SELF ADAPTIVE GREY WOLF OPTIMIZATION ALGORITHM FOR LARGE SCALE OPTIMIZATION PROBLEMS

A THESIS

Submitted by

R. RAJAKUMAR

in partial fulfilment for the award of the degree

of

DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTER SCIENCE

SCHOOL OF ENGINEERING AND TECHNOLOGY

PONDICHERRY UNIVERSITY

PUDUCHERRY 605 014

INDIA

MARCH 2019

SELF ADAPTIVE GREY WOLF OPTIMIZATION ALGORITHM FOR LARGE SCALE OPTIMIZATION PROBLEMS

A THESIS

Submitted by

R. RAJAKUMAR

in partial fulfilment for the award of the degree

of

DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTER SCIENCE

SCHOOL OF ENGINEERING AND TECHNOLOGY

PONDICHERRY UNIVERSITY

PUDUCHERRY 605 014

INDIA

MARCH 2019

PONDICHERRY UNIVERSITY
(A Central University)
SCHOOL OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE

BONAFIDE CERTIFICATE

This is to certify that this thesis entitled **"Self Adaptive Grey Wolf Optimization Algorithm for Large Scale Optimization Problems"** is submitted by **Mr. R. Rajakumar,** to the Department of Computer Science, School of Engineering and Technology, Pondicherry University, Puducherry, India for the award of the degree of Doctor of Philosophy in Computer Science and Engineering is a record of bonafide research work carried out by him under my guidance and supervision.

This work is original and has not been submitted elsewhere, in part or full to this or any other University or Institute for the award of any degree.

Place : Puducherry
Date : 01-03-2019

SUPERVISOR
Dr. T. VENGATTARAMAN
Assistant Professor
Department of Computer Science,
School of Engineering and Technology
Pondicherry University
Puducherry - 605 014.

DECLARATION

I hereby declare that this thesis entitled **"Self Adaptive Grey Wolf Optimization Algorithm for Large Scale Optimization Problems"** submitted to the Department of Computer Science, School of Engineering and Technology, Pondicherry University, Puducherry, India for the award of the degree of **Doctor of Philosophy in Computer Science and Engineering** is record of bonafide research work carried out by me under the guidance and supervision of **Dr. T. VENGATTARAMAN** and that has not formed the basis for the award of any other degree by any University/Institution before.

Place : Puducherry **R. RAJAKUMAR**

Date : 01-03-2019

Abstract

Large scale optimization (LSO) is non-linear, high complex, multi-modal in nature that consist of large number of decision variables. Traditional optimization algorithms fail to solve these problems because of serious local optima. In this study, LSO problems are chosen as the major concern and to provide efficient algorithm for handling the same. Most of the research work are contributed on traditional algorithms by refining its superior global search ability and efficient technique on handling scaling-up problems.

Grey Wolf Optimization (GWO) is a recently proposed optimization algorithm which mimics the behavior of grey wolves. This algorithm has superior search mechanism into two parts viz., leadership and hunting strategies. This algorithm proves its efficiency on various small scale optimization problems. However, it degrades its performance in handling the large scale optimization problems. As like as, all conventional algorithm GWO also face the same problems such as premature convergence and local optima stagnation. These issues degrade the performance of the GWO in case of handling the large scale optimization problems.

The work presented in this thesis mainly focuses on the qualitative study of Large scale optimization using modified Grey Wolf Optimization algorithm. Modified GWO is proposed with the name Self-Adaptive Grey Wolf Optimization (SAGWO). In the proposed work, three phases are introduced into classical GWO; first phase exploits the search space by enhancing the learning behavior while second phase enhance the diversity to explore the global search space and final phase global best oscillation scheme performed to oscillate the search region in order to eradicate the local optima.

In similar work, the performance study of the SAGWO algorithm is carried out on large scale benchmark functions over varying dimensionality (i.e. 100- 1000 Dimensions). The performance result of the SAGWO using statistical analysis is used to measure the robustness of the best proposed method. In addition to that, other state-of-the-art meta-heuristics algorithms are used to determine the efficiency of the algorithm.

Furthermore, large scale real-time applications viz., Economic Load Dispatch (ELD) and Localization problem are chosen to analyse the performance of the SAGWO algorithm. In Economic Load Dispatch problem, the number of generating units are varied from 10-640 units to notify the efficacy of the proposed algorithm by varying dimensionality. In Localization problem, the number of unknown nodes with respect of anchor nodes and transmission range are utilized to determine the robustness of the algorithm. The result analysis especially success rate will help the researchers to determine the repetition of the best solution for different independent runs.

Keywords: Large Scale Optimization, Grey Wolf Optimization, Self-Adaptive Grey Wolf Optimization, Premature Convergence, Local optima, Economic Load Dispatch, Localization Problem.

ACKNOWLEDGEMENT

First of all, I thank the Almighty for blessing, wisdom and everything

At first, I express my deepest sense of gratitude towards my supervisor **Dr. T. Vengattaraman**, for his guidance, constant encouragement, inspiring suggestions and critical discussions throughout my doctoral studies. Throughout my thesis writing period, he provided encouragement, sound advice, good teaching, and lots of good ideas. I also thank his for his patience and kindness. His dedication to work and perfectionism will have a far-reaching impact on my life, academic and beyond.

I must express my heartfelt gratitude and thanks to my Doctoral Committee members **Prof. P. Dhavachelvan**, Department of Computer Science, Pondicherry University and **Dr. S. Janakiraman**, Department of Banking Technology, School of Management, Pondicherry University for their valuable suggestions and support throughout my research work.

I would like to express my gratitude to **Prof. R. Subramanian** for his constant motivation and encouragement. I would like to thanks to **Dr. P. Sujatha** for their support. I also wish to thank the **Dr. T. Chithralekha**, Associate Professor & Head (i/c) and **Prof. P. Dhanavanthan**, Professor & Dean (i/c) for their moral support.

I also wish to thank my fellow research scholars and all the teaching & non-teaching members of the Department of Computer Science, Pondicherry University for their cooperation and moral support throughout my research.

I would like to thank my beloved Parents, **Mr. R. Ramalingam** and **Mrs. R. Allimuthu** and Sister, **Mrs. R. Rajalakshmi** and brother-in-law **Mr. D. Manivannan** and to my lovable one for their unconditional love, support and eager to know about every advancement in my work.

[R. RAJAKUMAR]

Table of Contents

List of Figures

List of Tables

List of Abbreviations

SSO	Small Scale Optimization
LSO	Large Scale Optimization
ELD	Economic Load Dispatch
SA	Simulated Annealing
GA	Genetic Algorithm
ACO	Ant Colony Optimization
DE	Differential Evolution
CR	Crossover rate in DR and GA
CSGM	Chemical Shifts Guided Modeling
DE	Differential Evolution
PSO	Particle Swarm Optimization
ABC	Artificial Bee Colony
CS	Cuckoo Search
FA	Firefly Algorithm
GWO	Grey Wolf Optimization
EA	Evolutionary Algorithm
SI	Swarm Intelligence
DG	Dynamic Grouping
GRASP	Greedy Randomized Adaptive Search Procedure
CPSO	Competitive PSO
MOS	Multiple Offspring Sampling
SAGWO	Self Adaptive Grey Wolf Optimization
CC	Cooperative Coevolution
FEP	Fast Evolutionary Programming
SaNSDE	Neighborhood Search based Self adaptive DE algorithm
CCOABC	Cooperative Evolution Orthogonal based ABC
UEP	Unbaised Evolutionary Programm
WSN	Wireless Sensor Network
MS	Master Slave
IS	Initial Swarm Size
NS	Number of Swarm Size
GNS	Guided Neighborhood Search
GBOS	Global Best Oscillation Scheme

DM	Diversity Measure
NS	Non-Separable Function
ASD	Average Square Devation
SR	Success Rate
ABO	Average Best Objective
mDE-bES	Multi Population Differential Evolution with balanced Ensemble of Mutation Strategies
MCSO	Modified Competitive Swarm Optimization
DPSOEP	Dynamic Particle Swarm Optimization with Escaping Prey
CSO	Competitive Swarm Optimization
OGWO	Orthogonal Grey Wolf Optimization
LFA	Lightning Flash Algorithm
TFE	Total Number of Function Evaluations
MBA	Modified Bat Algorithm
EL	Total Localization Error
NNL	Number of Non-Localized Nodes

Chapter 1

Introduction

1.1 Overview

Now-a-days optimization is a booming research topic due to its necessity on wide range of application within the several problem domains. The goal of the optimization is to identify the system as effective as possible namely reducing the total computation cost of a system. Mostly optimization works on the system parameters to identify the best combination of the parameters in order to achieve the optimal results. The computation system with its parameters is considered as an optimization problem with its input parameters or decision variables. Based on the decision variables, the optimization problems are categorized into two different categories viz. global optimization problem and combinatorial optimization problems. The global optimization problems deal with the continuous real values whereas combinatorial optimization problems hold discrete real values [Mani et al. 2016].

Further these problems are categorized into two division based on the size or volume of the decision variables. Firstly, the problems with the minimum number of decision variables are considered as the Small Scale Optimization (SSO) problems (i.e. the decision variables of a problem will holds less than 100 variables). At the same time, SSO problems can be easily solved by various optimization algorithms. Secondly, the problems which deal with more or huge or extreme number of decision variables are considered as Large Scale Optimization (LSO) problems (i.e. the problem with more than 100 decision variables). These kinds of problems are quite difficult because of large number of decision variables which in result increases the complexity of the search space. This thesis mainly contributes on the LSO problems by providing an efficient optimization algorithm.

Most of the real-world science and engineering optimization problems deal with many decision variables, known as Large Scale Optimization (LSO) problems. In general, LSO problem can be mathematically expressed as follows (without loss of generality, minimization problem is considered here):

$$minf(x), \overrightarrow{x} = [x_1, x_2, ..., x_D] \in \mathbb{R}^D; [x_j^L, x_j^U], \forall j = 1, 2, ..., D \tag{1.1}$$

where f is the fitness function, $\overrightarrow{x}$ is the decision vector $\mathbb{R}^D$ space with D dimensions, D is the dimension of a problem, i.e., the number of variables are large ($D > 100$) to be optimized, and x_j^U and x_j^L are the upper and lower boundary regions of each decision vectors, respectively. The problem with decision vector which is greater than hundred makes crucial task to optimize [Mohamed 2017].

Most frequent LSO problems are large scale electronic system designing, handling huge resources for scheduling problems, vehicle routing in large scale traffic networks [Mahmoudi and Zhou 2016], gene recognition in bio-informatics, reversible problem in chemical kinetics, satellite layout design [Teng et al. 2010], seismic waveform inversion [Wang and Gao 2012] etc. For example, economic load dispatch (ELD) problem is a large-scale and highly complex optimization problem which has high number of generating units with different cost function viz., value points and multiple fuel options. Considering both, the constraints and cost functions may change the characteristics of the ELD problem into complex optimization problem with non-convexity, non-linearity and high dimensionality. The complexities are due to the high decision space and constraints on operating the generating units such as the spinning reserve, transmission losses, prohibited operation zones, value point effects and multiple fuel options [Meng et al. 2015].

1.2 Metaheuristics on LSO Problems

Generally, meta-heuristic approaches provide a fruitful solution to optimization problems because of its simplicity and flexibility. Traditional meta-heuristic algorithms such as Simulated Annealing (SA) [Kirkpatrick et al. 1983], Genetic Algorithm (GA) [Holland 1992], Ant Colony Optimization (ACO) [Dorigo and Di Caro 1999], Differential Evolution (DE) [Storn and Price 1997], Particle Swarm Optimization (PSO) [Poli et al. 2007] and Artificial Bee Colony (ABC) [Karaboga and Basturk 2007] have been applied in various fields of science and engineering. During the last few years, new algorithms such as Cuckoo search (CS) [Yang and Deb

2010], Firefly Algorithm (FA) [Yang 2010], Grey wolf optimization (GWO) [Mirjalili et al. 2014] and etc., have become familiar and showed its performance on various problems. All the meta-heuristic algorithms rely on two folds viz. diversification and intensification. Diversification means the search space of a problem is globally investigated, whereas intensification searches to a specific search region instead of utilizing global search regions.

In addition to that, meta-heuristics algorithm are divided into two approaches viz. Evolutionary Algorithms (EAs) and Swarm Intelligence (SI) algorithms [BoussaïD et al. 2013]. Evolutionary algorithm is inspired from the natural evolution. Among the several variants of EAs, Differential Evolution is a more popular optimization algorithm in solving various optimization problems. The differential evolution algorithm is an efficient, powerful and straightforward optimization algorithm. Based on the literature, we noticed that DE provides better performance than several other variants of EAs in terms of convergence speed and robustness over the several benchmark functions and real-time problems. However, DE was only tested on functions of up to 100 dimensions. On the other side, Swarm Intelligence (SI) is another category of meta-heuristic algorithms, which mimics from the group of animal behavior (i.e. which observes a scientific study about everything animal can perform). The foremost algorithm is Ant Colony Optimization (ACO), which was inspired by the foraging behavior of ants. Another SI algorithm is particle swarm optimization (PSO), which was inspired from the collective behavior of birds flocking or fish schooling.

The major hindrance of traditional meta-heuristics approaches is suffers from main scarcity due to curse of dimensionality [Omidvar et al. 2017]; i.e., the performance of these algorithms deteriorates when handling the large number of dimensional problems because of the neighborhood search space becomes so narrow that it becomes very difficult to locate the global optimal solutions. Moreover, these approaches require large number of function evaluations and quite difficult to solve problems with interaction variables and complex search spaces [Sun et al. 2017; Salomon 1996].

On the other hand, various design and development of new techniques are integrated with meta-heuristic algorithms to tackle LSO problems viz. local search ability, population reduction, surrogate modeling to handle the large scale optimization problems [Antonio Latorre et al 2015]. In local search ability, dynamic multi swarm concept is proposed and integrated with the particle swarm optimizer in addition to that randomized neighborhood topologies cooperatively work to solve the LSO problems. In addition to that simplex mathematical search model is utilized to enhance the stochastic nature of the optimization algorithms. In population reduction approach, the size of the swarm is reduced by dynamically changing its size based on the dimensionality of the problem whereas in surrogate modeling determines the promising regions of optimal point within the considerable budgets.

1.3 Techniques on Large Scale Optimization

Broadly speaking, several techniques are addressed for solving the large-scale problems viz. reduction in dimensionality [Jolliffe 1986; Schölkopf et al. 1998] and function approximation approaches [Timan 2014]. Another alternative approach is decomposing methodology which splits the large-scale problem into a group of smaller subsets which can be easier to manage and solve. After the decomposition is processed the whole large-scale problem can be solved separately by optimizing the individual subproblems. This decomposition method also determined as the divide-and-conquer method [Descartes 1993] which states that *"[the division of] each of the problems is examined into as many parts as possible, and as quite adequate to obtain the saddle point or appropriate solution"*.

The efficacy of problem decomposition method has been utilized in many standard optimization algorithms [Bertsekas 1999]. In order to split up the decision vectors different models are used namely static grouping and dynamic grouping. Static grouping divides the decision space into fixed size for all the sub components but delivers poor performance on non separable problems where as dynamic grouping (DG) determines the interaction variables in a high dimensional vector based on the variable interaction learning. In particular DG has the following major hindrance.

1. Requires large number of function evolutions which in results depict the high computational cost on fully independent variable problems (i.e. fully separable).

2. Inefficacy to identify objective functions with overlapping decision vectors, i.e. some decision vectors are mutually connected with other decision vectors.

3. Sensitivity to computational roundoff error and a threshold parameter is needed from the user without knowledge of the problems.

Non-decomposition based approaches are considered as an alternative approach of decomposition based approaches. This approach avoids the divide-and-conquer strategy and rather *it applies different strategies to enhance the performance of the algorithms.* At the same time, this approach *best suit for solving both separable and non-separable problems* with the help of the enriched search mechanisms. [Prabhujit Mohapatra] addressed that these approach has been classified into three different process thus they are as local search based, evolutionary based and swarm intelligence based approaches. Firstly, Local search based method raise the different methodologies to solve the optimization problems which includes tabu search, simulated annealing, variable neighborhood search, and Greedy Randomized Adaptive Search Procedure (GRASP) and so on.

Secondly, Evolutionary based approaches is inspired by the biological evolution with different process such as reproduction, crossover, mutation, and selection. Brest and Maučec 2011 performed to reduce the population size with three different mutation strategies to solve the large scale optimization problems. Yang et al. 2011 introduced Scalability of generalized adaptive differential evolution for large-scale continuous optimization in which parameter adaptation is performed to improve the performance of the algorithm.

Thirdly, Swarm Intelligence (SI) algorithm which observes the intelligent behaviors such as searching, flocking, feeding, schooling, attacking of animals such as ant, birds, honey bees, grey wolves and so on. Generically, SI holds unique features to solve optimization problems thus they maintain previous search space information for course of iterations, it *stores the iterative best solution, holds few adjustable parameters* and easy to implement. Though, SI holds intelligence behavior, it lags in efficient exploration and exploitation search mechanisms and balancing the same for scaling problems. Tri-competitive mechanism has been addressed in [Mohapatra et al. 2017] to explore the search space and speedup the convergence rate. [Zhao et al. 2008] provided dynamic multi-swarm technique with local search to improve particle swarm optimization approach.

With these perspectives, the work presented in this thesis is aimed at proposing a swarm intelligence based non-decomposition approach with efficient search capabilities, diversity and also to introduce an efficient self-adaptive property to obtain the potential solution for high dimensional complex optimization problems with faster convergence rate.

1.4 Characteristics of LSO Problems

Large scale optimization problem deals with the large number of decision variables makes the problem quite complex. In reality, based on the dimensionality of the problem the number of objective assessment increases. The properties of optimization process in low dimensional problems might vary in case of high dimensional problems.

The problems with increase in dimensional vector may change its characteristics. Thus the characteristics of LSO problems are commonly seen in the real world problems such as variable dependent and variable independent [R Cheng et al 2016]. Furthermore, they are categorized into four major divisions namely separable, partially separable, fully non-separable and partially additively separable. For all the description of properties we have considered minimization problems. In order to provide an example for each characteristic we have used CEC 2010 benchmark functions [Tang et al. 2009].

Fully Separable

The variables in separable problems does not interact with any of the other variables (i.e. all the decision variables are independent to each other). These variables are easily divided into several subcomponents and can be optimized separately. Without loss of generality, the problem may be considered as separable one if and only if consists the following properties.

$$\arg\min_{x} f(x) = (\arg\min_{x_i} f(x), \arg\min_{\forall x_j, j\neq i} f(x) \tag{1.2}$$

where x is a decision vector of D dimensions in which $x \in (x_1, ...x_D)$. The representation of $\arg\min_{x_i}$ denotes the optimum value of x_i is determined while all the other decision vectors are kept stable. In CEC 2010, sphere function is considered as the fully separable function which has the high dimensional decision vectors of 1000 dimensions. The function is given as follows

$$F_{Sphere}(x) = \sum_{i=1}^{D} x_i^2 \tag{1.3}$$

where D is the dimension and $x \in (x_1, ...x_D)$ is a D-dimensional decision vector.

This function is very simple and is mainly used for demonstration. MLSoft algorithm [Omidvar et al. 2014] based on reinforcement learning and CPSO algorithm based on arbitrary pair wise competition mechanism [Cheng and Jin 2015] provides better result for this function.

Partially Separable

A Function $f(x)$ with m variables of that small number of variables acts as dependent variables and remaining variables act as independent variables. In optimization process especially for decomposition approaches the dependent variables are to be grouped in one subcomponent and independent variables can be divided into several sub components where as non-decomposition approaches will utilize the entire components of decision vectors without dividing it into several components. Without loss of generality, the partial separable problem is mathematically expressed as follows.

$$\arg\min_{x} f(x) = (\arg\min_{x_1} f(x_1, ...x_n), \arg\min_{x_m} f(x_n + 1, ...x_m) \tag{1.4}$$

where x is a decision vector of D dimensions in which $x \in (x_1, ...x_D), x_1, ...x_m$ are partially separable sub vectors of x, and $n \leq m \leq D$. The function $D/2m$ grouped shifted m-dimensional Rosenbrock from CEC 2010 has the characteristic of partially separable in which DM-HDMR [Mahdavi et al. 2014] based on high dimensional model representation technique and FT-DNPSO with dynamic neighborhood topology [Fan et al. 2014] provides better result for this function.

Fully Non-Separable

A function $f(x)$ with m decision vectors in which all the decision vectors in a problem are to be optimized as single component. This type of problem cannot be divided into sub components and every pair of its decision variables interact with each other. This problem is entirely different from the separable problems $(i.e.m = D)$. The representation of fully non separable problem is given as

$$\arg\min_{x} f(x) = (\arg\min_{x} f(x_1, ...x_m) \tag{1.5}$$

Where $x \in (x_1, ...x_m)$ is the solution with m decision vectors $m \in D$. Shifted Schwefels Problem 1.2 from CEC 2010 has the fully non separable properties in which all the decision vectors are interact with each other. The mathematical representation of this function is given as follows:

$$F_{Schwefel}(x) = \sum_{i=1}^{n}(\sum_{j=1}^{i} x_i)^2 \tag{1.6}$$

Where D is the dimension vector ($i.e. n \in D$) and x is the decision vector of $D-$dimensional space. [Zhang et al 2017] proposed an efficient mechanism with assisted Trust-Tech methodology provides better results for this function.

