Kim J.H., Loganathan G.V. (2001).A New Heuristic Optimization Algorithm: Harmony Search. SIMULATION 76: 60.

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

[41] Yadav, A., Yadav, N., & Kim, J. H. (2017, February). A Comparative Study of Exploration Ability of Harmony Search Algorithms. In International Conference on Harmony Search Algorithm (pp. 22-31). Springer, Singapore.

[42] Simon, D. (2008). Biogeography-based optimization. IEEE Transactions on Evolutionary Computation, 12(6), 702-713.

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

[44] Shah-Hosseini, H. (2009). Optimization with the nature-inspired intelligent water drops algorithm. In Evolutionary Computation. InTech.

[45] Tan, Y., & Zhu, Y. (2010, June). Fireworks algorithm for optimization. In International Conference in Swarm Intelligence (pp. 355-364). Springer, Berlin, Heidelberg.

[46] Rao, R. V., Savsani, V. J., & Vakharia, D. P. (2011). Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. Computer-Aided Design, 43(3), 303-315.

[47] Mehrabian, A. R., & Lucas, C. (2006). A novel numerical optimization algorithm inspired from weed colonization. Ecological informatics, 1(4), 355-366.

[48] Jaddi, N. S., Alvankarian, J., & Abdullah, S. (2017). Kidney-inspired algorithm for optimization problems. Communications in Nonlinear Science and Numerical Simulation, 42, 358-369.

[49] Mirjalili, S. (2015). Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. Knowledge-Based Systems, 89, 228-249.

[50] Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. Advances in Engineering Software, 95, 51-67.

[51] Mirjalili, S. (2016). SCA: a sine cosine algorithm for solving optimization problems. Knowledge-Based Systems, 96, 120-133.

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

[53] Yang, X. S. (2014). Nature-inspired optimization algorithms. Elsevier.

[54] Kar, A. K. (2016). Bio inspired computing–A review of algorithms and scope of applications. Expert Systems with Applications, 59, 20-32.

[55] Yang, X. S., Chien, S. F., & Ting, T. O. (2015). Bio-inspired computation in telecommunications. Morgan Kaufmann.

[56] Mirjalili, S. (2015). The ant lion optimizer. Advances in Engineering Software, 83, 80-98.

[57] Wolpert, D. H., & Macready, W. G. (1995). No free lunch theorems for search (Vol. 10). Technical Report SFI-TR-95-02-010, Santa Fe Institute.

[58] Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation, 1(1), 67-82.

[59] Yao, P., & Wang, H. (2016). Dynamic Adaptive Antlion Optimizer applied to route planning for unmanned aerial vehicle. Soft Computing, 1-14.

[60] Saha, S., & Mukherjee, V. (2017). A novel quasi-oppositional chaotic antlion optimizer for global optimization. Applied Intelligence, 1-33.

[61] Zawbaa, H. M., Emary, E., & Grosan, C. (2016). Feature selection via chaotic antlion optimization. PloS one, 11(3), e0150652.

[62] Emary, E., Zawbaa, H. M., & Hassanien, A. E. (2016). Binary antlion approaches for feature selection. Neurocomputing, 213, 54-65.

[63] Mirjalili, S., & Hashim, S. Z. M. (2012). BMOA: binary magnetic optimization algorithm. International Journal of Machine Learning and Computing, 2(3), 204.

[64] Mafarja, M., Eleyan, D., Abdullah, S., & Mirjalili, S. (2017, July). S-shaped vs. V-shaped transfer functions for ant lion optimization algorithm in feature selection problem. In Proceedings of the International Conference on Future Networks and Distributed Systems (p. 14). ACM.

[65] Mirjalili, S., Jangir, P., & Saremi, S. (2017). Multi-objective ant lion optimizer: a multi-objective optimization algorithm for solving engineering problems. Applied Intelligence, 46(1), 79-95.

[66] Pradhan, R., Majhi, S. K., Pradhan, J. K., & Pati, B. B. (2017). Performance Evaluation of PID Controller for an Automobile Cruise Control System using Ant Lion Optimizer. Engineering Journal (Eng. J.), 21(5), 347-361.

[67] Satheeshkumar, R., & Shivakumar, R. (2016). Antlion optimization approach for load frequency control of multi-area interconnected power systems. Circuits and Systems, 7(09), 2357.

[68] Padhy, B. P., Srivastava, S. C., & Verma, N. K. (2012). Robust wide-area TS fuzzy output feedback controller for enhancement of stability in multimachine power system. IEEE Systems Journal, 6(3), 426-435.

[69] Gupta, E., & Saxena, A. Performance evaluation of antlion optimizer based regulator in automatic generation control of interconnected power system. Journal of Engineering, 2016.