Partially Additively Separable

Partially additively separable is an extension characteristics of partially separable in which decision vectors are mutually exclusive with each other. This function holds multiple independent decision vectors and small number of non separable vectors. The mathematical notation of this type is provided as follows

$$f_(x) = \sum_{i=1}^{m} f_i(x_i) \tag{1.7}$$

Where x_i are mutually exclusive decision variables of f_i, x is the global decision variables in which $x \in (x_1, ...x_D)$ and m is the independent subcomponent. The D/m group shifted and m rotated elliptic function is a type of CEC 2010 benchmark function which has the D/m-group m-rotated and D/m-group m-nonseparable properties in which MMTS based on the modified version of multiple strategy search [Ali et al. 2016] and RPSO-vm with velocity modulation and restarting approach [García-Nieto and Alba 2011] used for the efficient searching and solving the problems.

Apart from that, the achievement in obtaining the optimal solution for LSO problems using a meta-heuristic algorithm must have the inherent characteristics of coordination, collaboration and cooperation among the individuals. These characteristics serve the algorithm to maintain a tradeoff between the global diversification and local intensification capabilities. [Wu et al. 2016; Yu and Zhang 2011; Li et al. 2016; Chen et al. 2017a] multi-population based approach in differential evolution algorithm realize an ensemble of multiple individuals interactions based on the recombination and mutation strategies.

Multiple Offspring sampling (MOS) [de la Fuente and Sánchez 2009] and vector generation strategy [C Segura et al 2015] provides the natural extension to avoid the huge reduction in the diversification for the individuals. Two stage variable interaction mechanism [H Ge et al 2017] and differential grouping strategies [Omidva et al 2014] allows decomposing the complex search space problem and collaboratively optimizing the decision vectors manageable of a complex problem.

In addition to these, the adequate nature derived multi agents interactions are much efficient to handle the interactive variables problem in which sufficient learning and cooperation are exhibited from the nature. Incremental Social learning has been adopted with increasing population size mechanism with PSO to observe the collective information from various particles and also to exploit the current search space [Lanzarini et al. 2008]. Segment-based predominant learning mechanism is one among in the series, in which variables from multiple segments are updated by observing the characteristics of various exemplars.

With the advancements, opposition based learning [Mahdavi et al. 2017] and self organizing migration algorithm [Zelinka 2004] provides the sufficient learning based opposition over the search agents as well as the migration over the dynamic changes on the surroundings and global environment. In addition to that, Joint operation based approach [Sun et al. 2016] and predator prey model [Deb et al. 2005] favors the competitive technique to survive in the environment. The search agents in these approaches cooperatively work to achieve the appropriate position in the competitive environment. Furthermore, reorganization strategy is included to enhance the collaboration between the search agents.

In this view, meta-heuristics must have all the set of characteristics to solve the LSO problems. This dissertation presented a new variant of meta-heuristic algorithm with the enhanced mechanism of having all the capabilities to tackle the LSO problems.

1.5 Challenges

LSO problems have different challenges thus they are listed as follows:

1. The search space of the problem may scale with respect to increase in number of decision variables. For example, in an optimization of binary problem, when the decision variables increase from 10 to 20, then the size of the search space increases from $2^{10} = 1024$ to $2^{20} = 1048576$. In the 10 decision variables the search space of the problem is low whereas in case of 20 decision

variables the search space of the problem grows suddenly and makes hindrance for optimization algorithms.

2. The characteristics of an optimization problem may change as the number of decision vector increases. For example, Let us consider the Rosenbrock function which acts as uni-modal function in two dimensions whereas the same function converts into multimodal modal when the number of decision vectors increases.

3. At the same time, solving LSO problems using meta-heuristics algorithm requires more number of function evaluations in order to locate the global optimal or near optimal solution thus in result it turns the algorithm into computationally high cost.

4. Another measure of the complicate LSO problems is interactions between the decision variables. For example, consider a function $f(x) = 3x_1^2 + 4x_2^2$, the global optimum of each decision vector can be easily found independently from the other decision vector. However, in function $f(x) = 3x_1^2 + 2x_1x_2 + 4x_2^2$, the expression $2x_1x_2$ has variable interaction, which affects the optimizer capability by picking the value from one decision vector in order to locate the global optimum of other decision vector. Generally, the interaction of the variables termed as non-separability problem.

1.6 Motivations

The challenges which are discussed in section 1.5 motivate us to design and develop an efficient and effective optimization algorithm with betterment in providing high quality solutions and high convergence performance with minimal computational cost. The foremost attempt to solve large-scale optimization problems by using enhanced feature of evolutionary programming was achieved by [Yao et al 1999], in which 1000 decision vector has been utilized. Most of the large scale optimization problems are quite complex, in which traditional meta-heuristics approaches fail to find the appropriate solutions. The complexity of an optimization problem increases based on the number of decision vectors. In literature various algorithm are introduced to solve the LSO problems. In genetic algorithm, the non-separability is considered as the gene interaction or epistasis. In most of the case, optimizing the LSO problems with no interaction on variables are to be solved by optimizing the variables independently. In another aspect, the LSO with variable interaction are to be optimized together. In that we notify that most of the real-world problems reside in the both cases.

On the other hand, Grey wolf optimization (GWO) is a recently proposed optimization algorithm belong to family of swarm intelligence [Mirjalili et al. 2014] and widely used by various researchers because of the its simplicity and ease of implementation. Naturally, Grey wolves have effective social and intelligent co-operative behavior compared to other living organisms. They live in a pack and follow very strict dominant hierarchy in addition to that it has excellent leadership and hunting strategies to survive in a complex natural environment. Based on this inspiration, we chosen the grey wolf optimization algorithm to handle large scale optimization problems.

In many recent studies, GWO has been applied on various kind of real-world problems namely Economic load dispatch [Pradhan et al. 2016], Clustering problems [Zhang and Zhou 2015], Wind Speed forecasting [Madhiarasan and Deepa 2016], engineering design problems [Kohli and Arora 2017]. Most of the works and developments on the GWO are contributed to solve the low dimensional problems. In addition to that, we noticed that GWO fails to handle to high dimensional problems because of scalability issues and ineffective search mechanisms which in result provide poor convergence and stagnation in local optimal point. Furthermore, parameters in GWO are problem-dependent and it is difficult to adjust them for different problems.

Hence, the co-operative behavior of grey wolves motivated us to choose the Grey wolf optimization algorithm to handle both the separable and non-separable problems with high dimensional complex search space. In addition to this, it is also aimed at improving the algorithm using efficient search mechanism and self-adaptive properties to solve various kind of real-world large scale problems.

1.7 Our Contributions

As discussed earlier, Grey Wolf Optimization has intrinsic cooperative and grouping behavior which plays a vital role in determining the quality of searching to obtain the optimal solution. In addition to that, though various searching mechanism is integrated with GWO only few modified algorithms provides potential position for complex problems. Thus, the efficient searching and self adaptive properties must be supplemented with GWO for effective search capability.

In this perspective, an enhanced approach of self adaptive grey wolf optimization algorithm with the combination of efficient searching strategies and self adaptive properties is devised and experimental evaluation schemes are intended to validate the significance of the proposed approach with respect to the existing best working approaches of large scale meta-heuristic algorithms.

To accomplish the above, this experimental research is being organized into several phases and the major contributions are listed as follows:

1. An extensive survey has been made over the recent related works and it has been concluded with the necessity for having an improved model of GWO with enhanced search mechanism and self adaptive properties.

2. Three layer experimental framework is being designed to practically prove the asserts made in this research work.

3. Proposed an efficient grey wolf optimization based on self adaptive properties, with guided neighborhood search, position repulsion operator and global best oscillation scheme has been incorporated for efficient search space exploration and solution exploitation characteristics.

4. Applied the statistical techniques to analyze the results obtained from experiments in order to validate the outcomes of the research presented in this dissertation.

1.8 Organization of Chapters

The remainder of this thesis is organized as follows:

- **Chapter 2** presents a brief review on recent contributions related to the work presented in this dissertation. The review is organized into two sections based on the categorization of large scale optimization techniques. Discussion in the first section is focused on the decomposition based approaches, whereas the next section is focused on the non-decomposition based approaches.

- **Chapter 3** defines the goal of this research and explains the line of research to achieve the goals defined. This chapter also defines the layered framework of experimentation methodology followed in this research.

- **Chapter 4** elucidates the formulation of self adaptive strategies with multi-swarm approach. Further, an efficient guided neighborhood search mechanism and its effects on generic GWO are presented.
- **Chapter 5** provides the resting proposed work of the SAGWO with Position repulsion mechanism to enrich the exploration. In addition to that, the detailed flow of the proposed work and its algorithm are described.
- **Chapter 6** describes about the experimentation and result analyses of proposed self adaptive strategies on GWO. Here, the large scale benchmark functions are considered and performances are measured based on the varying dimensionality of the functions. This chapter justifies the performance improvement of proposed model over the best existing meta-heuristic algorithms for the large scale benchmark functions.
- **Chapter 7** illustrates the experimentation over Real-time applications viz., Economic Load Dispatch (ELD) and Localization problem in WSN. Problem specific performance factors are considered to analyse the improvement in performance proposed SAGWO than classical algorithms.
- **Chapter 8** provides the concluding remarks of the work presented in this dissertation and the future enhancements of the proposed line of research.

Chapter 2

Literature Survey

2.1 Introduction

In recent times, several researchers from various fields have contributed on developing a novel optimization approaches to solve the large scale problems due to its necessity in the field of science and engineering. Various optimization algorithms such as competitive swarm optimization, differential evolution, particle swarm optimization, genetic algorithm, artificial bee colony algorithm and so on are introduced to handle the low dimensional problems whereas these algorithms degrades its performance when the scalability of problem increases [Song et al. 2016]. In order to improve these algorithms various modified approaches and novel search techniques are proposed and integrated with the generic optimization algorithm to handle the high dimensional problems [Mahdavi et al. 2014].

In this section, a detailed survey is provided based on the various techniques and methodologies addressed so far to handle the large scale optimization problems. This review helps to find the research gap in the recent approaches on considering the scalability of the problems and also the effective methodologies to solve it efficiently.

Generally, LSO techniques are classified into two major categories namely Decomposition based approaches and non-decomposition based approaches. The classification of LSO techniques is shown in figure 2.1. Decomposition based approaches works as like divide and conquers mechanism which decomposes the LSO problems into single-variable or n number of low dimensional subcomponents. However, non decomposition based approaches utilize the entire decision vectors instead of dividing the decision vectors into n chunks. These approaches use different effective search operators along with various optimization algorithms to enhance their performance in handling large scale problems.

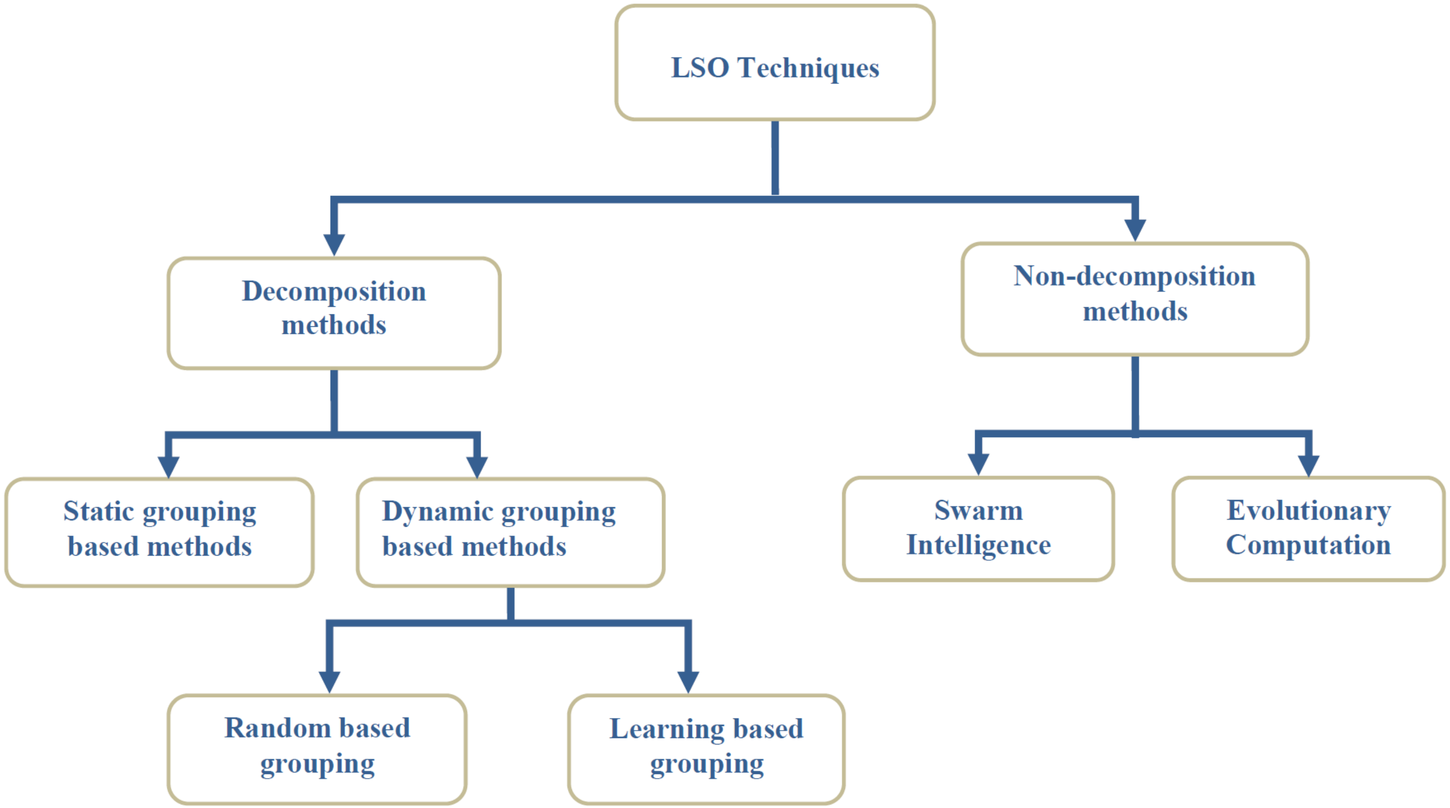

Figure 2.1: Classification of LSO techniques

2.2 Decomposition Based Approaches

Broadly speaking, decomposition based approach initially designed and proposed by [Potter 1997; Potter and De Jong 1994], this approach firstly integrated with generic GA to increase the performance in solving high dimensional problem. These approaches are also termed as Cooperative Coevolution (CC) approach which works based on the divide and conquer strategy. Generally, CC approaches were single dimensional and also dividing-in-half method [Potter and Jong 2000]. The single dimension based approach splits an n-dimensional problem into n sub dimensional problems whereas the dividing-in-half approach splits an n-dimensional problem into two halves ($i.e. n/2$) sub components. The basic steps of CC approach are as follows

- *Problem decomposition:* Large dimensional decision vector decomposes into several small non-overlapping subcomponents.
- *Subcomponent optimization:* Each subcomponent separately executed in a conventional optimization method for a certain number of iterations based on round-robin methodology.
- *Subcomponent combination:* Combining the solutions of all subcomponents to form an n-dimensional solution.

In subcomponent optimization step, n-dimensional decision vector are formed by integrating the each subcomponent solution with the selected solution from the other sub components. In order to merge the solution in each subcomponent [Potter and De Jong 1994] suggested two collaboration techniques thus they are best fitness based and random fitness based collaboration techniques. The best fitness based collaboration technique addressed in CCGA-1 algorithm which calculates the fitness of each solution by merging it with the present best solutions of other components. The random collaboration technique is addressed in CCGA-2 algorithm which calculates the fitness of each solution by merging it with the arbitrarily chosen solutions of other subcomponents. The decomposition based approach has been widely applied on enormous amount of real-world problems [Barrière and Lutton 2009; Cao et al. 2008; Mingming et al. 2011]. This approach splits the decision vector into n groups of variables and cooperatively solves them for a certain number of iterations. Based on the several studies, these approaches are further classified into two categories namely static and dynamic grouping techniques.

2.2.1 Static Grouping Based Decomposition Approaches

Potter and De Jong (1994) introduced a novel decomposition approaches for the enhancement of generic GA with two algorithms namely CCGA-1 and CCGA-2 algorithms which are executed on 30 dimensional problems. The results of CCGA-1 shows better performance on separable problems compare to generic GA, whereas the same algorithm degrades its performance on non-separable problems. A problem with independent variables referred to as separable problems in which the decision vectors are has no interaction between them whereas non separable problems refers to dependent variables which is also termed as epistasis problems.

Liu et al. (2001) introduced the CC approach with fast evolutionary programming (FEP) namely FEPCC to optimize the real-valued benchmark functions with 100 to 1000 decision variables. The author in FEPCC approach states the inefficacy of CCGA algorithm on dealing with non-separable problems. Van den Bergh and Engelbrecht (2004) initially attempt to combine CC approach with PSO algorithms in order to produce a two variants namely CPSO-HK and CPSO-SK. The CPSO-SK observes the original CC approach proposed by [Potter and De Jong 1994] in which the decision vectors are divided into K s-dimensional sub groups (i.e. $n = k*s$). The best particles in all K swarms are referred as context vectors y which guide to calculate the objective of a particle in a swarm. In CPSO-HK, the original PSO and CPSO-SK are merged together that is the particles in CPSO-HK perform CPSO-SK in one iteration and original PSO in next iterations. CPSO-SK and original PSO cooperatively work for information exchange that is the best particles in CPSO-SK which is obtained in certain iteration replaces the arbitrarily chosen particles from the original PSO and in next iteration original PSO replaces the arbitrarily chosen particles from the CPSO-SK. These algorithms were tested on 30 dimensional problems.

El-Abd (2010) proposed CABC-S and CABC-H which is similar to Vanden Bergh et al approaches. In addition to that, the static grouping based CC approach is integrated with DE algorithm in which the search space was divided into two halves of $n/2$ dimensional sub components. However, this static grouping based CC approach is efficient on solving low dimensional (i.e. $D > 100$) problems.

2.2.2 Dynamic Grouping Based Decomposition Methods

As the static grouping based CC approaches are inefficient in tackling dependent variables problems, various researchers have been contributed to propose a new approach to identify the dependent variables and these variables are collectively grouped into one subcomponent. Dynamic grouping based CC approaches predominately fluctuates the grouping structure whereas static grouping method has predetermined size of grouping subcomponent. Further, these methods have been classified into two categories namely random and learning based grouping approaches.

2.2.2.1 Random Based Grouping Approaches

Yang et al. (2008a) proposed arbitrary grouping with DE-based CC approach namely DECC-G to tackle non-separable LSO problems with decision vector of 500 and 1000. In this approach, n dimensional decision vector is divided arbitrarily into multiple chunks of low dimensional decision vector with fixed sizes; each chunk is optimized using neighborhood search based self-adaptive DE algorithm (SaNSDE) [Yang et al. 2008b]. An adaptive weighting technique has been addressed in which the each subcomponent have been allocated with the weight factor in order to improve the solution quality. The subcomponent with minimum weight can be easily optimized using optimization algorithm because of the subcomponent will consists low dimension than n-dimensional problem. This algorithm degrades its performance in case of increase in dependent variables.

Yang et al. (2008c) proposed a multilevel cooperative Co-evolution (MLCC) approach to enhance the DECC-G algorithm. This algorithm uses a decomposer pool to split the decision vector into many sub vectors. In addition to that the selected decomposer guides to determine the different group sizes based on the historical performance of the decomposer. The decomposer pool in MLCC is updated iteratively based on the current performance records. The selection of decomposer is measured using self-adaptive approach which uses the previous iterative performance of the decomposer. Omidvar et al. (2010) introduced a strategy to enhance DECC-G and MLCC. In this strategy, they expanded the predefined likelihood of the arbitrary gathering to each number of collaborating factors. Additionally, they demonstrated that the utilization of versatile weighting in the arbitrary weighting gathering which is not much effective while more habitually arbitrary gathering is effective without expanding the quantity of objective assessments, particularly when the number of dependent variables is more than two. Additionally, a straightforward technique for picking a decomposer from the pool in MLCC was presented. Yang et al. (2009) proposed an adaptive weighing process namely JACC-G which improves the original DECC-G algorithm. This process minimizes the computational time and number of function assessments.

Ren and Wu (2013) introduced a new orthogonal artificial bee colony approach (CCOABC) which uses arbitrary crowding strategy to split up the decision vector into different sub components. This approach improves the probability of crowding dependent variables in one group. Each subcomponent is decomposed using the orthogonal design based artificial bee colony to improve its solution quality. The orthogonal design factors determines as a parameter which holds three hierarchy, first hierarchy is the standard one and other two hierarchy are measured as

$$NewX_{ij} = X_{ij} + rand \times (X_{ij} - X_{kj}) NewX_{ij} = X_{ij} + rand \times (X_{ij} - X_{kj}) \quad (2.1)$$

where $NewX_{ij}$ denotes the new position, rand determines the random number between $[-1, 1]$, $i \in 1, 2, ..., N, k \in 1, 2, ..., N$ (N denotes the varying food sources) and $k \neq i, j \in 1, 2, ..., D$ (D is the dimensionality of the problem). In cooperative Co-evolution process, the arbitrary crowding mechanism uses to divide the decision vector into several low dimensional sub groups and each sub groups optimized using CCOABC to improve the solution quality.

2.2.2.2 Learning Based Grouping Approaches

Learning based grouping approaches deliberates an efficient strategy of grouping which focus on prior knowledge of dependent variables in a problem. It gain the experiences of grouping based on the earlier or at the time of optimization. These approaches contribute to crowd the dependent variable in a single group even though the size of the problem increases.

Ray and Yao (2009) proposed a correlation matrix with CC approach in which the decision variables are optimized using standard evolutionary approach for certain iterations. Later the correlation matrix of two half of the populations best are computed in order to split up the decision variables into several sub components. If the value of correlation coefficient of a decision variable is greater than the specified threshold value then the decision variables are crowded in a same subcomponent. Later, [Omidvar et al. 2011] addressed a contribution based cooperative co-evolution (CBCC) in which the subcomponents are chosen based on the global fitness value. This approach eradicates the improper balance between the separable and non-separable components and also improves the utilization of the computation resources more effectively. In addition to that, round-robin based selection approach is addressed to select the subcomponents for optimization process which in results improves the global fitness. This method is suitable to consume the significant quantity of computational instance.