[70] Trivedi, I. N., Jangir, P., & Parmar, S. A. Optimal power flow with enhancement of voltage stability and reduction of power loss using ant-lion optimizer. Cogent Engineering, 2016; 3(1), 1208942.

[71] Mouassa, S., Bouktir, T., & Salhi, A. (2017). Antlion optimizer for solving optimal reactive power dispatch problem in power systems. Engineering Science and Technology, an International Journal, 20(3), 885-895.

[72] Tung, N. S., & Chakravorty, S. (2016). Ant Lion Optimizer based Approach for Optimal Scheduling of Thermal Units for Small Scale Electrical Economic Power Dispatch Problem. Optimization (PSO), 9(7), 211-224.

[73] Rajan, A., Jeevan, K., & Malakar, T. (2017). Weighted elitism based Ant Lion Optimizer to solve optimum VAr planning problem. Applied Soft Computing, 55, 352-370.

[74] Radha, J., Subramanian, S., Ganesan, S., & Abirami, M. (2017). Emission/fuel energy restricted dynamic optimal power flow using fuzzy-ant lion optimizer. International Journal of Energy Sector Management, 11(2), 208-256.

[75] Ali, E. S., Elazim, S. A., & Abdelaziz, A. Y. (2016). Antlion optimization algorithm for renewable distributed generations. Energy, 116, 445-458

[76] Ali, E. S., Elazim, S. A., & Abdelaziz, A. Y. (2017). Ant Lion Optimization algorithm for optimal location and sizing of renewable distributed generations. Renewable Energy, 101, 1311-1324.

[77] Edenhofer O, Pichs-Madruga R, Sokona Y, Seyboth K, Matschoss P, Kadner S, et al. IPCC special report on renewable energy sources and climate change mitigation. IPCC. In: Prepared by Working Group III of the Intergovernmental Panel on Climate Change. United Kingdom and New York, NY, USA: Cambridge University Press, Cambridge; 2011. p. 1075 [Chapter 7 & 9].

[78] Sam'on, I. N., Yasin, Z. M., & Zakaria, Z. (2017). Ant Lion optimizer for solving unit commitment problem in smart grid system. Indonesian Journal of Electrical Engineering and Computer Science, 8(1), 129-136.

[79] Tripathi, R. K., Singh, Y. N., & Verma, N. K. (2013). Clustering algorithm for non-uniformly distributed nodes in wireless sensor network. Electronics Letters, 49(4), 299-300.

[80] Yogarajan, G., & Revathi, T. (2018). Improved cluster based data gathering using ant Lion optimization in wireless sensor networks. Wireless Personal Communications, 98(3), 2711-2731.

[81] Mostafa, A., Houseni, M., Allam, N., Hassanien, A. E., Hefny, H., & Tsai, P. W. (2016, November). Antlion optimization based segmentation for MRI liver images. In International Conference on Genetic and Evolutionary Computing (pp. 265-272). Springer, Cham.

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

[83] Bedi, P., Bansal, R., & Sehgal, P. (2011, July). Using PSO in image hiding scheme based on LSB substitution. In International Conference on Advances in Computing and Communications (pp. 259-268). Springer, Berlin, Heidelberg.

[84] Bedi, P., & Sharma, C. (2016). Community detection in social networks. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 6(3), 115-135.

[85] Babers, R., Ghali, N. I., Hassanien, A. E., & Madbouly, N. M. (2015, December). Optimal community detection approach based on Ant Lion Optimization. In Computer Engineering Conference (ICENCO), 2015 11th International (pp. 284-289). IEEE.

[86] Pizzuti, C. (2008, July). Community detection in social networks with genetic algorithms. In Proceedings of the 10th annual conference on Genetic and evolutionary computation(pp. 1137-1138). ACM.

[87] Pizzuti, C. (2008, September). Ga-net: A genetic algorithm for community detection in social networks. In International Conference on Parallel Problem Solving from Nature (pp. 1081-1090). Springer, Berlin, Heidelberg.

[88] Danon, L., Diaz-Guilera, A., Duch, J., & Arenas, A. (2005). Comparing community structure identification. Journal of Statistical Mechanics: Theory and Experiment, 2005(09), P09008.

[89] Saxena, P., & Kothari, A. (2016). Ant Lion optimization algorithm to control side lobe level and null depths in linear antenna arrays. AEU-International Journal of Electronics and Communications, 70(9), 1339-1349.

[90] Subhashini, K. R., & Satapathy, J. K. (2017). Development of an enhanced ant Lion optimization algorithm and its application in antenna array synthesis. Applied Soft Computing, 59, 153-173.