Weicker and Weicker (1999) introduced a simple approach to determine the dependent variables. In this approach, *best* is considered as the best individual obtained so far, new determines the best solution and rand represents arbitrarily chosen solution from the population. Two new solutions are produced based on these three processes

$$X_k = \begin{cases} new_i, & if k = i \\ best_k, & otherwise \end{cases}, X'_k = \begin{cases} new_i, & if k = i \\ rand_j, & if k = j \\ best_j, & otherwise \end{cases} \tag{2.2}$$

The fitness of new solution $f(X')$ is improved than $f(X)$, the probability of interaction between the solutions i and j gets increased. Based on this simple approach [Chen et al. 2010] introduced a CC approach with interaction between variable learning (CCVIL) in which the group sizes change iteratively. This approach holds two processes i.e. learning and optimization. In learning process, the variable interactions are determined based on the similarity among the variables whereas in optimization process the variables are optimized in order to detect the global optimum fitness.

2.3 Non-Decomposition Methods

Moreover standard operators and several strategies in the meta-heuristics approaches are addressed to tackle the small scale or low-dimensional problems and approaches lack in handling the large-scale problems. Several modifications over the classical operators are proposed to improve the meta-heuristics algorithm to show its performance on the large scale problems. The non-decomposition approaches is an alternative approach of decomposition approaches which avoids the divide and conquer mechanism rather it concentrates on improving some strategies such as introducing new selection mechanism, crossover and mutation operators, developing new local search strategies or utilizing existing local search strategies, sampling approach and varying population size mechanism to improve the exploration process to handle the LSO problems.

2.3.1 Evolutionary Computing

MacNish and Yao (2008) introduced an unbiased evolutionary programming (UEP) which observes the directional bias for large dimension problems. In this study, they analyzed the impact of directional bias as the dimensionality of the problem increases. In addition to that, they analyzed the properties of the optimization search space especially for the variable independent and modality problems and also analyzed different parameters which improve the performance of algorithm. Genetic algorithm with a population division approach was introduced by [(Hedar and Ali, 2009)] in which the solution is generated in a single component and then this component is divided into several subcomponents at every iteration. The novel operators such as mutation and crossover are used to optimize the subcomponents. In addition to that, a modified stopping criterion was addressed in [Hedar et al. 2007]. A new evolutionary based search approach, namely SP-UCI was proposed by [Chu et al. 2011a] in which four stages are addressed namely, shuffling complex structure, screening and repairing population dimensionality, enhanced competitive complex development, and different-normal resampling.

Univariate assessment of distribution algorithm namely LSEDA-gl is proposed by [Wang and Li 2008] for tackling LSO problems. In this approach three mechanisms namely mixed Gaussian based sampling approach, Lévy distribution and restart mechanism are utilized in order to improve the performance of EDA. They address the covariance matrix for all variables in order to adapt a random Gaussian and to avoid the non-diagonal elements in the matrix which in result it minimizes the computation time especially for LSO problems. In [Dong et al. 2013], an EDA approach was introduced with the combination of weakly dependent (WI) decision vector and subspace modeling process to manage the difficulty of multivariate method on large dimensional problems. In WI process, the global correlation component is measured and then decision vectors with minimum correlation value compared to specified threshold value are considered as weakly dependent variables. In subspace modeling process, the high dimensional solution space is divided into different subcomponents, and then a multivariate method for each sub component is designed.

Wang et al. (2013a) and Wang and Li (2010) was proposed a two stage based ensemble optimization evolutionary approach (EOEA) which holds two mechanism namely, the global reduction and the local diversification. In global reduction mechanism, Gaussian and Cauchy technique are included in EDA which guides to reduce the searching scope and directs to the potential section. In the local diversification mechanism, sizes of each subcomponent are adjusted adaptively in CC based approach. Each sub component is developed via selected arbitrarily chosen mechanisms. A new sub component is generated with the best fitness values of solutions for certain iterations. Nesic et al. 2012 proposed a sequence parameters for neighborhood search approach which holds three major methods namely generation, enhancement and shaking. These methods produce a Covariance Matrix Adaptation Evolution Strategy (CMA-ES). In addition to that, a novel assortative crossover approach was introduced and integrated with the continuous local EA. In assoratative crossover approach two parents are selected, first parent is selected randomly and second parent is selected using two conditions one is based on the best fitness and fitness closer to first parent. The performance of CMA-ES is observed and compared with other CC approaches.

Mariani et al. (2011) proposed three shuffled complex procedures (namely SCE-UA) with the help of simplex mechanism and it is integrated with PSO and DE algorithms in order to tackle the large scale benchmark problems. Population degeneration strategy is addressed after a certain number of iterations which helps the population to attain the sub space and limits the search region into sub space to determine the global best. Furthermore, Principal components Analysis (PCA) is developed to identify the population degeneration and to eradicate the bad influence on certain individuals.

2.3.1.1 DE Based Algorithm

Differential evolution algorithm attracts most of the researchers due to its simplicity and easy implementation in order to solve the LSO problems. Enormous amount of variants are introduced in DE in order to tackle the LSO problems. Zhang and Sanderson (2009) introduced modified differential evolution namely

JADE which improves the performance of generic algorithm by adopting new mutation strategy 'DE/current-to-pbest' which also holds external archive and updating control parameters. De/current-to-best is the basic approach which utilize the previous historical information of the search direction. The historical data are gathered in an external archive to update the individuals and to improve the diversification and global convergence rate. The control parameters are in an adaptive manner which avoids the predefined setting of values from the users. The parameter values are self determined using the individuals search region and then it performs the mutation and crossover.

In mutation strategy, external archive is utilized to maintain the historical information which helps to intimate the success and failure ratio of an individual. Let A determines the achieve set of inferior solutions and P denotes the current population. In DE/current-to-pbest/1 without archive mechanism two solutions $X_{r1,g}$ and $X_{r2,g}$ are randomly chosen from the population P (i.e $P \in A$) the current search individual updates its current location with the help of randomly chosen solutions whereas in case of archive mechanism one random solution is drawn from the population P and another solution is drawn from the archive set A (i.e. $P \in A$). Additionally mutation factor F_j is computed for each solution X_j based on the Cauchy distribution with the position parameter μF and weighing parameter 0.1. These mutation factor dynamically changes for every iterations when the individual search direction upgrades.

Takahama and Sakai (2012) introduced a modified differential evolution mechanism in order to detect the modality of the problem. The modality of the search space is identified using the fitness function values by setting the sample search point on a direction which the point is computed using the median of search point and the best search solution. When the fitness function varies in an ascending order (i.e. from less to high) then there is one saddle point. If there is one saddle point then the problem is uni-modal problem; or else the search space is multimodal. In this method the scaling factor is fixed using the search space modality identification.

Wang and Gao (2014) introduced a novel selection mechanism based on the strength of the solution and furthermore, global fitness evaluation is addressed to measure all the decision vectors of a solution [Wang and Gao 2010]. For example, Let us consider two individuals A and B then global fitness value is used to identify the individuals. If individual A is better than individual B, then some vectors in individual B holds high quality decision values. In order to maintain, the high quality decision vectors in an individual's local fitness function is used which observed from the inspiration of genetic engineering and modern medicine. The local selection operator divides the high dimensional decision vectors into several low dimensional decision vectors and then local fitness function is assigned to compute each low dimensions. These two fitness based approaches works simultaneously to generate the population.

Fan and Yan (2015) introduced a self-adaptive DE algorithm which includes the population deduction mechanism and a technique for the individual sign adjustment of scale coefficient F. During the search process, the scale coefficient parameter F is determined with a probability based on the fitness values of arbitrarily chosen solution for mutation strategy. Wang et al. (2012) introduced quantized orthogonal crossover mechanism on differential evolution (OXDE) in which two individuals are quantized and then the levels of each individuals search regions are quantized. A problem holds with high dimensional decision vector compare to the number of level then QOX technique is addressed to split up the vector into few sub vectors. This QOX technique helps the algorithm to minimize the computation cost.

Morley and Tricarico (2014) introduced multi-population based shuffled parallel DE which uses two arbitrary methods. At the beginning of search process, the population of the algorithm is scattered into different sub populations and each sub population holds a scale factor. The first method uses shuffled strategies, that is the population is again divided into different sub-populations with a specified probability and the second method updates the scale factors of each sub population by arbitrary sampled values which resides between 0.1 and 1. Yang et al. proposed a generalized parameter adaptation technique to improve the generic algorithm of DE by analyzing the existing adaptation techniques.

Chen and Tseng (2014) proposed an Enhanced Multiple Trajectory Search (EMTS) mechanism which utilizes the estimated orthogonal sequence to generate the initial solutions. In MTS, the multiple solutions are used for diversification and then each solution performs a local search iteratively. Furthermore, a novel local search mechanism is used to strengthen the neighborhood environment. García-Martínez et al. (2011) introduced a modified DE based on two criteria namely role differentiation and malleable mutation. In standard DE, the solutions in a population are arbitrarily chosen to perform either crossover or mutation process. Whereas in case of role differentiation method the appropriate solution are selected to generate the new solution based on the different groups of generation. In addition to that, the malleable mutation strategy is used to adjust the mating capability of fixing vectors in appropriate decision locations.

Iorio and Li (2008) proposed a sampling based DE approach for creating more new individuals using crossover operators in order to enhance the exploration capability. Gomez and Leon (2010) proposed co-evolutionary chromosome generating scheme to quantify the individuals. In this scheme, each subcomponent is quantifying its individuals with the help of information exchange between the subcomponents. Xuemei (2010) introduced a novel mutation strategy in DE which creates a new individual with the help of three individual's namely local best individual, randomly drawn individual and global best individual. In addition to that, Cauchy mutation based approach was proposed by [Pan et al. 2012] which improve the global best if it is not improved for a course of iterations. This mechanism generates a neighbor individual around the global best solution and it updates the global best if and only if generated new neighbor individual is better than the current global best solution.

2.3.1.2 Opposition Based Approaches

Rahnamayan et al. (2006) proposed a novel mechanism based on the concept of opposition based learning in order to improve the standard DE algorithm. This algorithm introduces a two strategies namely, opposition-based population initialization and iterative jumping mechanism. In opposition-based population initialization mechanism, the two subcomponents are generated in which one subcomponent consists randomly generated decision vector and another subcomponent holds opposite number of decision vectors of first subcomponent. The objective values of each subcomponent are quantified based on the union of two subcomponents. At the time search process, the iterative jumping mechanism is quantified to opposite number of subcomponent with the help of prior probability rate. In addition to that, the territory of solutions are measured dynamically based on the upper and lower boundary values of each decision vectors.

Gao et al. (2012) introduced a hybrid opposition based learning and harmony search optimization algorithm in which the integration works simultaneously in the mutation process. Center-based sampling approach was introduced by [Rahnamayan and Wang 2009] in which they uses Euclidean distance in order to provide the closeness probability to an unidentified individuals for the certain positions of a search space in a black-box problem. In addition to that the closeness probability is identified using the Monte-Carlo based simulation. The closeness probability of an unidentified individual for the mid position increases with the increase in dimensionality of the problems. The newly generated samples which are closer to the center position, then samples are considered as the closer position to the unidentified samples.

In addition to that, opposition points are used to compare the features of center position. Mahdavi et al. (2015) introduced center-position based approach in simulated annealing which observes the center position as a starting position in order to improve the performance of an original algorithm. In recent times, a detailed review of initial population seeding schemes are introduced in evolutionary computation by [Kazimipour et al. 2014] in which generic population seeding is initialized based on the three methods namely random based, composition based and generality based schemes. Furthermore, the effectiveness of each population seeding schemes is tested on differential evolution for LSO problems.

2.3.2 Swarm Intelligence

Hsieh et al. (2008) introduced an efficient population utilization strategy for particle swarm optimization (EPUS-PSO). In this algorithm, they includes three novel mechanism namely population supervisor, solution sharing and searching range sharing (SRS). Firstly, population supervisor is used to generate new particles or to maintain the particles which have efficient search capability by eradicating the unnecessary particles. Secondly, they used solution sharing mechanism based on [Li et al. 2015] in which the unified learning probability is identified for each particle in order to adjust its positions. If the arbitrary value is lesser that the computed learning probability value then two neighbor particles are selected from the population P and if the selected particles is better than the particle best then the random particles will share their information to the current particle otherwise particle best share its information.

The solution sharing probability is computed based on the dimension of the problems and size of the population. Thirdly, search range sharing strategy adopts both the local and global searching boundary to each particle. In local search sharing strategy, the search range is restricted to a limited range (i.e. $pbest_{min}$ and $pbest_{max}$ especially for the perturbed particles. This local strategy forces the perturbed particles to exploit its current search environment. In global strategy, the perturbed particles are scattered in the specified boundary region which helps the particles to search the new best position in un-searched region and eradicate the stagnation in local optima.

Lozano et al. (2011) introduced a novel velocity modulation and restarting scheme in particle swarm optimization algorithm in order to handle the scalability problems. Firstly, velocity modulation strategy is used to measure the overall adjustment of the particles iteratively. Velocity modulation strategy controls the fast movement on the particles and guides the particles to search within the appropriate search region. Secondly, restarting strategy is observed from Eshelman and Auger et al. approaches. This strategy works based on the specified termination conditions which includes two independent conditions namely 1) if the entire particles in the swarm is below the threshold1 (i.e. $1e-3$) then the position of particles are arbitrarily initialized based on the probability of inverse dimensionality, 2) if the change in objective function of Specified generation (i.e. $10\times$ dimensions/ population size) is smaller than threshold2 (i.e. $1e-8$) then the particles are regenerated within the global best position.

In addition to that, [Korenaga et al. 2007] proposed a modified PSO based on the rotated particle scheme. In the rotated particle swarm approach, two particles are randomly selected from the population and the diagonal position of each particle is measured based on the previous particle's best. Based on the diagonal measurement the new particle with best position is identified and updates its historical best particles.

Cheng and Jin (2015) introduced new optimization algorithm namely competitive swarm optimizer which works based on the pair wise competitive mechanism. Generally, generic particle swarm algorithm improve based on four different strategies namely, adaptive control parameters, improvement in neighborhood topology, hybrid with other search mechanism and finally multi-swarm concept. In the CSO approach, particles are randomly initialized and then the population P is divided into two swarms (i.e. P/2). From the two swarms particles are arbitrarily chosen for the pair wise competition. The two particles are randomly chosen that is one from the swarm1 and another from the swarm2, the particles which hold highest fitness value then the swarm is considered as the winner and another will be considered as loser. The loser particle learns from the winner particle in order to determine its search direction. In addition to that, the self-adaptive parameters and balanced search mechanism are used to handle the high dimensional problems.

Akay and Karaboga (2012) applied artificial bee colony algorithm to handle unconstrained and constrained large scale optimization problems. Based on the Debs method they selected the bees from the hive to produce a new solution. In order to handle the constrained optimization problem, they used constraint violation mechanism in which the probability values are measured to identify the solutions which are within the feasible regions. Nickabadi et al. (2011) introduced soft adaptive mechanism in particle swarm algorithm in which adaptive inertia weight is presented. In addition to that the acceleration feature is included to balance the global and local search process. The adaptive strategy provides good opportunity to the particles to update its search positions and also helps to eradicate the particles which are move far away from the search region. Each particle in this approach maintains a legible safety distance termed as proximity index in order to regulate the particle to be nearer to the best particles.

Chen and Vargas (2010) introduced a novel algorithm using locust swarms in which the particles adjust its position based on *devour and move on* method. In devour methods the search particles forced to search the local optimum for a certain number of iterations and then *move on* method is processed to jump away from the local optimum. This algorithm best suit for unconstrained multi-modal problems but whenever the scaling or constraints of the problem is high then it degrades the performance on locating the global optimum. Chu et al. (2011b) proposed a boundary handling mechanism in particle swarm optimization which uses three strategies namely random, reflecting and absorbing. In random based boundary handling scheme, the particles which move out of the boundary values are replaced by generating random values using uniform distribution between the upper and lower boundary region and the generated values are replaced with the outside boundary values. In absorbing scheme, instead of replacing the entire particles the specific outside boundary decision vectors are absorbed and replaced. Finally, in case of reflecting approach the particles which moves away from the boundary region are reflected as like a mirror and projected within the boundary.

Fister Jr et al. (2013) introduced a new algorithm namely fast bacterial swarming algorithm (FSBA) in which foraging approach of BFA and swarming mechanism of PSO is combined to cooperatively solve the high dimensional problems. Furthermore, an adaptive step length is addressed to improve the local search process and also attraction factor is used to reduce the premature convergence. Though adaptive step length is used to adjust the position, it slows down the search process and provides poor exploration on multi-modal problems. Gozde and Taplamacioglu (2011) addressed automatic iterative tuning process in particle swarm optimization which eradicates the poor exploration and provides global search capability for scalability problems. The automatic iterative tuning process provides two efficient processes namely the size of the search space is determined for each particles and the parameter values are self-adjusted based on the scalability of the problem.

Wang et al. (2013b) applied two strategies namely diversity enhanced mechanism and neighborhood strategies in particle swarm optimization. These strategies help the algorithm to maintain a balance between the global and local search mechanism. Diversity enhanced mechanism improved based on [Jordehi 2015] in which the repulsion phase is defined to modify the velocity updating process. In order to update the particle, a sample particle is generated based on the historical best particle and global best particle. The sample which has better fitness then the particle is replaced with the current particles otherwise greedy selection mechanism is used to generate the particle. Furthermore, the particles in neighborhood search strategy updates its position based on two methods two methods namely locally and globally mechanism. In local mechanism the particle updates its position using the historical information and randomly chosen particles. In case of global mechanism the particle updates its position using global best particle and two randomly chosen particles. This approach possesses a tradeoff between the search mechanisms. However, setting the number of particles in swarm is quite complex in case of varying high dimensional problems.

2.4 Summary

In this chapter, the literature survey has been presented in two folds. The first one is based on the decomposition based approaches and another one is based on the non-decomposition based approaches.

In the view of decomposition based approaches, various approaches like Cooperative Co-evolution based dynamic grouping and static grouping are introduced but it still lags in solving large scale problem with interacting variables (i.e. finding the exact position of interaction variables in a problem is difficult in order to split up high dimensional decision vector into several sub component structure especially for the unknown high dimensional problems). Whereas, non-decomposition based approaches are considered as an alternative solution for solving both separable and non-separable high dimensional problems. Though these approaches have various parameters tuning strategies, opposition based approaches and learning mechanisms, still it requires an efficient search mechanisms and adaptive parameters in case of handling the varying size of high dimensional problems with complex search space (i.e. high dimensional with multi-optimal problems).

Chapter 3

Goals and Research Methodology

3.1 Introduction

With fast growing technology vast variety of real world problems are quite complex due to curse of dimensionality and vast search space. Most commonly accept that meta-heuristic based algorithms are an alternate solution for dealing huge decision vectors and complex search space problems. But those algorithms are not directly applied to solve LSO problems due to premature convergence and stagnation in local optima. Wide variety of novel techniques are designed and developed to improve those algorithms which are discussed in chapter 2. In addition to that, we observe that the algorithm with the characteristics of cooperation, collaboration and coordination among the search agents may provide the efficient results as well as suits to tackle the large scale problems.

From the wide variety of optimization algorithm, we notify that grey wolf optimization algorithm holds effective leadership and hunting mechanism as well as it has three characteristics naturally as mentioned above. In addition to that, the performance of GWO is superior when compared to other meta-heuristic algorithms (i.e. GA, DE and PSO) with the benefit of holding few control parameters [Mirjalili et al. 2014; Singh and Singh 2017a,b; Joshi and Arora 2017; Diwan and Khan 2016]. Apart from that, GWO has gained much interest among the researchers, because of its simplicity and ease of implementation. This algorithm quite superior in solving small scale practical applications but degrades its performance when dealing with the complex high dimensional problems. Further modifications over GWO might improve the performance of the algorithm to a large extend and adapts for the growth of the search space dimensional problems.

In this perspective, enhanced model of self adaptive strategies are integrated with generic grey wolf optimization for solving large scale optimization problems efficiently. The research goals and experiments of the proposed work are formulated in order to reveal its efficiency with other state-of-art meta-heuristics algorithms over the large scale problems. This chapter deals with the goals of this research work offered in this dissertation along with the outline for experimentation procedure performed for the proposed research.

3.2 Research Goals

The fundamental model of GWO is to fabricate a population of feasible search agents and its positions are adjusted using three best solutions to obtain the optimal point. The traditional GWO has been improved by various researchers using novel exploration schemes and parameter tuning models [Madhiarasan and Deepa 2016; Kohli and Arora 2017; Mittal et al. 2016; Guha et al. 2016; Zhang et al. 2016; Yang et al. 2017] to enrich the performance of the algorithm with the search space diversification and capability to achieve the optimal point. The performance of GWO mostly works based on the iterative best solutions and coefficient parameters. Though the position adjustment in GWO based on these processes, it is not sufficient to tackle the LSO problems when compared to other meta-heuristics algorithms.

Additionally, the modified GWO have been applied on several applications with small instances viz. multilevel thresholding [Khairuzzaman and Chaudhury 2017], optimal reactive power dispatch problem [Wong et al. 2014], optimal power flow [El-Fergany and Hasanien 2015], feature subset selection [Emary et al. 2015], unit commitment problem [Kamboj et al. 2016], evolving kernel extreme learning machine [Wang et al. 2017] and so on and also provides better performance by identifying the optimal solution. However, on dealing large dimensional problems, only some of the variants have the capability to achieve the optimal or near optimal solution and also eradicates the local optima stagnation. Still, they face the problems of local optima struck, slow convergence speed and poor balance on search mechanism especially on high dimensional or large instances problems. The enhancement over the process of interaction, adaptation and learning might improve the performance of the algorithm and helps to handle the large scale problems.

From the above perspective and from the hindrance of existing approaches in grey wolf optimization, the main intension is described as to design and develop self adaptive strategies which have interactive learning and adaptive mechanism for

Grey Wolf Optimization algorithm. The research goals are defined such that to propose an enhanced model for Grey Wolf Optimization algorithm using efficient self adaptive strategies to enhance the performance with respect to computation time, search diversity, variation on population size, scalability of dimensionality and success rate.

The proposed research work was carried out with two goals:

- To propose an efficient self adaptive search schemes to enhance the search process of search agent and to adopt on growth of dimensionality problems.
- To solve various large scale dimensionality benchmark functions and real-time applications.

The preliminary goal is derived to improve the search process of the search agent as well as to adapt on the varying dimensionality problems whereas the next goal is derived to solve the various large scale benchmark functions to show its efficiency among other state-of-art meta-heuristics algorithms. The proposed algorithm is implemented with a suitable set of benchmark functions at different dimensions.

In this view of attaining the goals derived in the research are developed into many divisions and the major objectives are described as follows:

1. To design and develop a multi-swarm approach and guided neighborhood search mechanism.
2. To design and develop a position repulsion mechanism and global best oscillation scheme.
3. To propose an enhanced model for grey wolf optimization by merging all the novel mechanism with self adaptive parameters.
4. To build an experimental framework for analyzing the performance of the proposed algorithm.
    (a) To construct a Test-Bed with the set of large scale benchmark functions at different dimensions and for real time problems viz. economic load dispatch problem as well as for node localization in WSN.
    (b) To validate the performance of the enhanced approach of Grey wolf optimization algorithm.

In looking to contend for these goals, obviously this work varies in flavor from most of the research works. It provides an enhanced model of self adaptive GWO with different search mechanism to improve the performance to the large extent. One of the major features of the proposed work is that it measures the performance of the proposed work utilizing planned test-bed with appropriate performance metrics and benchmark functions with different dimensions. The following section depicts the experimental study attempted to seek after these goals and the general operation performed in this proposed research work.

3.3 Research Methodology

3.3.1 Experimental Framework

The empirical structure of the experimental architecture is described in the figure 3.1. This architecture holds three different levels: Experimental Analysis Layer, Test-Bed Layer and Validation Layer. The functionality of each layer are derived and illustrated in the following sections:

3.3.1.1 Experimentation and Analyses Layer

The experimental setup and the performance proposed methodology of self adaptive grey wolf optimization algorithm are provided in the experimental and analyses layer.

The experimental setup was derived to attain the efficiency of the proposed model under different environmental criteria. The different parameters such as population size, number of function evaluations, and maximum number of iterations, independent runs, guided probability and Lévy step size. In these, initially population size of the proposed algorithm is varied based on the literature [Ali et al. 2015; Nieto et al. 2015] in order to prove the efficiency over different population size as well as the group size is varied to identify the best group size. Number of function evaluations and number of runs are fixed based on the literature [Omidvar et al. 2017; Dong et al. 2013]. Guided probability is static which induce limited search agents to learn and finally the Lévy step size are set based on the literature study [Jensi and Jiji 2016; Haklı and Uğuz 2014]. Experiments are performed

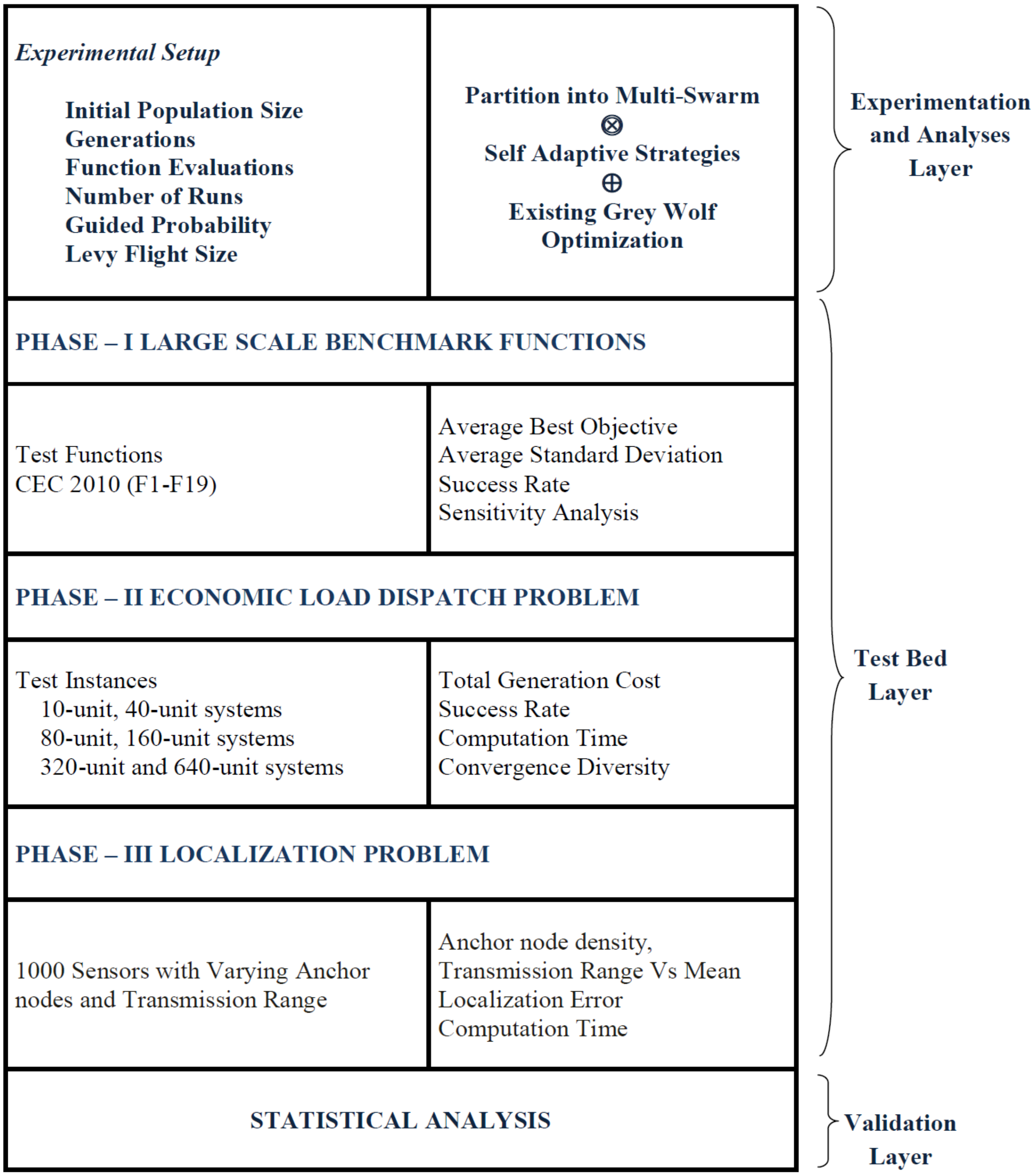

Figure 3.1: Layered View of Experimentation Methodology

using MATLAB tool with three different phases which are discussed in test-bed layer.

3.3.1.2 Test-Bed Layer

In our research contribution, the Test-Bed layer is divided into three different phases. In the initial phase on large scale benchmark functions, second phase on Economic Load dispatch (ELD) problem, and final phase on Localization problem (i.e. Localization of nodes in wireless sensor network).

Phase I - Large scale benchmark functions: In order to evaluate the performance of the proposed approach, the standard large scale benchmark functions are utilized. The test function has different properties such as unimodal, multimodal, separable and non separable and it is defined in CEC 2010 [Zhao et al. 2011]. These test functions are widely used to evaluate the capability of the newly designed and developed metaheuristics especially for handling LSO problems. Most of the works in the literature have utilized these set of standard benchmark functions for scalability and adaptability of the algorithm on scaling high dimensional problems. The objective of this function is to minimize the values of decision variables and to attain the optimal solution. Each of the function has different optimal values and different ranges with either separability or non-separability characteristics. The dimensionality of the function has been varied to maximum of 1000 dimensions which has 1000 number of decision variables.

Performance assessment criteria: Performance assessment criteria are selected and classified into the following factors, namely Quality, Exploration and Time. The Quality based features are the indicator to determine the quality of the solutions under the examination in which error rate and success rate are included. The Exploration based factors aid to determine the capability of algorithm to explore the search agents to a global search space of the problem and to analyze the efficiency on eradicating the local optima struck, convergence diversity has been used. The time based factor determines the efficiency of the algorithm in order to obtain the optimal or near optimal solution with corresponding to its computational time. Additionally, the proposed model is compared with other state-of-art meta-heuristics algorithms to show its efficacy over these performance metrics.

Phase II Economic Load Dispatch Problem: This phase contributes on the performance of the proposed work with respect to the real world problem namely Economic Load Dispatch (ELD) problem. Economic Load dispatch problem is considered as one of the real world large scale problem in which the search space is highly complex and has multi-modality features.

Most of the researchers widely used this problem to evaluate the performance of the algorithm in-terms of capability and high accuracy on obtaining the optimal solution. The test instance of the ELD is collected from the literature study [Zaman et al. 2016; Meng et al. 2016]. The objective of this problem is to generate the total power by reducing the total cost of generations. In that n number of generators with varying load demand is provided with the main intension to obtain the load by minimizing the generation cost. Most of the research works in ELD problem have been experimented only for the 10 unit, 13 unit and 40 unit system and only few considered 80 unit system and large scale of above 80 unit system ($>$ 80 generating unit) varies between 10 to 640 unit system. The large scale benchmark instances of ELD have been performed to analyze the performance of the proposed model.

Performance assessment criteria: In order to evaluate the performance of the proposed approach on ELD problem some of the standard performance metrics are considered based on the literature study [Pradhan et al. 2016; Sahoo et al. 2015]. The main performance metrics are convergence diversity with respect to the obtained best fitness based on the number of generations. In addition to that success rate, which is used to analyze that how efficiently provides the optimal or near optimal for n independent runs. Total generation cost is used to determine that the proposed algorithm how efficiently solves by providing the minimum generating cost with respect to the number of generations. Finally, computation time is used to identify the rapidity of the proposed model to achieve the desired optimal result or the generation limit for the considered complex ELD problem. Additionally, the proposed model is compared with other state-of-art meta-heuristics algorithms to show its efficacy over these performance metrics.

Phase III Localization problem: Localization problem is one of the complex multimodality or non-convex problem which is selected as another test problem. The localization problem is a well-known problem which is to identify or locate the exact position of the unknown node position with the help of known or anchor nodes. Most of the research works have been contributed in this problem to provide a valid solution using meta-heuristics algorithm [Goyal and Patterh 2014; Mihoubi et al. 2018]. The objective of this problem is to locate the position of unknown nodes by minimizing the localization error. Experiments on this problem are performed based on the increase in number of sensor nodes (i.e. 1000 node scenario) with varying the number of anchor nodes.

Performance assessment criteria: Performance assessment criteria for localization problem are chosen as standard metrics based on the literature study. The performance evaluation is performed based on the different anchor nodes how effectively locate the position of unknown node by providing the minimum localization error. Secondly, by considering the different transmission range how effectively the proposed model provides the better results with minimum localization error. The time based factor is considered as computation time to determine the quickness of the proposed model to obtain the required optima solution (or) generation limit for the considered problem. Additionally, the proposed model is compared with the standard state-of-art meta-heuristic algorithms.

3.3.1.3 Validation Layer

Validation layer is used to validate the proposed model with respect to the statistical analysis. In this research work, we have considered Friedman statistical test which is a non-parametric statistical test analysis. This test is used to validate the efficiency of the proposed model comparing to $1\times$ N meta-heuristic algorithms (i.e. N is the number of alternative or other state-of-art algorithms). This test is done by ranking the algorithms based on the performance for each benchmark function.

3.4 Summary

The principal concern in this research is to design and develop an enhanced model of GWO with the multi population approach and self adaptive strategies and to evaluate the performance using standard test problem. A three layer experimentation methodology is formulated and developed for the research described in this chapter. The experimentation and analyses layer consists of the experimental setup and proposed methodology considered with this research. The experimental setup with different parameters has been described. The test bed layer describes the three phases of the test problem and the various performance assessment criteria selected for experimentation to validate the performance of the enhanced model of self adaptive GWO. Finally, the validation layer is responsible to validate the performance of the proposed model with the corresponding other state-of-art meta-heuristics algorithm.

Chapter 4

Proposed Methodology-I

4.1 Introduction

Grey Wolf Optimization (GWO) is a recently proposed swarm intelligence based optimization algorithm proposed by [Mirjalili et al 2014]. GWO is inspired by the behavior of grey wolves which holds efficient leadership and hunting mechanism. A wide variety of researchers from various fields have utilized this optimization algorithm to solve their domain specific problem because of its ease implementation and simplicity. GWO provides sufficient solution for too many real world applications viz. combined heat and power dispatch problem [Jayakumar et al. 2016], multilevel thresholding [Khairuzzaman and Chaudhury 2017], feature subset selection [Emary et al. 2015], optimal reactive power dispatch problem [Wong et al. 2014], train multilayer perceptrons [Mirjalili 2015], unit commitment problem [Kamboj et al. 2016], evolving kernel extreme learning machine [Wang et al. 2017], power system stabilizer design [Shakarami and Davoudkhani 2016] and so on.

Like other meta-heuristics algorithms namely GA and PSO, GWO also degrades its performance on handling the increase in search space dimension problems. Generally, GWO lags in providing faster convergence rate for the problems with single local optima. In addition to that, GWO get stuck into local optima in case of handling large scale complex multi-modal problems. In order to handle these issues, an efficient search mechanism with self adaptive properties are essential for tackling the growth of search space dimension problems.

In our research work, we propose an enhanced model with self-adaptive strategies namely Self-Adaptive Grey Wolf Optimization (SAGWO) which regulates the search agents in GWO to tackle the large-scale problems. The essence of self-adaptive strategy is to deliver a technique to preserve the exploration over the swarm and a novel dynamics to avoid the local optima stuck.

The proposed strategy is combined with GWO to solve the scaling problems in an efficient manner. The self-adaptive strategies are designed with three mechanisms namely 1) Guided Neighborhood search mechanism which observes the experience of search agents from neighbor groups, 2) Position repulsion based mechanism which repulse the search agents to explore the global search space to eradicate the stagnation in local optima basin and 3) Global best based Oscillation mechanism which is used to adjust the global best search agent alone.

With the above perspectives, this section deals with an overview of the proposed framework. In that, Multi-swarm approach and Guided Neighborhood Search strategies are discussed in detail with its advantages on handling the large scale optimization problems and other remaining strategies are discussed in chapter 5.

4.2 Self Adaptive Background of Strategies

Self-adaptive strategy mimics the learning behavior of human beings intergroup interactive approach. The entire population of search agents is sub divided into multiple groups to exchange their learning experiences at the time of searching process. In addition, various existing approaches are designed in the concepts of learning mechanism to induce the global search capability. The main contribution of the proposed work is described as follows.

1. In most of the multi-swarm based meta-heuristic algorithms, the roles of each group are predefined in advance [Niu et al. 2007; Chen and Yu 2005; Liu et al. 2013] as like Master-Slave (MS) mechanism. In this MS mechanism, one group will act as the master and the other group will act as the slave in which the slave group observes the search experience and adjusts its current positions based on the master group experience. In case of self-adaptive approach, it overcome the MS mechanism and has no predefined roles over the search group. The roles of the multi swarm changes dynamically during the search process, for example, the group which act as the master/learned group in current iteration will get a chance to act as slave/learner group in next iterations so that the search agents can exchange their information in bidirectional over the groups.

2. The MS mechanism based algorithms generally uses the migration and/or regroup parameters based on that the search agents exchange their informa-

tion over the multi groups at an acceptable interval. However, these search process improves the exploration over the search group and the global search process, but it diminish the search speed of the algorithm. Whereas, in case of self-adaptive strategy, the best search agent has maximum probability to exchange its experience with the search agents in another groups. At the same time, this strategy maintains the diversification over the population and provides faster convergence rate in order to achieve the optimal solution.

3. The proposed strategy maintains the promising balance between the exploration and exploitation over the exchanging of the search experience. The search agents from the learner groups absorb the best experience from the neighbor learned group, and other search agents maintains the existing learning mechanism of GWO along with exploration mechanism. This strategy adjusts the search agents to some extent with in the search space, which guides to maintain the diversification. In addition, learner group search agents has the individual measurement to determine the rate of learning from the learned group in order to provide a chance to search agents to exploit its search location with the help of the learned group experiences. The self-adaptive strategy mainly focuses on exchanging the information in a bidirectional way between the search groups.

4. In addition to that, the swarm diversity has been addressed to examine the proposed search process. Moreover, this process guides the search groups to utilize the global search space without any restriction over the search agents in case of scaling problems.

Search agents are divided into j groups denoted as G (i.e. $G_1, G_2, ..., G_j$) in proposed work. Each group contains an alpha which acts as the first best search agent of the group, beta which act as the second best search agent of the group, delta determines the third best search agent of the group and neighborhood search agents which are selected randomly from the group and finally the global best which is best among all the groups. Let us consider, $X^t_{i,j}, X^t_{n1,j}, X^t_{n2,j}$ denoted as the current search agent position, first neighborhood search agent position and second neighborhood search agent position of the i^{th} search agent in j groups respectively.Likewise, the alpha search agent denoted as which is expressed as $\alpha^t_j = [\alpha^t_{1,j}, \alpha^t_{2,j}, ..., \alpha^t_{D,j}]$ for j groups over t iterations.

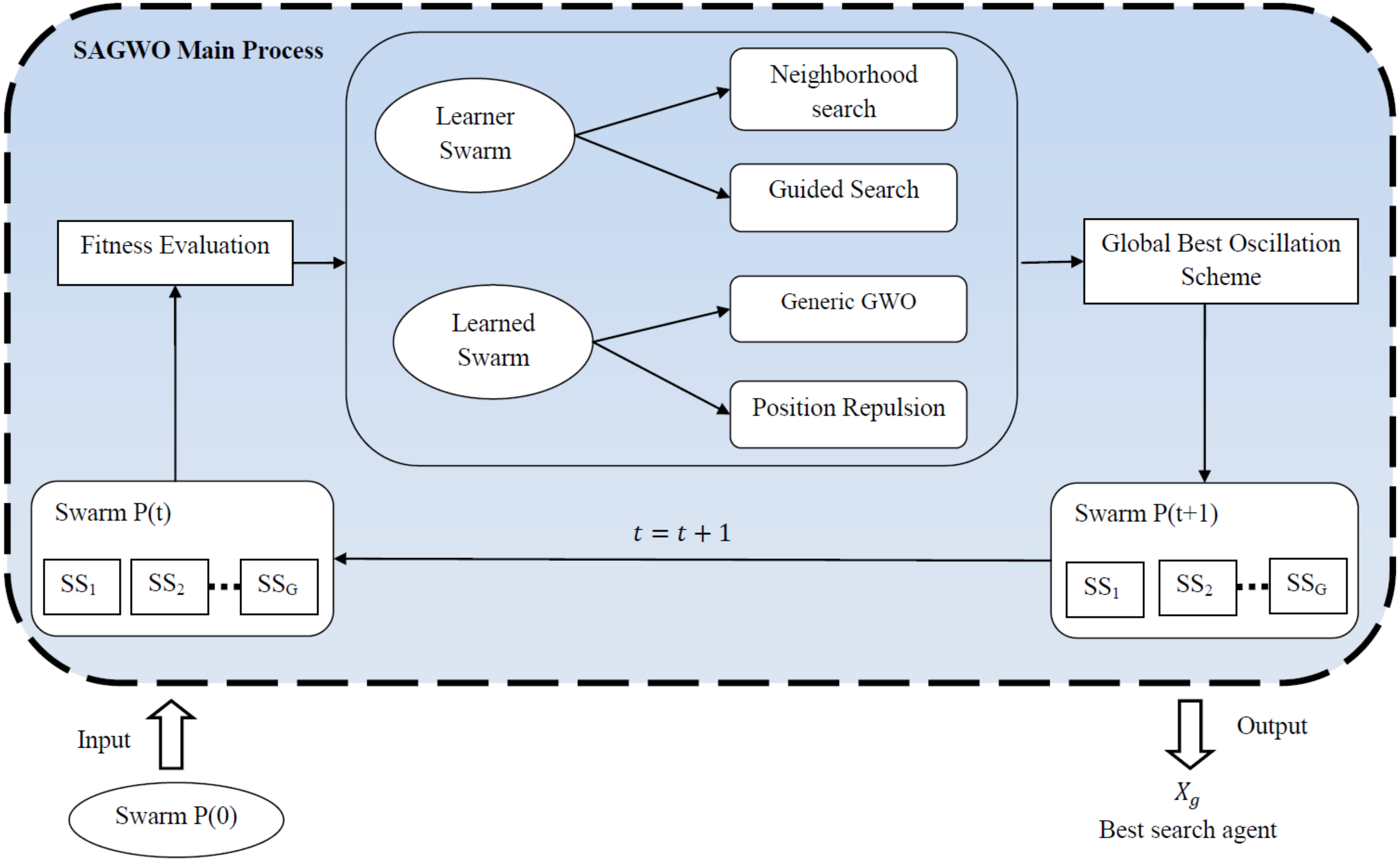

Figure 4.1: Main process of the proposed SAGWO

Figure 4.2: Group behavior of Grey Wolves

The global best search agent is denoted as X_g^t which is chosen from the overall j groups based on its fitness values of $f(\alpha_j^t)$ (i.e. $f(X_g^t) = \min[f(\alpha_j^t)]$). The approach of group behavior process in our proposed work is shown in figure 4.1.