[91] Yamany, W., Tharwat, A., Hassanin, M. F., Gaber, T., Hassanien, A. E., & Kim, T. H. (2015, September). A new multi-layer perceptrons trainer based on ant lion optimization algorithm. In Information Science and Industrial Applications (ISI), 2015 Fourth International Conference on (pp. 40-45). IEEE.

[92] Verma, N. K., Meena, S., Bajpai, S., Singh, A., Nagrare, A., & Cui, Y. (2010, December). A comparison of biclustering algorithms. In Systems in Medicine and Biology (ICSMB), 2010 International Conference on (pp. 90-97). IEEE.

[93] Hamouda, E., El-Metwally, S., & Tarek, M. (2018). Ant Lion Optimization algorithm for kidney exchanges. PloS one, 13(5), e0196707.

[94] Elaziz, M. A., Moemen, Y. S., Hassanien, A. E., & Xiong, S. (2018). Quantitative structure-activity relationship model for HCVNS5B inhibitors based on an Antlion Optimizer-Adaptive Neuro-Fuzzy Inference System. Scientific reports, 8(1), 1506.

[95] Gupta, S., Kumar, V., Rana, K. P. S., Mishra, P., & Kumar, J. (2016, February). Development of Ant Lion Optimizer toolkit in LabVIEW™. In Innovation and Challenges in Cyber Security (ICICCS-INBUSH), 2016 International Conference on (pp. 251-256). IEEE.

[96] Gupta, P., Rana, K. P. S., Kumar, V., Mishra, P., Kumar, J., & Nair, S. S. (2015, February). Development of a Grey Wolf Optimizer Toolkit in LabVIEW™. In Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE), 2015 International Conference on (pp. 107-113). IEEE.

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

[98] Tizhoosh, H. R. (2005, November). Opposition-based learning: a new scheme for machine intelligence. In Computational intelligence for modelling, control and automation, 2005 and international conference on intelligent agents, web technologies and internet commerce, International Conference on (Vol. 1, pp. 695-701). IEEE.

[99] Rahnamayan, S., Tizhoosh, H. R., & Salama, M. M. (2008). Opposition-based differential evolution. IEEE Transactions on Evolutionary Computation, 12(1), 64-79.

[100] Wang, H., Wu, Z., Wang, J., Dong, X., Yu, S., & Chen, C. (2009, August). A new population initialization method based on space transformation search. In Natural Computation, 2009. ICNC'09. Fifth International Conference on (Vol. 5, pp. 332-336). IEEE.

[101] Ahandani, M. A., & Alavi-Rad, H. (2012). Opposition-based learning in the shuffled differential evolution algorithm. Soft Computing, 16(8), 1303-1337.

[102] Rahnamayan, S., Tizhoosh, H. R., & Salama, M. M. (2008). Opposition versus randomness in soft computing techniques. Applied Soft Computing, 8(2), 906-918.

[103] Elaziz, M. A., Oliva, D., & Xiong, S. (2017). An improved Opposition-Based Sine Cosine Algorithm for global optimization. Expert Systems with Applications, 90, 484-500.

[104] Sharma, H., Bansal, J. C., & Arya, K. V. (2013). Opposition based lévy flight artificial bee colony. Memetic Computing, 5(3), 213-227.

[105] Rahnamayan, S., Tizhoosh, H. R., & Salama, M. M. (2008). Opposition-based differential evolution. IEEE Transactions on Evolutionary computation, 12(1), 64-79.

[106] Ahandani, M. A., & Alavi-Rad, H. (2015). Opposition-based learning in shuffled frog leaping: An application for parameter identification. Information Sciences, 291, 19-42.

[107] Wang, G. G., Deb, S., Gandomi, A. H., & Alavi, A. H. (2016). Opposition-based krill herd algorithm with Cauchy mutation and position clamping. Neurocomputing, 177, 147-157.

[108] Yao, X., Liu, Y., & Lin, G. (1999). Evolutionary programming made faster. IEEE Transactions on Evolutionary Computation, 3(2), 82-102.

[109] Liang, J. J., Qu, B. Y., & Suganthan, P. N. (2013). 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.

[110] Van Den Bergh, F., & Engelbrecht, A. P. (2006). A study of particle swarm optimization particle trajectories. Information sciences, 176(8), 937-971.

[111] Saremi, S., Mirjalili, S., & Lewis, A. (2017). Grasshopper optimisation algorithm: theory and application. Advances in Engineering Software, 105, 30-47.

[112] Chen, K., Zhou, F., Yin, L., Wang, S., Wang, Y., & Wan, F. (2018). A hybrid particle swarm optimizer with sine cosine acceleration coefficients. Information Sciences, 422, 218-241.

[113] Viswanathan, G. M., Afanasyev, V., Buldyrev, S. V., & Murphy, E. J. (1996). Lévy flight search patterns of wandering albatrosses. Nature, 381(6581), 413.

[114] Humphries, N. E., Queiroz, N., Dyer, J. R., Pade, N. G., Musyl, M. K., Schaefer, K. M., & Hays, G. C. (2010). Environmental context explains Lévy and Brownian movement patterns of marine predators. Nature, 465(7301), 1066.

[115] Sims, D. W., Southall, E. J., Humphries, N. E., Hays, G. C., Bradshaw, C. J., Pitchford, J. W.,& Morritt, D. (2008). Scaling laws of marine predator search behaviour. Nature, 451(7182), 1098.

[116] Yang, X. S., & Deb, S. (2009, December). Cuckoo search via Lévy flights. In Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on (pp. 210-214). IEEE.

[117] Kalantzis, G., Shang, C., Lei, Y., & Leventouri, T. (2016). Investigations of a GPU-based lévy-firefly algorithm for constrained optimization of radiation therapy treatment planning. Swarm and Evolutionary Computation, 26, 191-201.

[118] Heidari, A. A., & Pahlavani, P. (2017). An efficient modified grey wolf optimizer with Lévy flight for optimization tasks. Applied Soft Computing, 60, 115-134.

[119] Bhateja, A. K., Bhateja, A., Chaudhury, S., & Saxena, P. K. (2015). Cryptanalysis of vigenere cipher using cuckoo search. Applied Soft Computing, 26, 315-324.

[120] Haklı, H., & Uğuz, H. (2014). A novel particle swarm optimization algorithm with Levy flight. Applied Soft Computing, 23, 333-345.

[121] Jensi, R., & Jiji, G. W. (2016). An enhanced particle swarm optimization with levy flight for global optimization. Applied Soft Computing, 43, 248-261.

[122] Atay, Y., Koc, I., Babaoglu, I., & Kodaz, H. (2017). Community detection from biological and social networks: A comparative analysis of metaheuristic algorithms. Applied Soft Computing, 50, 194-211.

[123] Yang, X.S. (2010).Appendix a: test problems in optimization. Engineering Optimization, pp. 261–266.

[124] Yang, X.S. (2010) Nature-inspired metaheuristic algorithms, 2nd edn. Luniver Press, Beckington.

[125] Shlesinger, M. F., Zaslavsky, G. M., & Frisch, U. (1995). Lévy flights and related topics in physics. Lecture Notes in Physics, 450, 52.

[126] Digalakis, J. G., & Margaritis, K. G. (2001). On benchmarking functions for genetic algorithms. International Journal of Computer Mathematics, 77(4), 481-506.

[127] Wang, H., Li, H., Liu, Y., Li, C., & Zeng, S. (2007, September). Opposition-based particle swarm algorithm with Cauchy mutation. In Evolutionary Computation, 2007. CEC 2007. IEEE Congress on (pp. 4750-4756). IEEE.

[128] Wu, Q. (2011). Hybrid forecasting model based on support vector machine and particle swarm optimization with adaptive and Cauchy mutation. Expert systems with applications, 38(8), 9070-9075.

[129] Qin, H., Zhou, J., Lu, Y., Wang, Y., & Zhang, Y. (2010). Multi-objective differential evolution with adaptive Cauchy mutation for short-term multi-objective optimal hydro-thermal scheduling. Energy Conversion and Management, 51(4), 788-794.

[130] Ali, M., & Pant, M. (2011). Improving the performance of differential evolution algorithm using Cauchy mutation. Soft Computing, 15(5), 991-1007.

[131] Norman, L., Kotz, S., & Balakrishnan, N. (1994). Continuous Univariate Distributions.

[132] Feller, William (1971). An Introduction to Probability Theory and Its Applications, Volume II(2 ed.). New York: John Wiley & Sons Inc. p. 704. ISBN 978-0-471-25709-7.

[133] Sikander, A. A., & Prasad, B. R. (2015). A novel order reduction method using cuckoo search algorithm. IETE Journal of Research, 61(2), 83-90.

[134] Sikander, A., & Prasad, R. (2015). Linear time-invariant system reduction using a mixed methods approach. Applied Mathematical Modelling, 39(16), 4848-4858.

[135] Biradar, S., Hote, Y. V., & Saxena, S. Reduced-order modeling of linear time invariant systems using big bang big crunch optimization and time moment matching method. Applied Mathematical Modelling, 2006; 40(15), 7225-7244.

[136] Davison, E. (1966). A method for simplifying linear dynamic systems. IEEE Transactions on automatic control, 11(1), 93-101.