The Overall Framework

Like the classical GWO, the proposed SAGWO initially starts with the random generation of search position for all search agents within the search boundary space and then it divides the search agents into j swarms. Then the fitness of all search agents in every j swarm is measured. The current search agents $X_{i,j}^t$, neighbor search agents $X_{n1,j}^t$ and $X_{n2,j}^t$, alpha search agents α_j^t and global best search agents X_g^t of all j swarms are determined. On the other hand, the role of each j swarm for all t iterations is determined. If the swarm act as the learner swarm, then the learning rate for all search agents in the learning swarm is measured.

With the help of individual learning rate and search agents from learned swarm, update the current search position of the search agents in the learner swarm. If the swarm act as the learned group perform the existing learning mechanism of GWO for some search agents and perform position repulsion mechanism for remaining search agents in the learned swarm. Then, compute the fitness for all search agents with its new position using the objective function. Later, update the search agents of $X_{i,j}^t, X_{n1,j}^t, X_{n2,j}^t, \alpha_j^t, X_g^t$ in all j swarms. Additionally, if the global best solution is not improved for k iteration then perform global best oscillation mechanism for global best solution alone. Repeat the above process, until the algorithm reaches maximum number of iteration or global best solution achieves the optimal solution. The generic flow of the proposed SAGWO algorithm is shown in algorithm 4.1 and the illustration of general framework is shown in figure 4.2.

Algorithm 4.1 Generic Flow of Proposed Work

Initialize the swarm $S(0)$ with N search agents and divide the swarm into j sub swarms ($j \in 1, 2, ..., G$) within the search boundary;
Compute the fitness of all search agent in j sub swarms;
Determine $X^t_{i,j}, X^t_{n1,j}, X^t_{n2,j}, \alpha^t_j, X^t_g$ of all j sub swarms;
while stopping condition(s) not true **do**
  Determine the role of each sub swarm;
  if Learner Group **then**
    Identify the learning rate of each search agent in the learner swarm;
    Update the position of the search agent in learner swarm;
  else
    Update the search agent of learned swarm according to the position repulsion and existing learning mechanism of GWO;
  end if
  Evaluate the fitness of all search agents in j sub swarms;
  Determine $X^t_{i,j}, X^t_{n1,j}, X^t_{n2,j}, \alpha^t_j, X^t_g$ of all j sub swarms;
  Process Global Best Oscillation mechanism for X^t_g if *globalbest* is not improved for k iterations;
end while
visualize the global best X^t_g;

4.3 Multi-Swarm Approach

In this work, the design of multi-swarm approach is made simple. The algorithm first randomly initializes the entire search agents in population and then it randomly partitions the swarm S into G sub swarms $S(x(1, ..., N)) = SS_1(x), ..., SS_G(x)$. The size of the swarm N is computed with respect to the dimensionality of the problem and its mathematical formulation is expressed in Eq. 4.1.

$$N = IS + |\frac{D}{10}| \tag{4.1}$$

where N is the determined swarm size of the proposed approach, IS represents the initial swarm size and D is the dimensionality of the problem. The small swarm size is quite enough for uni-modal problems whereas for high dimensional multi-modal problems the size of swarm is to be high in order to efficiently explore its search space.

Algorithm 4.2 Multi-swarm approach

Input: Determine the entire population $S(x_{1,...,N}) = x_1, x_2, ..., x_N$.
Output: A size of G sub swarms $SS_1, SS_2, ..., SS_G$.
Step 1: Identify the size of each sub swarm (NS):
$NS = N/G$
where NS denotes the number of search agents in each sub swarms, N is the total number of search agents in Swarm S and G represents the size of the group.
Step 2: For each subgroup $SS_{i,j}(i = 1, 2, ..., NS; j = 1, 2, ...G)$, assign the search agents;
Step 3: Terminate and visualize output;

The swarm size is dynamically varied based on the dimensionality of the problem. Then each sub swarm is arbitrarily assigned with set of search agents NS from the entire swarm S. This approach helps the algorithm to enhance the exploration among the population, more specifically for solving high dimensional multi-modal problems. This approach allows each subgroup to perform the interaction between the search agents. The algorithm of multi-swarm approach is given in algorithm 4.2.

4.4 Determining the Role of Sub Swarm

In this process, the sub swarm roles are determined iteratively either to perform as learner or learned one. For example, some set of sub swarms will be act as learner group and other group will act as the learned group. The role of each group changes iteratively for all t iterations based on the best solution (i.e. alpha solution) obtained in the sub swarm. The search agent in a learner group observes the experience from the learned neighbor sub swarm. In this determination, the sub swarm with best search agent is more likely to have chance to act as learned one. To determine the role, two sub swarms are initially selected from the G sub swarm. Then, the Boltzmann selection scheme is applied to compute the probability for selecting one as the learned sub swarm. Boltzmann selection scheme is also known as SoftMax method which is developed by [Abed-alguni and Alkhateeb 2017]. This scheme uses the global time varying parameter to select the appropriate sub swarm as the leader one. This scheme most probably select one as the learned sub swarm which attains the global best position so far. The probability PL is determined based on the following equation.

$$PL_m^t = \frac{e^{-f(\alpha_m^t)-f(X_g^t)/\tau}}{\sum_{m=1}^{2}\left(e^{-f(X_{\alpha,m}^t)-f(X_g^t)/\tau}\right)} \tag{4.2}$$

where, m represents the two sub swarms selected from the j sub swarms. τ denotes the temperature or global time adjustment parameter. If τ value is maximum then the probability of selecting learned group holds equal opportunity, otherwise the selection resides within the $f(\alpha_m^t)$ and $f(X_g^t)$. In order to choose the learned group, computed probability is compared with the random numbers between $[0, 1]$. This comparison based role determination guides to dynamically change the role of the group for every t iteration. The dynamic change over the role of the group helps iteratively to preserve the diversity.

4.5 Guided Neighborhood Search Mechanism

Guided Neighborhood search (GNS) mechanism is contributed to improve the exploitation process of the search algorithm. This mechanism is further classified into two categories namely 1. Neighborhood search mechanism and 2. Guided search mechanism. In case of neighborhood search mechanism, the search agents adjust its position based on the randomly selected search agent or by the search agent which is closer to the selected search agents. Whereas, in case of guided search mechanism, the search agents update its position using Limited memory BFGS (L-BFGS) mechanism.

The proposed guided neighborhood search mechanism works only for the learner group. This mechanism initially computes the learning rate of each search agent based on the rank. Later, the search agent with higher learning rate undergoes the guided search mechanism and all other remaining search agent with minimum learning rate adjust its position based on the neighbor learned group search experience. GNS strategy is quite effective in case of obtaining global optima whenever the local optimum is nearer to it. The GNS is capable to generate the best position for the search agent based on its current search region. This mechanism is quite helpful in exploiting the search environment in frequent iterations.

4.5.1 Measuring Learning Rate Probability

The idea of measuring the learning probability is inspired from the natural social learning. This learning probability is used to compute the rate of learning based on the ranking assignment of the search agent to adjust the search position from the learned group. Based on the nondeterministic method, ranking assignment is processed in which the rank for each search agent is assigned using its current fitness value. In particular, the learner group is rearranged as worst to best (i.e. as like descending order) based on the fitness value of the search agent, and the ranking of the search agent is expressed in eq. 4.3.

$$R_i = NS - I_i + 1 \tag{4.3}$$

where, R_i represents the ranking of i^{th} search agent, NS denotes the number of search agents in the learner group and I_i is the index value of the i^{th} search agent after an arrangement. In specific, the current best search agent will attain the highest ranking (i.e. $X_{best} = NS$)and the current worst search agent obtain the lowest ranking (i.e. $X_{worst} = 1$).

With the help of ranking assignment, the learning probability is measured for all search agents. For the measurement, parameter PM is used which holds different probability measurement value for all search agents in the learning group. Moreover, the probability value is determined as much small as to provide a chance to exploit its current search state. The learning probability measure PM fluctuates in the range of $[0.1, 1]$ based on the empirical simulations and thus the formulated expression is given in Eq. 4.4.

$$PM_i = (1 - \frac{R_i - 1}{NS})^{\phi \log(\frac{D}{S})} \tag{4.4}$$

where NS is a number of search agents in learning swarm, D represents the dimensionality of the problem and S denotes the size of the swarm and ϕ is the positive exponents. More likely, the search agents with best fitness has less probability of learning rate and the search agents with worst fitness has high probability of learning rate. Since, the best search agents have a chance to sustain its current state with small adjustments and also helps to exploit the current search space. In other words, the fluctuation on the parameter ϕ provides different learning rate which degrades the performance. Based on the empirical results ϕ is fixed as 0.5

which helps to improve the learning rate for n-dimensional problems.

The figure 4.3 illustrates the relationship between the probability of learning PM_i with swarm size S and the dimensionality D. The curves in the figures shows the relationship between the learning probability and dimensionality of problem which fluctuates between $D \leq 100 to D = 2000$. For small dimension ($D \leq 100$), the learning probability is same for all search agents (i.e. $PM_i = 1$). At the same time, for large dimension the learning probability linearly decreases with respect to increase in fitness values or dimensions (D).

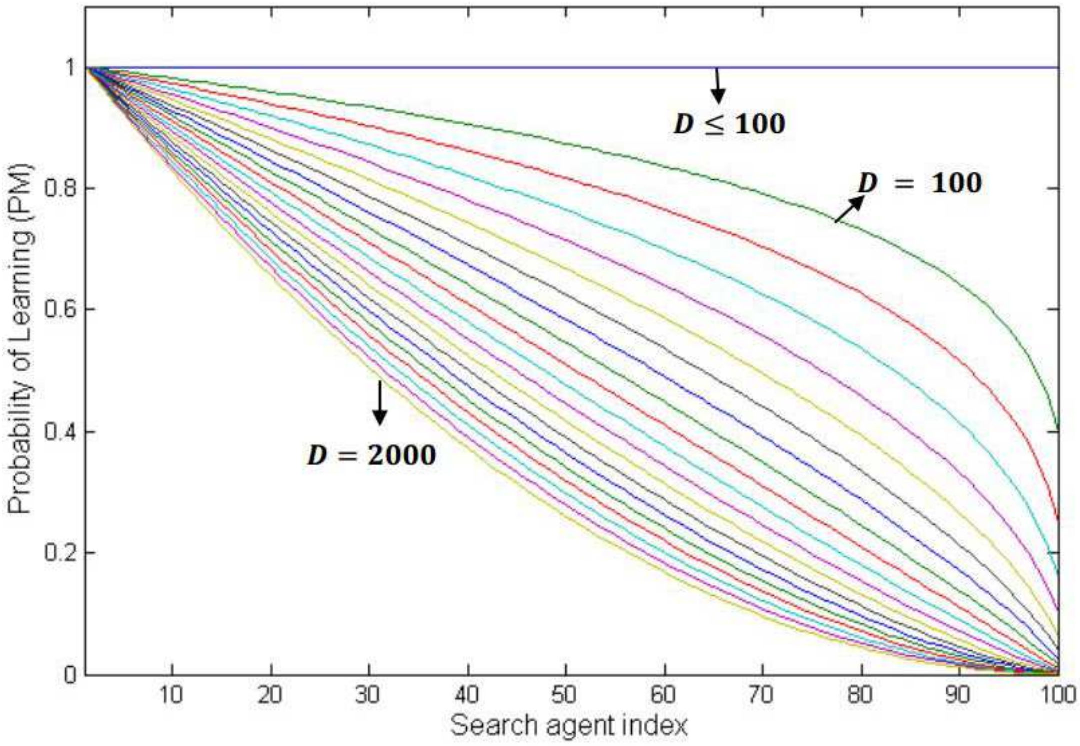

Figure 4.3: Probability of learning curves for varying dimensionality from $D100$ to $D = 2000$.

4.5.2 Position Update using Neighborhood Search

The search agents in learner swarm updates its search direction based on the two modes. Firstly, some set of search agents update its position based on the experience of the neighbor learned group (i.e., based on the three search agents X^t_{n1d}, X^t_{n2d} and α^t_l from the learned group).In existing GWO algorithm, the search agents adjust its position based on the three best search agents whereas this updating process stops converging in a course of iterations due to similar position of the three search agents.

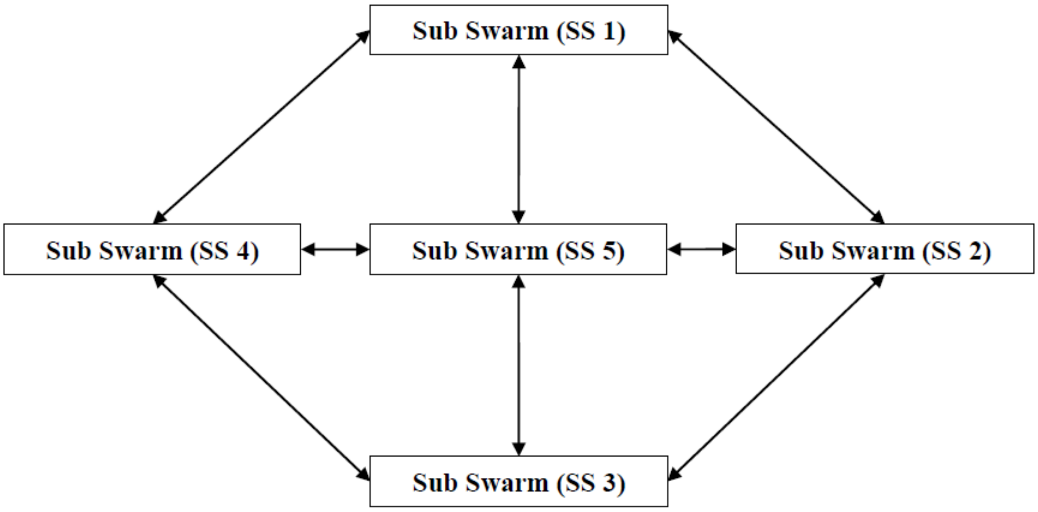

Figure 4.4: Example of Information Exchange between the five sub swarms with complete free form of neighborhood topology.

The figure 4.4 illustrates that any sub swarm which acts as the learned one will exchange their information to their neighbor learning sub swarm. Here, five sub swarms are considered namely sub swarm (SS1), sub swarm (SS2), sub swarm (SS3), sub swarm (SS4) and sub swarm (SS5) in which sub swarm (SS1) act as the learner then it can exchange the learning experience from any of the other sub swarm (SS2 - SS5). Likewise, the information exchange dynamically happens based on the changes on the learned sub swarm. In the proposed neighborhood based search mechanism, two search agents are arbitrarily selected from the learned group for information exchange between the groups namely X_{n1d}^{t} and X_{n2d}^{t}.Along with that the learned group best search agent, α_l^t is utilized to adjust the search agent position. The neighborhood based search mechanism is formulated and expressed in eq. 4.5.

$$X_{id,j}^{t+1} = 0.33 \times (A'_1 r_1 (X_{n1d}^{t} - Xid, j^t) + A'_2 r_2 (X_{n2d}^{t} - Xid, j^t) + PM_i r_3 (X_{\alpha,l}^{t} - X_{id,j}^{t})) \tag{4.5}$$

where A'_1, A'_2 are the coefficient parameters used to observe the experience from the neighbors (i.e. $X_{n1d}^{t} and X_{n2d}^{t}$) of learned group. The values of coefficient parameter are different from the generic GWO values which aids to achieve better exploitation as well as for the faster convergence rate. Another parameter PM_i is used to control the magnitude of learning from the learned group.

4.5.3 Position Update using Guided Search Mechanism

In this section, in order to boost the search agents we have chosen the L-BFGS technique for guided search process. Broadly speaking, limited memory BFGS (L-BFGS) mechanism is an extension of BFGS technique. Let us consider x_0 be the initial point and the next point is iteratively determined based on the line search technique using the eq. 4.6.

$$x_{k+1} = x_k + \lambda_k d_k \tag{4.6}$$

where λ_k represents the step length and d_k denotes the descent direction. Most of the researchers addressed different techniques to estimate the step length λ_k. Initially, [Liu and Nocedal 1989] introduced a mechanism to obtain λ_k which is given in eq. 4.7.

$$\phi(x_k + \lambda_k d_k) - \phi(x_k) \leq \sigma \lambda_k \nabla g(x_k)^T d_k \tag{4.7}$$

where σ is an arbitrary value (i.e. $\sigma \in (0,1)$) and $\nabla g(x_k)$ denotes the Jacobian matrix. With the help of above formula, Zhu [ ] introduced the monotone line search mechanism as given in eq. 4.8.

$$\phi(x_k + \lambda_k d_k) - \phi(x_{lk}) \leq \sigma \lambda_k \nabla g(x_k)^T d_k \tag{4.8}$$

$\phi(x_{lk}) = max_(0 \leq j \leq m(k))\phi(x_{k-j}), m(0) = 0$,
$m(k) = \min m(k+1) + 1, M(fork \geq 1)$, and M is a positive integer. Based on the above formulas, it is quite simple to compute $\nabla g(x_k)$, but it might increase the workload for large-scale problems. In order to eradicate the issue, we presented the modern backtracking inaccurate mechanism that is expressed in eq. 4.9.

$$\|g(x_k + \lambda_k d_k)\|^2 \leq \|g(x_k)\|^2 + \psi \lambda_k^2 g_k^T d_k \tag{4.9}$$

where $\psi \in (0,1)$, g is symmetric for all x_k (i.e. $g_k = g(x_k)$). The numerical performance of this modified line search mechanism provides more accurate than the standard line search mechanism. Li and Li 2011 developed an exact monotone line search mechanism which is given in eq. 4.10.

$$\phi(x_k + \lambda_k d_k) - \phi(x_{lk}) \leq -\psi_1 \|\lambda_k d_k\|^2 - \psi_2 \|\lambda_k g_k\|^2 + \Psi_k \|g(x_k)\|^2 \tag{4.10}$$

where ψ_1 and ψ_2 are nonnegative constants, $\lambda_k = r^{i_k}$, $r \in (0, 1)$, i_k denotes the minimum positive integer i and Ψ_k determined as eq. 4.11.

$$\sum_{k=0}^{\infty} Psi_k = \infty \tag{4.11}$$

The line search technique is included with BFGS update equation and obtained efficient results. Based on this modification we presented some mechanisms for d_k. Initially, Newton method is a most effective method which generally requires a smallest number of function estimations and it's efficiency in handling unconstrained problems. But, its efficacy mainly depends on the assumption to handle the linear system that occurs when iteratively calculating d_k.

$$\nabla g(x_k) d_k = -g(x_k)$$

Additionally, the approximate solution of linear system might be too oppressive, or it is not required when x_k is far away from a solution. Inaccurate Newton techniques perform the fundamental method for Newton-type large-scale algorithms. The present estimation of the solution is refurbished by relatively solving the linear system. From the class of Quasi-Newton methods, BFGS is the most effective method in which search direction d_k is determined in eq. 4.12.

$$B_k d_k + g_k = 0 \tag{4.12}$$

where $g_k = g(x_k)$ and B_k is manipulated by modified BFGS equation 4.13.

$$B_{k+1} = B_k - \frac{B_k S_k S_k T B_k}{S_k^T B_k S_k} + \frac{y_k y_k^T}{y_k^T S_k} \tag{4.13}$$

where $s_k = x_{k+1} - x_k$ and $y_k = g_{k+1} - g_k$. Let the inverse of B_k is denoted as H_k, then the inverse update equation is determined as eq. 4.14.

$$\begin{aligned} H_{k+1} &= H_k - \frac{y_k^T(S_k - H_k y_k)S_k S_k^T}{(y_k^T S_k)^2} + \frac{(S_k - H_k y_k)S_k^T + S_k(S_k - H_k y_k)^T}{(y_k^T S_k)2} \\ &= (1 - \frac{S_k y_k^T}{y_k^T S_k} H_k (1 - \frac{y_k S_k^T}{y_k^T S_k}) + \frac{S_k S_k^T}{S_k y_k^T} \end{aligned} \tag{4.14}$$

This is the twofold representation of DFP update equation in the logic that $H_k \leftrightarrow B_k, H_(k+1) \leftrightarrow B_(k+1)$ and $s_k \leftrightarrow y_k$. Moreover, Limited memory BFGS (L-BFGS) technique is an extension of the BFGS technique for solving large-scale problems. The L-BFGS is almost similar to BFGS, the only difference is that the inverse Hessian estimation is not generated easily, whereas this may be estimated by small number of BFGS updates. This mechanism speedup the convergence rate and consumes minimum storage for computation. In the LBFGS technique, H_k matrix is determined by updating the standard matrix $H\tilde{m}(>0)$ times with the help of BFGS equation with the earlier $\tilde{m}$ iterations. The rudimentary BFGS improvement has the following eq. 4.15.

$$H_{k+1} = V_k^T H_k V_k + \epsilon_k S_k S_k^T \tag{4.15}$$

where $\epsilon_k = \frac{1}{s_k^T y_k}, V_k = 1 - \epsilon_k y_k S_k^T$, and I is the element matrix. H_{k+1} in the L-BFGS technique has the following eq. 4.16.

$$H_{k+1} = V_k^T H_k V_k + \epsilon_k S_k S_k^T$$

$$\begin{aligned} &= V_k^T [V_{k-1}^T H_{k-1} V_{k-1} + \epsilon_{k-1} S_{k-1} S_{k-1}^T] V_k + \epsilon_k S_k S_k^T \\ &= ... \\ &= [V_k^T, ..., V_{k-\tilde{m}+1}] H_{k-\tilde{m}+1} [V_{k-\tilde{m}+1}, ..., V_k] + \epsilon_{k-\tilde{m}+1} \\ &[V_{k-1}^T, ..., V_{k-\tilde{m}+2}^T] S_{k-\tilde{m}+1} S_{k-\tilde{m}+1}^T [V_{k-\tilde{m}+2}, ..., V_{k-1}] +, ..., + \epsilon_k S_k S_k^T \end{aligned} \tag{4.16}$$

Some modifications on L-BFGS technique has been addressed with the help of line search techniques to guide the search agents to solve the large-scale optimization problems. The algorithm for L-BFGS technique is given as in algorithm 4.3.