[137] Chidambara, M., & Davison, E. (1967). Further comments on "A method for simplifying linear dynamic systems". IEEE Transactions on Automatic Control, 12(6), 799-800.

[138] Ahmed, N., & Karni, S. (1967). On obtaining transfer functions from gain-function derivatives. IEEE Transactions on Automatic Control, 12(2), 229-229.

[139] Chidambara, M. R. (1969). Two simple techniques for the simplification of large dynamic systems. In Joint Automatic Control Conference (No. 7, pp. 669-674).

[140] Ackermann, J. E., & Bucy, R. S. (1970). Canonical minimal realization of a matrix of impulse response sequences. UNIVERSITY OF SOUTHERN CALIFORNIA LOS ANGELES DEPT OF AEROSPACE ENGINEERING.

[141] Ho, B. L., & Kálmán, R. E. (1966). Effective construction of linear state-variable models from input/output functions. at-Automatisierungstechnik, 14(1-12), 545-548.

[142] Rissanen, J., & Kailath, T. (1972). Partial realization of random systems. Automatica, 8(4), 389-396.

[143] Silverman, L. (1971). Realization of linear dynamical systems. IEEE Transactions on Automatic Control, 16(6), 554-567.

[144] Aoki, M. Control of large-scale dynamic systems by aggregation. IEEE Transactions on Automatic Control, 1968; 13(3), 246-253.

[145] Shamash, Y. Stable reduced-order models using Padé-type approximations. IEEE Transactions on Automatic Control, 1974; 19(5), 615-616.

[146] Hutton, M., & Friedland, B. Routh approximations for reducing order of linear, time-invariant systems. IEEE Transactions on Automatic Control, 1975; 20(3), 329-337.

[147] Sinha, N. K., & Kuszta, B. Modelling and identification of dynamic systems. Springer, 1983.

[148] Krishnamurthy, V., & Seshadri, V. Model reduction using the Routh stability criterion. IEEE Transactions on Automatic Control, 1978; 23(4), 729-731.

[149] Sharma, H., Bansal, J. C., & Arya, K. V. Fitness based differential evolution. Memetic Computing, 4(4), 2012; 303-316.

[150] Wilson, D. A. (1970, June). Optimum solution of model-reduction problem. In Proceedings of the Institution of Electrical Engineers (Vol. 117, No. 6, pp. 1161-1165). IET.

[151] Wilson, D. A. (1974). Model reduction for multivariable systems. International Journal of Control, 20(1), 57-64.

[152] Parmar, G., Pandey, M. K., & Kumar, V. (2009, January). System order reduction using GA for unit impulse input and a comparative study using ISE & IRE. In Proceedings of the International Conference on Advances in Computing, Communication and Control (pp. 3-7). ACM.

[153] Panda, S., Yadav, J. S., Patidar, N. P., & Ardil, C. (2009). Evolutionary techniques for model order reduction of large scale linear systems. International Journal of Applied Science, Engineering and Technology, 5(1), 22-28.

[154] Desai, S. R., & Prasad, R. A novel order diminution of LTI systems using Big Bang Big Crunch optimization and Routh Approximation. Applied Mathematical Modelling, 2013; 37(16), 8016-8028.

[155] Smamash, Y. Truncation method of reduction: a viable alternative. Electronics Letters, 1981; 17(2), 97-99.

[156] Pal, J. (1986). An algorithmic method for the simplification of linear dynamic scalar systems. International Journal of Control, 43(1), 257-269.

[157] Singh N (2007) Reduced order modelling and controller design. PhD thesis. Indian Institute of Technology Roorkee, India.

[158] Tesch, K., Atherton, M. A., Karayiannis, T. G., Collins, M. W., & Edwards, P. (2009). Determining heat transfer coefficients using evolutionary algorithms. Engineering Optimization, 41(9), 855-870.

[159] Safikhani, H., Abbassi, A., Khalkhali, A., & Kalteh, M. (2015). Modeling and optimization of nanofluid flow in flat tubes using a combination of CFD and response surface methodology. Heat Transfer—Asian Research, 44(4), 377-395.

[160] Murshed SMS, Leong KC, Yang CA. Combined model for the effective thermal conductivity of nanofluids. Appl Therm Eng 2009; 29:2477–2483.

[161] Lotfi R, Saboohi Y, Rashidi A. Numerical study of forced convective heat transfer of Nanofluids: Comparison of different approaches. Int Commun Heat Mass 2010;37:74–78.

[162] Shariat M, et al. Numerical study of two phase laminar mixed convection nanofluid in elliptic ducts. Appl Therm Eng 2011;31:2348–2359.

[163] Razi P, Akhavan-Behabadi M, Saeedinia M. Pressure drop and thermal characteristics of CuO– base oil nanofluid laminar flow in flattened tubes under constant heat flux. Int Commun Heat Mass 2011;38:964–971.

[164] Vajjha RS, Das DK, Namburu PK. Numerical study of fluid dynamic and heat transfer performance of Al2O3 and CuO nanofluids in the flat tubes of a radiator. Int J Heat Fluid Fl 2010;31:613–621.

[165] Safikhani, H., Abbassi, A., Khalkhali, A., & Kalteh, M. (2014). Multi-objective optimization of nanofluid flow in flat tubes using CFD, Artificial Neural Networks and genetic algorithms. Advanced Powder Technology, 25(5), 1608-1617.

[166] Zhang, L., Zhang, N., Zhao, F., & Chen, Y. (2004). A genetic-algorithm-based experimental technique for determining heat transfer coefficient of exterior wall surface. Applied Thermal Engineering, 24(2-3), 339-349.

[167] Fonseca, C. M., & Fleming, P. J. (1995). An overview of evolutionary algorithms in Multi-objective optimization. Evolutionary Computation, 3(1), 1-16.

[168] Holland, J. H., & Goldberg, D. (1989). Genetic algorithms in search, optimization and machine learning. Massachusetts: Addison-Wesley.

[169] Laumanns, M., Thiele, L., Deb, K., & Zitzler, E. (2002). Combining convergence and diversity in evolutionary Multi-objective optimization. Evolutionary Computation, 10(3), 263-282.

[170] Fogel, D. B., & Michalewicz, Z. (2000). Evolutionary Computation 2: Advanced Algorithms and Operators.

[171] Coello, C. C. (2006). Evolutionary multi-objective optimization: a historical view of the field. IEEE Computational Intelligence Magazine, 1(1), 28-36.

[172] Srinivas, N., & Deb, K. (1994). Muiltiobjective optimization using nondominated sorting in genetic algorithms. Evolutionary Computation, 2(3), 221-248.

[173] Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. A. M. T. (2002). A fast and elitist Multi-objective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 6(2), 182-197.

[174] Mirjalili, S., Jangir, P., & Saremi, S. (2017). Multi-objective ant lion optimizer: a multi-objective optimization algorithm for solving engineering problems. Applied Intelligence, 46(1), 79-95.

[175] Atabani, A. E., Silitonga, A. S., Ong, H. C., Mahlia, T. M. I., Masjuki, H. H., Badruddin, I. A., & Fayaz, H. (2013). Non-edible vegetable oils: a critical evaluation of oil extraction, fatty acid compositions, biodiesel production, characteristics, engine performance and emissions production. Renewable and Sustainable Energy Reviews, 18, 211-245.

[176] Demibras, A. (2009). Progress and recent trends in biodiesel fuels. Energy Conversion and Management, 50(1), 14-34.

[177] Silva, G. F., Camargo, F. L., & Ferreira, A. L. (2011). Application of response surface methodology for optimization of biodiesel production by transesterification of soybean oil with ethanol. Fuel Processing Technology, 92(3), 407-413.

[178] Niu, S. L., Huo, M. J., Lu, C. M., Liu, M. Q., & Li, H. (2014). An investigation on the catalytic capacity of dolomite in transesterification and the calculation of kinetic parameters. Bioresource Technology, 158, 74-80.

[179] Santos, R. C., Vieira, R. B., & Valentini, A. (2014). Optimization Study in Biodiesel Production via Response Surface Methodology Using Dolomite as a Heterogeneous Catalyst. Journal of Catalysts, Volume 2014, Article ID 213607, 11 pages, 2014.

[180] Roschat, W., Kacha, M., Yoosuk, B., Sudyoadsuk, T., & Promarak, V. (2012). Biodiesel production based on heterogeneous process catalyzed by solid waste coral fragment. Fuel, 98, 194-202.

[181] Mittelbach, M. (2004). Biodiesel. The compherensive handbook (No. L-0577). Martin Mittelbach.

[182] Kovarik, B. (1998). Henry Ford, Charles F. Kettering and the fuel of the future. Automotive History Review, 32, 7-27.

[183] Knothe, G. (2001). Historical perspectives on vegetable oil-based diesel fuels (No. D-1568).

[184] Chavanne G. Procédé de transformation d'huiles végétales en vue de leur utilisation comme carburants (Procedure for the transformation of vegetable oils for their uses as fuels). Belgian Patent 422,877, 31 August 1937. Chem Abstr 1938;32:43132

[185] Chavanne G. Sur un mode d'utilization possible de l'huile de palme à la fabrication d'un carburant lourd (A method of possible utilization of palm oil for the manufacture of a heavy fuel). Bulletin des Sociétés Chimiques 1943;10(52):58. Chem Abstr 1944; 38:21839.