In this algorithm, we considered the inverse of H_k as B_k and also considered the fundamental matrix B_0 and its inverse H_0 are positive infinite.

In case of guided search mechanism, some search agents updates its search position based on this L-BFGS mechanism. This mechanism helps the solution to exploit their current search positions which are far away from the global best position.

Algorithm 4.3 L-BFGS

Step 1: Select the initial point x_0,an initial symmetric positive matrix $H_0 \in \Re^{n\times n}$, non-negative constants ψ_1, ψ_2 and constants r,ψ, $\epsilon \in (0,1)$, a non-negative integer m_1 and Ψ_k, let $k=0$.
Step 2: Stop if *condition satisfies*;
Step 3: Determine search direction d_k by $B_k d_k = -g_k$,
Step 4: If $\|g(x_k+d_k)\| \leq \epsilon\|g(x_k)\|$, then pick $\lambda_k = 1$ and move to Step 6, otherwise move to step 6.
Step 5: Let i_k be the minimum positive integer i such that line search 4.6 holds for $\lambda = r^i$. Let $\lambda_k = r^{ik}$.
Step 6: Let the next iterative be $x_{k+1} = x_k + \lambda_k d_k$.
Step 7: Let $\tilde{m} = mink + 1, m_1$. Set $S_k = x_{k+1} - x_k = \lambda_k d_k$ and $y_k = g_{k+1} - g_k$. Update B_0 for $l = k - \tilde{m} + 1, ..., k$ measure
$B_k^{l+1} = B_k^l - \frac{B_k^l S_l S_l^T B_k^l}{S_l^T B_k^l S_l} + \frac{y_l y_l^T}{y_l^T S_l}$
where $S_l = x_{l+1} - x_l, y_l = g_{l+1} - g_l$ and $B_k^{k-\tilde{m}+1} = B_0$ for all k.
Step 8: Set k:k+1. Go to Step 2.

In addition to that, L-BFGS mechanism provides more accurate and effective self-correcting functionalities in adjusting the search agent position. The position update for search agents is expressed as eq. 4.17.

$$X_{id,j}^{t+1} = (\frac{1}{2}PM_i(\alpha_l^t - X_{id,j}^t)) + \rho L - BFGS(X_{id,j}^t) \tag{4.17}$$

where ρ represents the constant parameter and its used to adjust the current search position of the search agents.

Generic GWO is quite complicated to identify the best-fit search agent in the local neighborhood. To eradicate this issue, L-BFGS is addressed to guide the search agent to exploit its local environment in order to identify the best position more quickly. When one search agent is assigned into the guided search, the local optimization technique L-BFGS is performed by setting this search agent as the starting point, from the initial position the technique converge towards the best fit position.

In addition to that, the guided probability parameter P_g is used to select the search agent either to perform the neighborhood based search mechanism or guided search mechanism. In our work, we suggest $P_g = 0.8$ that is one fourth of the learning search agents perform the guided search and remaining search agents process the neighborhood search mechanism. More specifically, the search agent with high learning rate has maximum probability to process the guided search mechanism.

This parameter is compared with the random value (i.e. $rand[0,1]$) and this provides dynamic changes on the selection of search agent and provides a better chance for efficient exploitation.

4.6 Summary

In this chapter, self adaptive strategies based GWO algorithm has been proposed to enhance the learning rate between the search agents. Likewise, the features of the self adaptive strategies are discussed. The proposed model consists of three strategies namely, Guided Neighborhood Search, Position Repulsion Mechanism and Global Oscillation Scheme. This chapter described the characteristic features of Guided Neighborhood Search Mechanism with its effect on the adjustment of search agent position in order to enrich the exploitation. Moreover, Multi-swarm based approach is also presented.

Chapter 5

Proposed Methodology-II

5.1 Introduction

The multi swarm approach and guided neighborhood strategies are discussed in the chapter 4. In this section, the position repulsion mechanism and global best oscillation scheme are discussed with its influence on adjusting the search agent in order to obtain the optimal solution. Furthermore, the swarm diversity is introduced to identify the influence of the neighborhood control which helps to maintain a trade-off between the intensification and diversification. Later, the entire structure of the proposed model with its flowchart is discussed in detail.

5.2 Position Repulsion Mechanism

In traditional GWO algorithm, the omega wolves are guided by the best search agents to adjust its position. The position of the search agent in every iteration determined as attractor which helps to speed up the search process but it degrades the algorithm by stagnating in local saddle point especially for large scale multi-modal problems. Furthermore, Guided Neighborhood Strategies (GNS) which are discussed in chapter 4 provide an efficient search over the multi-swarm. In some cases the search agent in learner swarm has the capability to explore the search space. However, in case of complex multi-modal problems the GNS intend to provide poor exploration over the search agents. To handle this issue, position repulsion mechanism with the help of Lévy flight technique is addressed. This repulsion mechanism has been introduced for the learner swarm in which the search agents are utilizing the global search space and enrich the search process. In addition to that, global search is occasionally motivated by longer walking distance, which provides the algorithm to move out of the saddle point and enhance the global search ability.

(a)

(b)

(c)

Figure 5.1: Grey wolf with different movements (a) Grey wolf stops moving or stagnate in a position, (b) Grey wolf with slow movement, (c) Grey wolf with fast movement

Figure 5.1 (a) illustrates that the grey wolf stops moving in a certain number of iteration due to struck in local optima or fails to explore the search space. In order to eradicate, Lévy flight with varying step size has been processed which adjust the search agent with two different movements namely search agent with slow and fast movement as shown in figure 5.1 (b) and (c). In case of slow movement the search agent will utilize the current search space and then it starts to explore the search space slowly whereas in fast movement the search agent jumps out of the saddle point, which in turn the global optima with its current position is not determined. So, the step size with arbitrary movement is to be determined by handling both the slow and fast movement.

Lévy distribution is generally expressed as eq. 5.1.

$$L\acute{e}vy(\beta) \sim s^{-1-\omega}, (0 < \omega \leq 2) \tag{5.1}$$

Generally speaking, Lévy flight is an arbitrary walk technique in which the step size follows the Lévy distributions, and walk direction follows the uniform distribution. The Lévy distribution is performed with the help of Mantegna law to obtain the step size [Haklı and Uğuz 2014; Tang et al. 2016]. The step size based on Mantegna law is measured as eq. 5.2.

$$step = \frac{u}{|v|^{\frac{1}{\omega}}} \tag{5.2}$$

where u and v denotes the positive integer which adopts the normal distribution, i.e. stated in eq. 5.3.

$$u \sim N(0, \sigma_u^2), v \sim N(0, \sigma_v^2)\sigma_v = \frac{\Gamma(1+\omega) \times \sin(\frac{\pi\omega}{2})}{\Gamma[\frac{(1+\omega)}{2}] \times \omega \times 2^{\frac{(\omega-1)}{2}}}^{\frac{1}{\omega}} \tag{5.3}$$

where Γ represent standard Gamma function which aids to update the position in a steady state and ω denotes the positive integer (i.e. $\omega \in [0.3, 1.99]$).

$$L\acute{e}vyFlight = rand(0,1) \times normal(0,1) \times step \tag{5.4}$$

Where $rand(0,1)$ denotes the random value of uniform distribution, $normal(0,1)$ represents the arbitrary umber of normal distribution. The random movement with step size value ($\alpha = 0.5$) is shown in figure 5.2. The equation 5.1 to 5.4

expresses the Lévy flight mechanism which generates the small integer values for random walk movement around the boundary space. This random walk movement aids to explore the search space of the particles within the search region.

The proposed position repulsion operation is expressed as eq. 5.5.

$$X_i^{t+1} = X_i^t + rand \times (LévyFlight \bigotimes (X_g^t - X_i^t)) \tag{5.5}$$

where X_i^t represents the current position of the search agent i, X_g^t is the global best search agent in current iteration t and $rand$ denotes the arbitrary value between $[0, 1]$, $\bigotimes$ determines an entry wise multiplications.

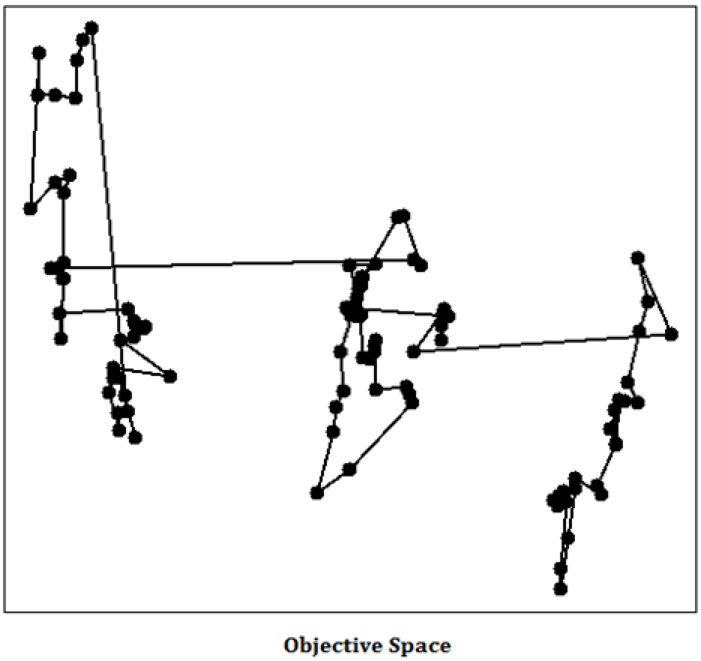

Figure 5.2: Lévy flight movements with step size ($\alpha = 0.5$)

In addition to that, the novel selection scheme is addressed to select the search agents to perform the position repulsion using Lévy mechanism. This selection scheme adopts the position repulsion operator for the solution which is not improved in the current iteration by comparing the previous iterations. The selection mechanism is given in Eq. 5.6.

$$pS_i = exp(-\frac{(f(X_{id,j}^{t-1}) - f(X_{id,j}^t))}{\sum_{j=1}^{m} f(X_{id,j}^t)}) \tag{5.6}$$

5.3 Global Best Oscillation Scheme

The global best oscillation scheme (GBOS) is a small adjustment mechanism especially for global best solution. This scheme works only when the global best solution is not improved for certain number of iterations. In SAGWO, the global best search agent is identified from the set of sub swarms and it is denoted by X_g^t. Iteratively, the global solution is updated in case of current best is better than the previous best. In some cases, the proposed algorithm regrets the chance to explore the global optima for complex multi-modal problems. In order to eradicate the issue, GBOS helps the global best search agent to jump out from the local optima struck and adjust the position vector in a better region. This GBOS scheme works on single dimension of X_g to conserve the current good structure. The single dimension perturbation provides more chances to generate a new better region. The notation O_d is used to represent the d^{th} dimension of X_g. GBOS is computed based on the Gaussian distribution [Krohling and Mendel 2009] and given as in Eq. 5.7.

$$O_d = X_{gd}^t + (Ub - Lb) \times Gaussian(X_{gd}^t) \tag{5.7}$$

where Ub and Lb represents the upper and lower dimension value, Gaussian (X_{gd}^t) is a random number of a Gaussian distribution with average and a standard deviation of global best X_{gd}^t. This strategy processes the extensive oscillation to explore the search space where as little oscillation aids to explore the current search agent position. Finally, the global best search agent will be replaced if and only if new global search agent fitness is better than X_{gd}^t. The generic flow chart of global best oscillation scheme is given in figure 5.3.

Finally, projection operator is used to adjust the position of the search agents which moves out of the search range $[Ub, Lb]$. The formulation of projection operator is given in eq. 5.8.

$$P \in (O_d, Ub, Lb) = \begin{cases} \min Ub, 2Ub - O_d & if O_d < Lb \\ \max Lb, 2Ub - O_d & if O_d > Ub \end{cases} \tag{5.8}$$

where UB and LB determines the upper and lower boundary spaces, O_d denotes the newly adjusted position.

Algorithm 5.1 Global Best Oscillation Scheme

Procedure GBOS
Randomly select decision vector d from the global best X_{gd}^t
Oscillate the selected global best decision vector
Check the newly adjusted position is out of boundary search space
if $O_d \notin [Lb, Ub]$ **then**
  Apply Projection operator to roll back within the boundary space
end if
Evaluate the fitness $f(O_d)$
if $f(O_d) < f(X_{gd}^t)$ **then**
  $X_{gd}^t = O_d$
end if
Return X_{gd}^t;

5.4 Swarm Diversity

In most of the swarm based approach contributes only on maintaining the high diversity among the swarm which in result alleviates premature convergence but higher degree of diversity may slow down the convergence speed. So, a balanced process has been measured which provides both the diversity among the swarm as well as the speed up the convergence rate. The Swarm diversity is used to examine the effect of the neighborhood control as well as used to balance the search process. For example, the higher exploitation on the learner sub swarm will result on local optima stagnation. This diversification measure helps to detect the effect of the neighborhood and chance of stagnation in local optima. If the effect of neighborhood is high then it induces the learned swarm to explore more [Mohapatra et al. 2017; Cheng and Jin 2015].

The Diversity measure of the groups measured during every iterative search process is given in eq. 5.9.

$$DM(G) = \frac{1}{m}\sum_{i=1}^{m}\sqrt{\sum_{j=1}^{n}(X_{ij} - \hat{X}_{ij})^2 with \hat{X}_{ij}} = \frac{1}{m}\sum_{i=1}^{m}(X_{ij}) \tag{5.9}$$

where DM(G) denotes the diversity measure of the group G, m is the group size, n is the dimension of the decision space, X_{ij} is the position of the search agent i of j dimension, and $\hat{X}_{ij}$ is the average value of the overall search agents of j dimensions.

Algorithm 5.2 Self Adaptive Grey Wolf Optimization Algorithm

Identify the number of search agents and size of the sub swarm using multi-swarm approach;
Initialize the positions of all N search agents $X_i (i = 1, 2, , n)$ in G swarms and assign the NS search agents to all G groups $(X_i \in (Ub, Lb))$.
while Stop Criterion is not obtained **do**
  Evaluate the fitness value for all search agents in G swarms
  Update α_j^t, X_g^t
  if $rand \le PL_m^t$ **then**
    //Perform learning process for learner sub swarm
    Identify learning rate (PM_i)
    if $rand < P_g$ **then**
      Update the position using neighborhood learned experiences
    else
      Update the position using guided search mechanism
    end if
  else
    //Perform Search mechanism for learned sub swarm
    Compute the swarm diversity for learned group
    Process the selection process to perform Position Repulsion operation
    if $rand < pS_i$ **then**
      Perform the Position Repulsion Operation
    else
      Perform the Operation of Generic GWO
    end if
    if $X_{i,j}^{t+1}$exceeds the boundary space **then**
      Project within the search space
    end if
  end if
  Global Best Oscillation Scheme
end while
Return X_g

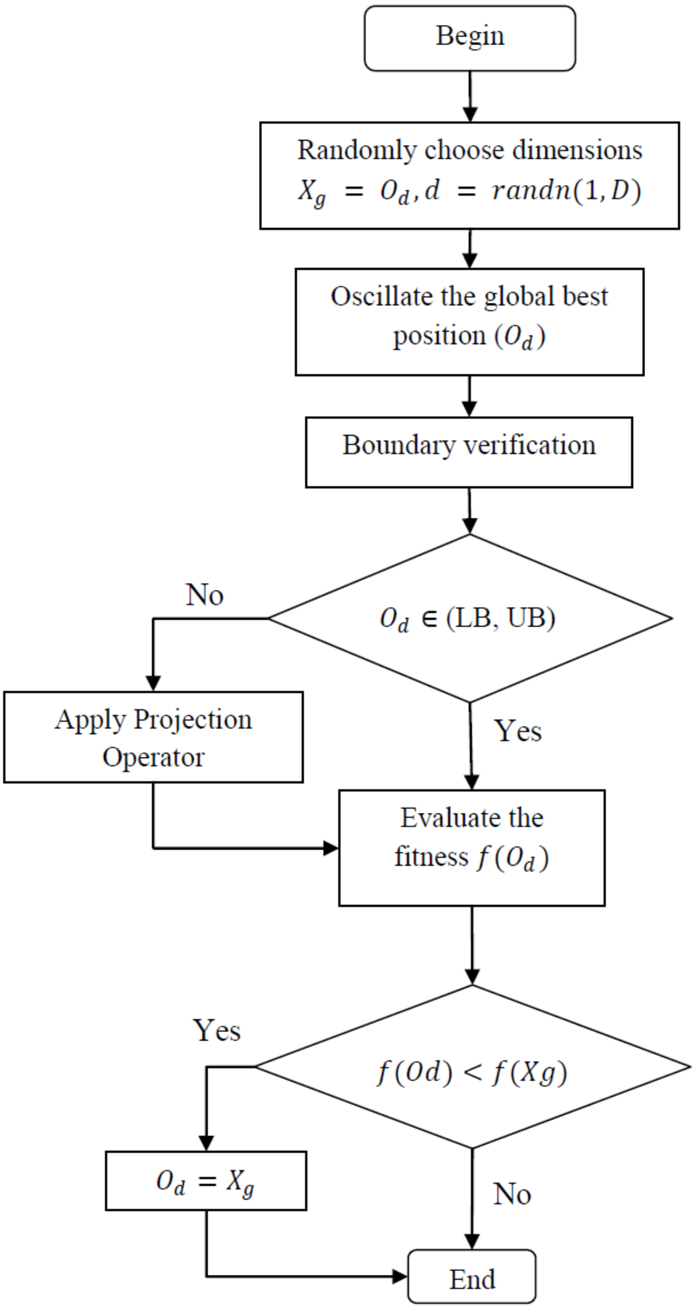

Figure 5.3: Flowchart of Global best Oscillation Scheme

5.5 Description of SAGWO Algorithm

The proposed SAGWO algorithm 5.2 starts with the multi-swarm approach which determines the number of search agents based on the dimensionality of the problem. Then, it initialize the search agents randomly within the upper and lower boundary search space (Ub, Lb) and divide the swarm $S(0)$ into G groups.

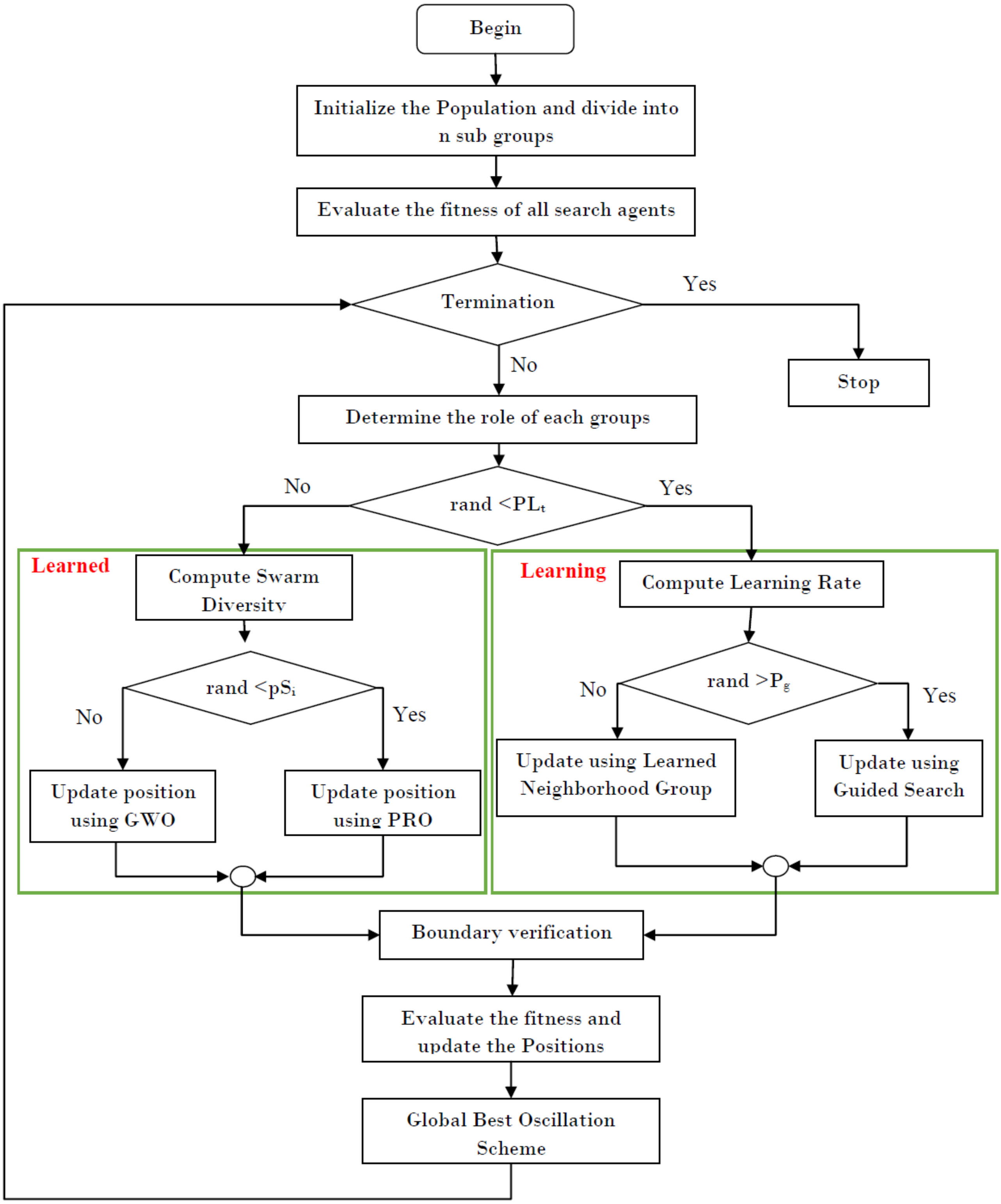

Figure 5.4: Flowchart of Global best Oscillation Scheme

Assign, SS search agents to all G sub swarms and evaluate the fitness of all search agents. Identify the alpha for all j sub swarms α_j^t and overall global best X_g^t.