[186] Knothe G, Van Gerpen J, Krahl J, editors. The biodiesel handbook. Urbana, IL: AOCS Press; 2010.

[187] Bruwer, J. J., Hugo, F. J. C., & HAWKINS, C. (1980). Sunflower seed oil as an extender for diesel fuel in agricultural tractors.

[188] Huber, G. W., & Corma, A. (2007). Synergies between Bio-and Oil Refineries for the Production of Fuels from Biomass. Angewandte Chemie International Edition, 46(38), 7184-7201.

[189] Knothe, G. (2010). Biodiesel and renewable diesel: a comparison. Progress in Energy and Combustion Science, 36(3), 364-373.

[190] Lee, A. F., Bennett, J. A., Manayil, J. C., & Wilson, K. (2014). Heterogeneous catalysis for sustainable biodiesel production via esterification and transesterification. Chemical Society Reviews, 43(22), 7887-7916.

[191] Lee, A. F., & Wilson, K. (2015). Recent developments in heterogeneous catalysis for the sustainable production of biodiesel. Catalysis Today, 242, 3-18.

[192] Fjerbaek, L., Christensen, K. V., & Norddahl, B. (2009). A review of the current state of biodiesel production using enzymatic transesterification. Biotechnology and Bioengineering, 102(5), 1298-1315.

[193] Christopher, L. P., Kumar, H., & Zambare, V. P. (2014). Enzymatic biodiesel: challenges and opportunities. Applied Energy, 119, 497-520.

[194] Guldhe, A., Singh, B., Mutanda, T., Permaul, K., & Bux, F. (2015). Advances in synthesis of biodiesel via enzyme catalysis: novel and sustainable approaches. Renewable and Sustainable Energy Reviews, 41, 1447-1464.

[195] Aguieiras, E. C., Cavalcanti-Oliveira, E. D., & Freire, D. M. (2015). Current status and new developments of biodiesel production using fungal lipases. Fuel, 159, 52-67.

[196] Muhammad, N., Elsheikh, Y. A., Mutalib, M. I. A., Bazmi, A. A., Khan, R. A., Khan, H., & Man, Z. (2015). An overview of the role of ionic liquids in biodiesel reactions. Journal of Industrial and Engineering Chemistry, 21, 1-10.

[197] Fayyazi, E., Ghobadian, B., Najafi, G., & Hosseinzadeh, B. (2014). Genetic algorithm approach to optimize biodiesel production by ultrasonic system. Chemical Product and Process Modeling, 9(1), 59-70.

Appendix I

Benchmark Test Problems

This Appendix presents a set of 31 unconstrained continuous benchmark problems having different difficulty level of characteristics. These problems include 23 state-of-the-art problems including (i) 7 scalable unimodal problems ($F_1 - F_7$) (ii) 6 scalable multimodal problems ($F_8 - F_{13}$) (iii) 10 fixed dimensional multimodal problems ($F_{14} - F_{23}$) and a set of 8 complex composition functions taken from IEEE CEC 2014 benchmark problem set. Scalable problems considered in this thesis are 10 and 30 dimensions. The list of these problems in tabular form are shown as under:

Function	***Search Range***	***Optimum Value***
Unimodal Functions	D=10,30	
$F_1(X) = \sum_{i=1}^{D} x_i^2$	[-100,100]	$F_1(X_{min}) = 0$
$F_2(X)=\sum_{i=1}^{D} \lvert x \rvert + \prod_{i=1}^{D} x_i$	[-10,10]	$F_2(X_{min})=0$
$F_3(X)=\sum_{i=1}^{D}(\sum_{j=1}^{i} x_j)^2$	[-100,100]	$F_3(X_{min})=0$
$F_4(X) = \max \lvert x_i \rvert, 1 \le i \le D$	[-100,100]	$F_4(X_{min})=0$
$F_5(X)=\sum_{i}^{D-1}[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	[-30,30]	$F_5(X_{min})=0$
$F_6(X)=\sum_{i}^{D}([x_i + 0.5])^2$	[-100,100]	$F_6(X_{min})=0$
$F_7(X) = \left(\sum_{i}^{D} i.x_i^4\right) + rand[0,1]$	[-1.28,1.28]	$F_7(X_{min})=0$
Multimodal Functions	D=10,30	
$F_8(X)=\sum_{i}^{D} -x_i.\sin(\sqrt{\lvert x_i \rvert})$	[-500,500]	$F_8(X_{min})$=-418.9829*D
$F_9(X) = \sum_{i}^{D}[x_i^2 - 10\cos(2\pi x_i) + 10]$	[-5.12,5.12]	$F_9(X_{min}) = 0$
$F_{10}(X)=-20\exp\left(-^{0.2}\sqrt{\frac{1}{D}\sum_{i}^{D} x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i}^{D}\cos 2\pi x_i\right) + 20 + e$	[-32,32]	$F_{10}(X_{min})=0$