The role of each sub swarm PL_m^t is determined and if the PL_m^t is greater than or equal to the rand then the sub swarm will act as the learner one otherwise the sub swarm will be act as the learned one. Identify the learning rate (PM_i) for all the search agents in learner sub swarm and using the guided probability perform either the neighborhood learned experience or guided search mechanism. When arbitrary value is less than the guided probability then the search agent will perform the neighborhood learned experiences otherwise it performs the guided search experience.

Then, compute the swarm diversity for the learned groups if the diversity of the sub swarm is less than the exploration of the search agents will be more. Then identify the selection probability, if the selection probability is greater than the random value then perform the position repulsion operation otherwise update the search positions using generic GWO. Using projection operation, project the search agents which are out of the boundary search space. Finally, perform the global best oscillation scheme iff the global best is not improved for certain number iterations. Repeat the process until the stop criteria of the algorithm reaches. The generic flowchart of the proposed SAGWO algorithm is shown in figure 5.4.

The proposed work enriches the capability of both exploitation and exploration using the Neighborhood search mechanism and Position Repulsion operator. In addition to that, diversity measure used to evaluate the effect the neighborhood control as well as used to balance the search process. For example, the higher exploitation on the learner swarm aids to exploit its local search space but within the certain number of iteration it struck into local optima without any sort of update in the search position. At the same time, higher exploration allows the search agents to move around the global search space but fails to exploits local search space.

In order to determine the steady state, diversification measure aids to detect the effect of the neighborhood and chance of stagnation in local optima. If the effect of neighborhood is high then it induces the learned swarm to explore the search space. Without loss of generality, this steady state process helps the proposed work to balance the exploration and exploitation among the search agents towards the optimal solution.

5.6 Summary

In this chapter, self adaptive strategies namely guided neighborhood search, position repulsion mechanism and global best oscillation scheme has been proposed to enhance the grey wolf optimization. These three mechanisms mutually work to accelerate the search and guide the algorithm to obtain the global optimal or near optimal solution in significant time. In addition to that, multi-swarm approach has been introduced to handle the high dimensional problems with varying population sizes with its dimensionality. Furthermore, swarm diversity has been used to improve the diversity among the swarms based on the influence of the neighborhood control and the formulation of each schemes with its adaptive parameters.

Chapter 6

SAGWO for Large Scale Benchmark Functions

6.1 Introduction

In the chapter 4 and 5, a self adaptive strategy has been proposed to enhance the grey wolf optimization algorithm in order to handle the high dimensional complex optimization problems. The proposed algorithm has to be evaluated and compared with respect to other state-of-art of meta-heuristic algorithms using suitable performance measures. In this perspective, an appropriate experimental setup has been formulated and experiments are carried out on different unconstrained large scale benchmark functions. In addition to that, real-world complex optimization problems such as Economic Load Dispatch (ELD) problem and Localization problem is utilised in order to validate the performance of the proposed work.

6.2 Experimental Setup

6.2.1 Test-Bed Design

The performance of the proposed model has been processed by large scale benchmark functions and the same have been demonstrated on real world problems. The test bed design have been classified into three phases namely first is on large scale benchmark function, then economic load dispatch problem and finally localization problem. Experiments carried out on proposed model under different phases to evaluate the performance in the most precision way.

The parameters of SAGWO and the corresponding values are shown in Table 6.1. Guided probability is used to exploit the search agent in order to locate the possible best position within its current environment. Lévy step size is to explore the search agent to avoid the stagnation in local optima or saddle point. The reason behind the setting of Lévy flight step size is to project the random walk over the search agent to identify the best position in a large complex search space.

Table 6.1: SAGWO Configuration Parameter

S.No	Parameter	Value
1	Initial Population Size	100
2	Swarm Size (s)	4
3	Generation Limit	200
4	Initialization	Random
5	Guided Probability	0.3
6	Lévy Step Size	0.55
7	Constant Parameter (ρ)	0.5

6.3 Large Scale Benchmark Functions

In this section, the experimentation have been carried out on the large scale benchmark functions in order to evaluate the performance of the proposed work as well as with other state-of-art meta-heuristics algorithm. This experimentation helps to judge the proposed SAGWO algorithm on different characteristics of problems viz. separability, non-separability, unimodal and multimodal. These benchmark functions are collected from the special issue of soft computing [Herrera et al. 2010] in order to evaluate the scalability of the algorithm.

The proposed work has been implemented on MATLAB 2014a. The algorithm has been performed over 25 independent runs for each test functions. The dimension of each test function varies from $50, 100, 200, 500$ and 1000 continuous real values within a desired range. The performance metrics observed based on the best, worst, median, mean and standard deviation of each function. The maximum number of fitness evaluation have been fixed as $5000 \times D$ (i.e. D represents the dimensions), which will act as the stop criteria of each run. The benchmark functions ($f1 - f19$) have been used and those details are stated in Table 6.2-6.5.

6.3.1 Benchmark functions

Optimization task of the proposed work is to maximize (or) minimize the objectives by following the environmental criteria. In this research all the test benchmark functions are minimization problems. The performance of the proposed work can be easily evaluated by these set of benchmark functions. The benchmark functions which have been considered are 19 functions with different properties viz. uni-modal, multi-modal, separable (i.e. variable independent), non-separable (i.e. variable dependent) and hybrid composed (i.e. variables with both dependent and independent). The test functions F1 to F6 were observed for CEC 2008 [Tang et al. 2007] and functions f7 to F11 were observed for IDSA 2009 [Lozano et al. 2011] and functions F12 to F19 have been created specifically for this special issue [Herrera et al. 2010]. The lists of benchmark functions with its special characteristics are given in Table 6.2-6.5.

Uni-modal functions

In this test case, seven uni-modal functions (F1-F2, F7-F11) are considered, each uni-modal has either same search range or different search range. This helps to analyze the optimization algorithms by determining the global maximum or global minimum without any form of local optima. This function is mathematically expressed and given in Table 6.2 and its properties are also provided in Table 6.4.

Multi-modal functions

In order to evaluate the performance of the algorithm four multi-modal functions ($F3 - F6$) are considered. This function has complex search space with many local optima. At the same the local optima increases with respect to increase in dimensionality of the problem. Multi-modal function is used to evaluate the capability of optimization algorithm by handling complex search of a problem.

Separable functions

The functions $F1, F6, F7$ and $F8$ are the separable problems in which the decision vectors can be divided into partitions for optimization purpose. In these function, $F1, F6$ and $F7$ are easily optimized dimension by dimension whereas function $F8$ cannot be optimized dimension by dimension.

Table 6.2: Benchmark functions F1-F7

Functions	Name	Description
F1	Shifted Sphere function	$\sum_{i=1}^{D} Z_i^2 + f_{bias}, z = x - o$
F2	Shifted Schwefel Problem 2.21	$max_i \|z_i\|, 1 \leq i \leq D + f_{bias}, z = x - o$
F3	Shifted Rosenbrock's function	$\sum_{i=1}^{D}(100(z_i^2 + z_{i+1})^2 + (z_i - 1)^2) + f_{bias}, z = x - o$
F4	Shifted Rastrigin's function	$\sum_{i=1}^{D}(z_i^2 - 10cos(2\pi z_i) + 10) + f_{bias}, z = x - o$
F5	Shifted Griewank's Function	$\sum_{i=1}^{D} \frac{z_i^2}{4000} - \prod_{i=1}^{D} cos(\frac{z_i}{\sqrt{i}}) + 1 + f_{bias}, z = x - o$
F6	Shifted Ackley's Function	$-20exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} z_i^2}) - exp(\frac{1}{D}\sum_{i=1}^{D} cos(2\pi z_i))$
F7	Shifted Schwefel's Problem 2.22	$\sum_{i=1}^{D} \|x_i\| + \prod_{i=1}^{D} \|x_i\|$

Table 6.3: Benchmark functions F8-F11

Functions	Name	Description
F8	Shifted Schwefel's Problem 1.2	$\sum_{i=1}^{D}(\sum_{j=1}^{D} x_j)^2$
F9	Shifted Extended F10	$(\sum_{i=1}^{D} f10(x_i, x_{i+1})) + f10(x_D, x_1)$ $f10 = (x^2 + y2)^{0.25} \times (\sin^2(x^2 + y^2))^{0.1} + 1$
F10	Shifted Bohachevsky	$\sum_{i=1}^{D}(x_i^2 + 2xi + 1^2 - 0.3\cos(3\pi x_i) - 0.4\cos(4\pi x_{i+1}) + 0.7)$
F11	Shifted Schaffer	$\sum_{i=1}^{D}(x_i^2 + x_{i+1}^2)^{0.25}(\sin^2(50(x_i^2 + x_{i+1}^2)^{0.1}) + 1)$

Table 6.4: Properties of function F1-F11

Function	Range	Optimum	Modality	Shifted	Separable	EOD
F1	$[-100, 100]^D$	-450^+	U	Y	Y	Y
F2	$[-100, 100]^D$	-449^+	U	Y	N	N
F3	$[-100, 100]^D$	-448^+	M	Y	N	Y
F4	$[-5, 5]^D$	-447^+	M	Y	Y	Y
F5	$[-600, 600]^D$	-446^+	M	Y	N	N
F6	$[-32, 32]^D$	-445^+	M	Y	Y	Y
F7	$[-10, 10]^D$	0	U	Y	Y	Y
F8	$[-65.536, 65.356]^D$	0	U	Y	N	N
F9	$[-100, 100]^D$	0	U	Y	N	Y
F10	$[-15, 15]^D$	0	U	Y	N	N
F11	$[-100, 100]^D$	0	U	Y	N	N

U- Uni-modal, M-Multi-modal, EOD- Easily Optimized dimension by dimension,

Y/N - Yes/No, D-Dimension, $+ - f_bias$

Table 6.5: Properties of function F12-F19

Function	Fns	F'	Mns	Range	Optimum
F12	NS-F9	F1	0.25	$[-100, 100]^D$	0
F13	NS-F9	F3	0.25	$[-100, 100]^D$	0
F14	NS-F9	F4	0.25	$[-100, 100]^D$	0
F15	NS-F10	$NS - F7$	0.25	$[-100, 100]^D$	0
F16	NS-F9	F1	0.5	$[-100, 100]^D$	0
F17	NS-F9	F3	0.75	$[-100, 100]^D$	0
F18	NS-F9	F4	0.75	$[-100, 100]^D$	0
F19	NS-F10	$NS - F7$	0.75	$[-100, 100]^D$	0

Fns -hybridize a non-separable function with other function F' (Fns $\bigoplus$ F'), Mns - m non-separable

Non-Separable functions

The functions $F2, F3, F5, F9$ and $F10$ are as non-separable functions. This function is quite special for our proposed work in order to analyze the performance by obtaining better results. These quite help to determine whether the introduced algorithm could be efficient in real-world problems.

Hybrid Composite functions

The Functions $F12 - F19$ are determined as hybrid composite functions. They are created by composing two functions together (i.e. a non-separable function NS is combined ($\bigoplus$) with other functions F' in order to act as both hybridized one). The decision variables in these function has both separability and non-separability characteristics. In addition to that the functions $F7 - F11$ act as the non-shifted version in order to hybridize the function.

6.3.2 Performance Metrics

In this section, the performance metrics derived aid to evaluate the performance of proposed SAGWO and also for the comparison of other recent meta-heuristics algorithms. These metrics are given as follows.

Average Standard Deviation: Standard deviation computes the distance that resides among the fitness values to obtain the best value w.r.t the known optimum value. ASD is determined with the help of estimated values using the Eq. 6.1. An algorithm which holds minimum value that indicates a better performance as compared to the other algorithms chosen.

$$ASD = \frac{\sum_{i=1}^{n} \text{SD of an benchmark}}{n} \tag{6.1}$$

where SD determines the standard deviation of a benchmark functions obtained from the n independent runs.

Success Rate (SR): Success Rate is the ratio of number of times an algorithm found the optimum or near optimum out of n independent runs. The measure is given in Eq. 6.2.

$$SuccessRate(\%) = \frac{SuccessfulRuns}{n} \times 100 \tag{6.2}$$

where n denotes the independent runs, successful runs represents the number of time the algorithm found the best.

Average Best Objective (ABO): The average best objective value is used to determine the performance of the proposed algorithm with other state-of-art meta-heuristics algorithms. The mathematical expression of the ABO is given in Eq. 6.3:

$$ABO = \frac{\sum_{r=1}^{R} \text{Best fitness of run r}}{Totalrunsn} \tag{6.3}$$

where r is the current run and n denotes the maximum number of independent runs which defines that the fitness solution obtained over the R number of runs.

6.3.3 Performance Scalability Study

In this section, the results are provided and several analyses are carried out: Initially, provided the performance obtained by the proposed work with respect to the multi population approach and self-adaptive strategies. In addition to that, the proposed work is compared with other state-of-art meta-heuristic algorithms to analyze the performance in terms efficiency and convergence diversity and as well as the scalability analysis is carried out for the proposed algorithm with other algorithms. Later, the computational time of each algorithm is compared.

The performance of SAWGO has been tested on 19 different set of benchmark functions which are introduced for the special issue CEC 2010. The performance analysis is carried out on varying the dimensions viz., $50-D, 100-D, 200-D, 500-D$ and $1000-D$ of these 19 set of benchmark functions (i.e. $D-Dimensional$). Most of the study deliberates that the $500-D$ and $1000-D$ dimensional decisions moves the function into "high-dimensional", the study is also aided to analyze the scalability of the SAGWO performance as the dimensionality of the objective function increases.

6.3.3.1 Compared Algorithms

The performance of the proposed algorithm SAGWO has been compared with recently introduced meta-heuristic algorithms which are adaptable to solve the large scale optimization problems. The algorithms namely Multi-population differential evolution with balanced ensemble of mutation strategies (mDE-bES)[Ali et al. 2015], Modified Competitive Swarm Optimization (MCSO) [Mohapatra et al. 2017] and Dynamic particle swarm optimizer with escaping prey (DPSOEP) [Chen et al. 2017b] are considered to be compared against the proposed SAGWO algorithm. The parameter settings are fixed based on the suggestions of the authors in the original sources of those papers.

6.3.3.2 Solution Quality Scalability

This section discuss about the major computation results obtained by proposed SAGWO algorithm and the various state-of-art meta-heuristic algorithms with respect to varying the dimensionality of the problems. The performance metrics such as average, median, maximum and minimum objective function values are used to analyze the performance of the algorithms. All of these algorithms are run independently 25 times to diminish arbitrary discrepancy. Table 6.6 represents the results obtained by the mDE-bES, MCSO, DPSOEP and SAGWO for evaluating the efficiency of the algorithms. Table 6.7 - 6.8 portrays the performance of the chosen algorithms and its analysis w.r.t benchmark function and performance metrics are discussed below.

F1 - Shifted Sphere Function

In Shifted Sphere function, a global optimum is found at (0, 0) and f(x) increases with the increase in scaling. The performance over the function has been tested on varying the number of dimensions from 100 to 1000 dimensions. Table 6.6 deliberates the results of SAGWO and other algorithms with varying dimensions. Thus, the SAGWO, mDE-bES, DPSOEP provides better results in terms of minimum, maximum, average best objective and median values compared to MCSO algorithm. In addition to that, we notify that the complexity of the problem increases when the problem is higher than 500 dimensions. Based on this we observed the results of 500 and 1000 dimensions for average standard deviation and success rate. The result table 6.16 for 500 dimensions portray that SAGWO, mDE-bES, DPSOEP are equal in performance but MCSO lags in its efficacy in success rate and average standard deviation. Table 6.18 results for 1000 dimensions conveys that the SAGWO, mDE-bES, DPSOEP are shows better performance compared to MCSO algorithm. Figure 6.1 and 6.2 provides the radar chart based on the success rate of the algorithm whereas the success rate of the SAGWO, mDE-bES, DPSOEP provides cent result and MCSO provides 92% in achieving the performance.

F2 - Shifted Schwefel Problem 2.21

The experimental result of Shifted Schwefel Problem 2.21 from Table 6.6 conveys that SAGWO has attained best optimum solution compared to other metaheuristics algorithms viz., mDE-bES, DPSOEP and MCSO. SAGWO considered to be better to obtain the objective value interms of the average best objective compared to other algorithms. Table 6.16 results portrays that the SAGWO is better in terms of average standard deviation and success rate for 500 dimension as well as it proves its efficacy in 1000 dimension as shown in Table 6.18. From the figure 6.1 and 6.2, success rate of the SAGWO achieves 92 percentages by repeatedly providing the best optimum for all the independent runs though the dimensionality of the problem varies.

F3 - Shifted Rosenbrock's Function

The experimental results of $ShiftedRosenbrock'sFunction$ from Table 6.7 portrays that SAGWO has attained global optimum of f(x) with null error rate for all the varying dimensionality. At the same, DPSOEP competes with the SAGWO for all the dimensionality by providing the better results over minimum, ABO and median but lags in providing better result for maximum. From Table 6.16 and 6.18 notifies that the SAGWO is better in terms of Average best objective, average standard deviation and when considering success rate SAGWO provides repeatedly better results for all independent runs over 500 and 1000 dimensions. In the aspect of the success rate, SAGWO is higher in percentage than mDE-bES, DPSOEP and MCSO indicates that SAGWO performance is better than other compared algorithms.

F4 - Shifted Rastrigin's Function

The experimental result of $ShiftedRastriginsFunction$ shows that for this test function global minimum is $f(x) = 0$ when $x = 0$ and this has been attained by the SAGWO with null error rate that has been shown in Table 6.7. SAGWO is better in achieving best value with n independent runs, thus the success rate of the SAGWO is 100 percent when compared to other algorithms for 500 dimensions as shown in Table 6.16. From Table 6.18, we notify that the success rate of SAGWO is better for 1000 dimension, thus clearly shows that even the dimensionality of the problem increases the proposed algorithm provides better results when compared to other algorithms. In addition to that, the average standard deviation and average objective value is better in performance. Figure 6.1 and 6.2 shows the radar representation of the success rate achieved by the SAGWO w.r.t to other algorithms.

F5 - Shifted Griewank's Function

The experimental result of Shifted Greiwanks test function, fitness value is 0 which has been attained by mDE-bES for small dimensions like 50 and 100 dimensions whereas as SAGWO attains the best value for varying dimensions viz., 50, 100, 200, 500 and for 1000 dimensions it provides better result when compared to other algorithms as shown in Table 6.8. Table 6.16 portrays that the SAGWO achieves better success rate compared to other algorithms. Though mDE-bES competes with SAGWO but it fails to attain the better success rate as well in average best objective and average standard deviations for both 500 and 1000 dimensions as shown in table 6.18. The radar representation of the success rate for both high dimensionality 500 and 1000 are shown in figure 6.1 and 6.2.