$F_{11}(X) = \frac{1}{4000}\sum_{i}^{D} x_i^2 - \prod_{1}^{D} \cos(\frac{x_i}{\sqrt{i}}) + 1$	[-600,600]	$F_{11}(X_{min}) = 0$
$F_{12}(X) = \frac{\pi}{D}\{10\, sin^2(\pi y_1) + \sum_{i=1}^{D}(x_i - 1)^2.[1 + sin^2(xy_{i+1})] + (y_D - 1)^2\} + \sum_{i}^{D} u(x_{i,} 10,100,4)$	[-50,50]	$F_{12}(X_{min})$=0
$F_{13}(X)$=0.1$\{sin^2(3\pi x_1)$+$\sum_{i=1}^{D}(x_i - 1)^2.[1 + sin^2(3\pi x_{i+1})] + (x_D - 1)^2\} + \sum_{i}^{D} u(x_{i,}5,100,4)$	[-50,50]	$F_{13}(X_{min}) = 0$
Fixed Dimension Functions		
$F_{14}(X) = \left(\frac{1}{500} + \sum_{j=1}^{25}(j + 1 + \sum_{i=0}^{1}(x_i - a_{ij})^6)^{-1}\right.$	[-65.54,65.54] D=2	$F_{14}(X_{min})$=0.998
$F_{15}(X) = \sum_{i=0}^{10}(a_i - \frac{x_0(b_i^2+b_i x_1)}{b_i^2+b_i x_2+x_3})^2$	[-5,5] D=4	$F_{15}(X_{min})$=0.0003075
$F_{16}(X) = 4x_0^2 - 2.1x_0^4 + \frac{1}{3}x_0^6 + x_0 x_1$	[-5,5] D=2	$F_{16}(X_{min})$=-1.0316
$F_{17}(X) = (x_1 - \frac{5.1}{4\pi^2}x_0^2 + \frac{5}{\pi}x_0 - 6)^2$+10(1-$\frac{1}{8\pi}$)cos$(x_0)$+10	[-5,5] D=2	$F_{17}(X_{min})$=0.398
$F_{18}(X) = \{1 + (x_0 + x_1 + 1)^2$(19-14$x_0 + 3x_0^2 - 14x_1 - 6x_0x_1 + 3x_1^2)\}\{$30+(2$x_0 - 3x_1)^2$(18-32$x_0 + 12x_0^2 + 48x_1 - 36x_0x_1 + 27x_1^2)\}$	[-2,2] D=2	$F_{18}(X_{min})$=3
$f_{19}(X) = -\sum_{i=1}^{5}((X - a_i)^{T}(X - a_i)+c_i)^{-1}$	[0,10] D=4	$F_{19}(X_{min})$=-10.1532
$F_{20}(X) = -\sum_{i=1}^{7}((X - a_i)^{T}(X - a_i)+c_i)^{-1}$	[0,10] D=4	$F_{20}(X_{min}) = -10.4029$
$F_{21}(X) = -\sum_{i=1}^{10}((X - a_i)^{T}(X - a_i)+c_i)^{-1}$	[0,10] D=4	$F_{21}(X_{min}) = -10.5364$
$F_{22}(X) = -\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2)$	[1,3] D=3	$F_{22}(X_{min}) = -3.86$
$F_{23}(X) = -\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2)$	[0,1] D=6	$F_{23}(X_{min}) = -3.32$
Composition Functions CEC 2014 D=10		
$F_{24}(X)$=Composition Function 1	[-100,100]	$F_{24}(X_{min}) = 2300$
$F_{25}(X)$= Composition Function 2	[-100,100]	$F_{25}(X_{min}) = 2400$
$F_{26}(X)$= Composition Function 3	[-100,100]	$F_{26}(X_{min}) = 2500$
$F_{27}(X)$= Composition Function 4	[-100,100]	$F_{27}(X_{min}) = 2600$
$F_{28}(X)$= Composition Function 5	[-100,100]	$F_{28}(X_{min})$ =2700
$F_{29}(X)$= Composition Function 6	[-100,100]	$F_{29}(X_{min}) = 2800$
$f_{30}(X)$= Composition Function 7	[-100,100]	$F_{30}(X_{min})$ =2900
$F_{31}(X)$= Composition Function 8	[-100,100]	$F_{31}(X_{min}) = 3000$