Table 6.6: Comparison of the average, median, maximum and minimum objective function values of (F1-F2) obtained with mDE-bES, MCSO, DPSOEP and SAGWO

Dim.	Algorithm	F1 Avg.	F1 Median	F1 Max.	F1 Min.	F2 Avg.	F2 Median	F2 Max.	F2 Min.
50	mDE-bES	0.00+00	0.00E+00	0.00E+00	0.00E+00	3.12E-01	3.01E-01	7.91E-01	2.12E-01
	MCSO	1.54E-10	1.43E-10	2.11E-10	1.75E-10	5.12E+01	5.71E+01	8.32E+01	4.74E+01
	JOA	0.00E+00	0.00E+00	0.00E+00	0.00E+00	2.01E-10	2.43E-09	1.94E-09	5.80E-10
	SAGWO	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**2.01E-14**	**1.04E-14**	**5.21E-14**	**0.00E+00**
100	mDE-bES	0.00E+00	0.00E+00	0.00E+00	0.00E+00	3.23E+00	4.23E+00	4.14E+00	2.31E+00
	MCSO	2.31E-11	2.43E-11	3.17E-11	1.75E-11	7.93E+01	1.75E+01	8.54E+01	7.12E+01
	JOA	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.47E-09	7.53E-10	2.13E-10	1.21E-11
	SAGWO	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**2.73E-13**	**2.58E-13**	**1.74E-13**	**0.00E+00**
200	mDE-bES	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.85E+01	1.72E+01	2.02E+01	1.87E+01
	MCSO	7.21E-11	2.31E-11	4.12E-06	2.54E-11	1.12E+01	1.23E+01	1.21E+02	8.43E+01
	JOA	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.21E-09	8.54E-09	4.75E-09	3.21E-10
	SAGWO	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**2.75E-14**	**2.32E-14**	**1.48E-13**	**0.00E+00**
500	mDE-bES	0.00E+00	0.00E+00	0.00E+00	0.00E+00	4.94E+01	4.91E+01	4.32E+01	3.21E+01
	MCSO	1.88E-10	3.21E-10	3.87E-10	1.39E-10	1.87E+01	1.73E+01	1.75E+01	1.01E+01
	JOA	0.00E+00	0.00E+00	0.00E+00	0.00E+00	2.43E-05	2.38E-05	4.87E-04	3.19E-05
	SAGWO	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**5.87E-14**	**5.21E-14**	**4.31E-13**	**5.31E-14**
1000	mDE-bES	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	4.32E+01	4.17E+01	3.87E+01	1.32E+01
	MCSO	2.12E-11	1.34E-11	3.70E-11	2.55E-12	1.85E+01	1.48E+01	1.74E+01	1.01E+01
	JOA	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	4.98E-04	4.12E-04	5.32E-03	4.94E-04
	SAGWO	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**9.43E-14**	**9.11E-13**	**7.31E-03**	**9.23E-14**

Table 6.7: Comparison of the average, median, maximum and minimum objective function values of (F3-F4) obtained with mDE-bES, MCSO, DPSOEP and SAGWO

Dim.	Algorithm	Benchmark Function							
		F3				F4			
		Avg.	Median	Max.	Min.	Avg.	Median	Max.	Min.
50	mDE-bES	2.72E+01	2.83E+01	2.96E+01	2.46E+01	2.17E-01	7.50E-13	7.84E-01	0.00E+00
	MCSO	1.32E+05	1.53E+02	2.13E+06	8.54E-01	3.47E+01	3.47E+02	1.00E+01	3.45E+00
	JOA	1.23E-06	3.10E-01	4.12E-06	**0.00E+00**	2.47E+00	2.10E+01	2.17E+00	3.14E-13
	SAGWO	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.02E+00	2.30E+01	0.00E+00	0.00E+00
100	mDE-bES	7.12E+01	7.14E+01	1.17E+02	7.18E+01	3.54E-08	2.00E+01	8.13E-01	2.47E-12
	MCSO	3.89E+06	5.12E+03	6.17E+06	8.19E+01	1.25E-14	2.20E+01	4.78E+01	3.47E+00
	JOA	**0.00E+00**	**0.00E+00**	1.08E-14	**0.00E+00**	0.00E+00	0.00E+00	3.17E-16	0.00E+00
	SAGWO	**0.00+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	9.12E-04	0.00E+00	0.00E+00	0.00E+00
200	mDE-bES	1.54E+02	1.19E+02	2.16E+02	1.53E+02	2.71E-01	2.79E-11	2.32E-01	5.97E-14
	MCSO	2.15E+06	1.06E+04	4.17E+07	3.01E+02	8.01E+01	3.47E+01	6.74E+01	3.79E+01
	JOA	**0.00E+00**	**0.00E+00**	1.79E-14	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	SAGWO	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
500	mDE-bES	4.78E+01	3.97E+01	3.78E+02	4.19E+02	3.31E-03	2.30E+01	3.17E+01	3.47E-11
	MCSO	1.53E+05	5.87E+02	1.97E+06	3.71E+02	8.46E-04	2.20E+01	2.77E+02	1.70E+02
	JOA	1.27E-13	**0.00E+00**	3.45E-12	**0.00+00**	**2.10E-02**	1.90E+01	**7.80E-12**	6.47E-16
	SAGWO	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.28E-04	**2.00E+01**	3.79E-02	**0.00E+00**
1000	mDE-bES	7.89E+01	8.17E+01	8.97E+01	8.46E+01	8.46E-09	2.20E+01	3.17E+00	2.78E-10
	MCSO	7.85E+03	2.47E+02	8.71E+02	7.47E+02	5.16E-02	2.00E+01	3.47E+00	6.47E-04
	JOA	6.99E+00	1.87E+00	2.47E+01	**0.00E+00**	8.46E-03	2.00E+01	7.48E-01	0.00E+00
	SAGWO	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**6.78E-14**	**2.40E+01**	**0.00E+00**	**0.00E+00**

Table 6.8: Comparison of the average, median, maximum and minimum objective function values of (F5-F6) obtained with mDE-bES, MCSO, DPSOEP and SAGWO

Dim.	Algorithm	Benchmark Function							
		F5				F6			
		Avg.	Median	Max.	Min.	Avg.	Median	Max.	Min.
50	mDE-bES	0.00E+00	0.00E+00	0.00E+00	0.00E+00	2.11E-14	3.48E-13	2.47E-05	3.14E-13
	MCSO	2.01E-02	3.17E-02	3.70E+00	3.10E-01	2.21E-06	1.32E-06	5.47E-03	2.45E-07
	JOA	6.54E+00	2.00E+01	5.47E-02	0.00E+00	9.47E-03	0.00E+00	2.47E-10	0.00E+00
	SAGWO	7.12E+00	1.80E+01	0.00E+00	0.00E+00	**4.50E-03**	0.00E+00	**3.78E-16**	0.00E+00
100	mDE-bES	**4.19E-03**	**2.00E+01**	**0.00E+00**	**0.00E+00**	**5.22E-08**	**7.44E-12**	**5.22E-12**	**6.47E-13**
	MCSO	1.74E-06	2.20E+01	5.17E-01	6.87E-10	4.94E-03	6.48E-06	1.48E-06	3.48E-07
	JOA	3.55E-03	2.94E-03	6.47E-04	6.74E-13	2.07E-12	1.78E-03	1.67E-01	3.49E-03
	SAGWO	3.99E-01	6.74E-12	3.78E-10	6.17E-16	1.27E-11	2.33E-10	3.64E-04	6.31E-11
200	mDE-bES	1.46E+00	1.78E-02	3.79E-02	7.48E-12	2.85E-01	8.31E-11	**7.04E-11**	3.78E-12
	MCSO	3.48E-02	5.79E-02	4.16E-01	3.19E-10	4.02E-06	3.14E-05	6.14E-03	7.68E-05
	JOA	8.74E-07	1.79E-06	3.47E-02	6.47E-09	8.31E-02	7.64E-03	2.59E-01	3.45E-03
	SAGWO	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**2.97E-13**	**5.11E-11**	2.64E-07	**3.64E-12**
500	mDE-bES	2.01E-01	1.80E+01	1.87E-03	6.47E-10	7.81E+00	1.90E+01	3.79E-09	6.97E-12
	MCSO	6.57E+00	2.00E+01	4.79E-02	1.11E-01	8.31E-03	2.20E+01	1.46E-03	5.40E-05
	JOA	1.03E+00	1.70E+01	3.47E-04	6.47E-09	2.50E+00	2.10E+01	1.99E-02	6.68E-04
	SAGWO	**9.69E-02**	**2.10E+01**	**0.00E+00**	**0.00E+00**	**3.81E-03**	**2.20E+01**	**0.00E+00**	**0.00E+00**
1000	mDE-bES	9.71E+00	1.90E+01	4.81E-02	3.78E-06	9.66E-04	2.20E+01	2.46E-07	3.31E-10
	MCSO	6.99E+00	1.80E+01	2.47E-01	3.14E-10	1.80E-05	2.20E+01	3.97E-05	1.45E-05
	JOA	5.35E+00	1.90E+01	**1.87E-02**	3.47E-05	8.95E-03	2.10E+01	1.44E-01	3.14E-03
	SAGWO	**3.54E+00**	**2.10E+01**	3.17E-02	**6.48E-15**	**1.41E-11**	**0.00E+00**	**0.00E+00**	**0.00E+00**

Table 6.9: Comparison of the average, median, maximum and minimum objective function values of (F7-F8) obtained with mDE-bES, MCSO, DPSOEP and SAGWO

Dim.	Algorithm	Benchmark Function							
		F7				F8			
		Avg.	Median	Max.	Min.	Avg.	Median	Max.	Min.
50	mDE-bES	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.66E-01	3.48E-01	3.69E-01	6.60E-01
	MCSO	2.24E-02	3.45E-02	3.19E-01	4.87E-03	6.16E+01	3.15E+02	5.97E+03	2.13E+01
	JOA	1.25E-09	6.45E-10	3.45E-08	6.47E-12	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	SAGWO	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	7.69E-10	3.45E-09	6.47E-06	3.74E-10
100	mDE-bES	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.23E-02	6.04E-02	2.72E-02	1.07E-02
	MCSO	2.68E-02	3.78E-02	4.67E-01	6.47E-02	3.00E+02	2.61E+01	4.98E+02	6.16E+01
	JOA	1.01E-10	6.09E-09	4.11E-08	3.97E-11	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	SAGWO	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	3.78E-11	1.23E-11	2.45E-10	3.67E-11
200	mDE-bES	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	9.29E-02	3.15E-02	6.45E-02	3.74E-02
	MCSO	1.64E-01	1.06E-01	1.16E-01	1.01E-01	2.59E+01	4.22E+01	3.78E+02	1.71E+01
	JOA	2.71E-08	3.84E-07	6.47E-06	9.81E-08	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	SAGWO	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	7.21E-12	6.97E-11	3.45E-10	6.54E-12
500	mDE-bES	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.84E-01	3.45E-02	1.27E-01	6.21E-02
	MCSO	1.49E-02	6.95E-02	1.09E-02	3.47E-02	1.66E+02	6.38E+01	7.54E+02	9.06E+01
	JOA	3.44E-11	3.64E-11	6.47E-10	7.97E-11	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	SAGWO	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.14E-13	1.53E-12	1.25E-10	3.54E-13
1000	mDE-bES	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.66E-02	3.87E-02	6.54E-02	1.64E-02
	MCSO	7.22E-01	6.47E-01	6.33E-01	5.17E-01	2.32E+01	1.01E+01	4.92E+01	1.45E+01
	JOA	5.94E-10	4.82E-09	3.47E-08	6.77E-11	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	SAGWO	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	5.94E-13	3.87E-12	4.64E-10	9.87E-13

Table 6.10: Comparison of the average, median, maximum and minimum objective function values of (F9-F10) obtained with mDE-bES, MCSO, DPSOEP and SAGWO

Dim.	Algorithm	F9 Avg.	F9 Median	F9 Max.	F9 Min.	F10 Avg.	F10 Median	F10 Max.	F10 Min.
50	mDE-bES	4.62E+02	4.58E+02	7.67E+02	2.64E+02	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	MCSO	1.10E+04	5.33E+03	1.69E+04	3.98E+03	2.12E+01	3.54E+01	6.54E+01	1.24E+01
	JOA	6.02E-02	2.14E-03	3.00E-01	2.66E-03	6.45E-01	3.45E-02	4.57E-01	7.12E-02
	SAGWO	**1.54E-04**	**3.52E-04**	**4.22E-03**	**1.40E-04**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
100	mDE-bES	7.12E+01	1.25E+01	3.46E+01	1.64E+01	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	MCSO	3.45E+03	4.58E+02	9.92E+04	2.34E+02	4.64E+02	6.21E+01	7.21E+02	3.97E+01
	JOA	7.13E-03	2.05E-02	3.50E-02	2.32E-03	3.74E-02	6.97E-01	3.21E-01	3.01E-02
	SAGWO	**8.86E-05**	**2.05E-04**	**4.58E-03**	**1.08E-05**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
200	mDE-bES	2.13E+01	4.57E+01	1.67E+02	3.64E+01	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	MCSO	7.21E+03	1.90E+02	6.16E+03	2.45E+02	3.34E+02	6.41E+01	3.54E+02	7.31E+01
	JOA	3.81E-03	1.95E-02	**1.20E-02**	**2.34E-03**	2.01E-03	6.12E-02	9.54E-02	3.42E-03
	SAGWO	**6.54E-04**	**3.54E-03**	7.96E-02	3.54E-03	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
500	mDE-bES	4.62E+01	4.77E+01	7.85E+02	4.58E+01	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	MCSO	2.04E+03	1.18E+02	1.41E+03	4.62E+02	4.14E+01	3.01E+01	5.42E+02	6.12E+01
	JOA	7.22E-03	9.34E-03	6.34E-02	3.12E-04	5.99E-03	1.64E-02	1.03E-02	3.45E-03
	SAGWO	**5.78E-06**	**1.75E-05**	**6.16E-03**	**3.50E-05**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
1000	mDE-bES	1.27E+01	1.53E+01	3.45E+01	1.01E+01	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	MCSO	1.95E+02	6.45E+01	6.54E+01	2.45E+01	6.31E+02	1.34E+01	6.34E+01	1.24E+01
	JOA	3.52E-01	6.24E-01	9.54E-01	3.01E-01	3.54E-02	1.24E-02	6.54E-01	3.01E-02
	SAGWO	**8.04E-05**	**4.62E-03**	**9.74E-03**	**6.54E-05**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**

Table 6.11: Comparison of the average, median, maximum and minimum objective function values of (F11-F12) obtained with mDE-bES, MCSO, DPSOEP and SAGWO

Dim.	Algorithm	Benchmark Function							
		F11				F12			
		Avg.	Median	Max.	Min.	Avg.	Median	Max.	Min.
50	mDE-bES	7.01E-06	2.54E-05	6.81E-05	7.12E-06	2.16E-06	3.74E-05	9.75E-04	6.75E-06
	MCSO	9.60E+00	1.42E+00	1.66E+00	1.20E-01	8.09E+02	1.03E+01	2.10E+02	6.71E+01
	JOA	3.54E-04	5.42E-04	3.12E-02	7.12E-03	9.72E-03	6.75E-02	9.12E-02	7.13E-04
	SAGWO	**7.87E-08**	**0.00E+00**	**1.21E-12**	**0.00E+00**	**3.75E-11**	**2.47E-10**	**1.87E-10**	**6.75E-11**
100	mDE-bES	1.68E-05	5.88E-04	5.14E-03	6.88E-04	3.01E-05	3.74E-06	4.12E-05	6.45E-06
	MCSO	3.82E+00	2.47E+00	3.45E+00	1.24E-01	9.74E+01	2.74E+01	3.97E+01	1.01E+01
	JOA	3.57E-05	6.54E-03	8.75E-02	6.75E-04	6.75E-03	6.12E-02	8.01E-02	1.06E-03
	SAGWO	**4.34E-12**	**0.00E+00**	**5.75E-12**	**0.00E+00**	**9.28E-10**	**7.01E-09**	**1.31E-09**	**7.64E-10**
200	mDE-bES	1.64E-06	3.21E-05	7.12E-05	3.45E-06	9.12E-05	3.45E-04	3.75E-02	6.74E-06
	MCSO	2.54E+00	3.41E+00	6.45E+00	3.21E+00	1.75E+00	6.75E+00	3.14E+00	1.21E-01
	JOA	1.75E-04	6.34E-03	2.34E-03	3.45E-04	3.74E-04	2.14E-03	1.74E-02	6.47E-03
	SAGWO	**3.21E-09**	**3.45E-08**	**3.64E-08**	**1.21E-09**	**6.71E-09**	**2.75E-08**	**3.74E-08**	**9.17E-09**
500	mDE-bES	4.43E-05	3.96E-04	2.61E-04	1.54E-05	1.72E-04	3.74E-03	6.74E-03	1.23E-04
	MCSO	6.54E+00	3.54E+00	6.75E+00	1.24E+00	7.16E+01	3.17E+00	5.74E+01	3.47E+00
	JOA	2.45E-05	3.45E-04	6.54E-04	7.01E-05	9.71E-04	3.74E-03	7.31E-02	3.73E-04
	SAGWO	**1.64E-11**	**6.75E-10**	**3.45E-09**	**6.75E-10**	**3.92E-08**	**1.75E-07**	**6.09E-08**	**2.47E-08**
1000	mDE-bES	3.98E-05	2.46E-04	1.42E-04	3.54E-05	4.15E-05	3.45E-04	6.87E-03	1.45E-05
	MCSO	4.75E+01	6.63E+00	3.75E+01	6.01E+00	3.87E+01	3.45E+00	1.34E+01	3.47E+00
	JOA	3.45E-03	1.87E-02	2.94E-02	3.07E-03	7.21E-05	1.84E-04	2.01E-03	1.07E-04
	SAGWO	**3.17E-10**	**5.21E-09**	**7.31E-08**	**6.75E-10**	**1.55E-06**	**6.37E-10**	**3.54E+00**	**9.75E-12**

Table 6.12: Comparison of the average, median, maximum and minimum objective function values of (F13-F14) obtained with mDE-bES, MCSO, DPSOEP and SAGWO

Dim.	Algorithm	Benchmark Function							
		F13				F14			
		Avg.	Median	Max.	Min.	Avg.	Median	Max.	Min.
50	mDE-bES	3.19E+01	6.17E+00	1.67E+01	3.75E+00	**2.08E-05**	2.44E-08	**2.64E-07**	**0.00E+00**
	MCSO	2.75E+03	4.17E+02	6.78E+03	5.97E+02	6.71E+02	3.47E+00	1.32E+02	3.87E-01
	JOA	6.12E-02	3.57E-03	1.37E-01	6.78E-04	5.61E-04	6.42E-03	5.08E-02	3.78E-04
	SAGWO	**3.75E-10**	**6.17E-10**	**1.12E-09**	**2.75E-11**	2.01E+00	**1.75E-11**	3.12E+00	6.45E-12
100	mDE-bES	7.30E+01	6.60E+00	5.19E+02	1.08E+00	8.01E-04	7.13E-04	**5.45E-08**	**0.00E+00**
	MCSO	3.83E+03	7.40E+02	1.55E+03	6.18E+01	3.21E+01	5.42E+00	3.47E+01	4.50E+00
	JOA	**1.43E-03**	7.17E-04	**3.45E-02**	6.19E-05	3.45E-05	6.45E-02	3.14E-02	2.41E-04
	SAGWO	3.45E-01	**6.75E-10**	3.75E-01	**6.75E-11**	**9.43E-08**	**3.17E-10**	2.47E-02	**0.00E+00**
200	mDE-bES	7.13E+00	6.97E+01	3.75E+02	3.47E+00	9.43E-04	1.91E-03	1.34E-02	1.36E-04
	MCSO	3.97E+03	1.75E+02	6.45E+01	7.13E+00	2.09E+01	1.74E+00	6.21E+02	1.42E-01
	JOA	**4.43E-03**	6.71E-02	**1.26E-01**	4.62E-03	1.07E-04	2.78E-03	**2.41E-03**	6.74E-04
	SAGWO	3.45E-01	**7.13E-09**	2.12E-01	**6.78E-10**	**5.54E-11**	**2.67E-10**	1.36E+00	**1.14E-12**
500	mDE-bES	1.27E+01	1.03E+00	6.71E+02	9.71E+00	2.45E-06	1.71E-03	2.67E-01	3.45E-05
	MCSO	2.71E+03	2.71E+00	3.74E+01	6.75E-02	2.31E+01	3.47+00	4.12E+01	3.08E-01
	JOA	**8.63E-04**	5.88E-03	1.64E-01	7.45E-04	1.19E-05	3.41E-03	3.75E-01	7.12E-04
	SAGWO	1.76E-03	**3.71E-08**	**2.71E-02**	**3.12E-10**	**1.67E-09**	**3.75E-13**	**2.45E-05**	**3.47E-12**
1000	mDE-bES	6.71E+00	5.18E-02	6.47E+01	3.97E-01	3.58E-03	3.58E-02	9.34E-01	3.64E-03
	MCSO	3.67E+03	6.11E+01	1.67E+02	3.74E+00	4.24E+01	3.74E+00	5.08E+01	1.36E+00
	JOA	**7.93E-03**	6.45E+00	6.75E+02	3.97E-03	2.49E-04	6.54E-03	2.17E-01	6.74E-03
	SAGWO	4.13E+00	**9.74E-08**	**3.45E-02**	**9.17E-09**	**5.97E-09**	**3.74E-10**	**3.74E-09**	**7.21E-10**

Table 6.13: Comparison of the average, median, maximum and minimum objective function values of (F15-F16) obtained with mDE-bES, MCSO, DPSOEP and SAGWO

Dim.	Algorithm	Benchmark Function							
		F15				F16			
		Avg.	Median	Max.	Min.	Avg.	Median	Max.	Min.
50	mDE-bES	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**2.32E-08**	2.30E-09	**2.75E-08**	5.46E-10
	MCSO	3.45E-01	2.74E-03	6.71E-02	3.31E-04	2.32E+00	3.21E-02	2.16E+00	4.42E-02
	JOA	1.07E-10	2.45E-10	1.74E-08	3.74E-10	4.18E-04	4.92E-04	1.61E-04	1.91E-05
	SAGWO	1.48E-11	3.74E-11	6.90E-10	4.26E-11	2.41E+00	**9.15E-12**	2.94E+00	**3.15E-12**
100	mDE-bES	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.04E+00	2.01E+00	2.30E+00	1.71E-01
	MCSO	2.71E-01	2.43E-01	1.97E+00	5.08E-03	6.74E+01	4.40E+02	4.71E+03	1.11E+00
	JOA	1.37E-09	2.64E-09	2.45E-08	4.16E-09	2.08E-04	2.32E-04	3.15E-03	5.46E-05
	SAGWO	1.32E-12	4.62E-11	2.45E-10	3.42E-12	**6.16E-10**	**9.55E-10**	**3.16E-08**	**4.03E-11**
200	mDE-bES	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**9.36E-05**	3.87E-05	**6.45E-04**	5.23E-05
	MCSO	7.31E+00	7.29E+00	1.07E+02	3.12E+00	4.42E+00	5.24E+00	1.25E+01	2.06E+00
	JOA	3.54E-09	1.80E-09	7.28E-09	6.91E-10	2.11E-03	1.05E-04	3.15E-03	1.25E-05
	SAGWO	1.81E-10	1.54E-11	1.01E-10	3.54E-12	2.04E+00	**1.61E-10**	1.25E+00	**5.28E-11**
500	mDE-bES	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**2.46E-06**	3.84E-06	**3.54E-04**	1.35E-07
	MCSO	1.02E+01	1.50E+01	1.78E+01	9.84E+00	7.00E+02	5.25E+01	3.81E+02	1.35E+00
	JOA	1.32E-08	3.87E-10	6.32E-09	1.39E-11	1.39E-03	2.34E-03	5.47E+00	2.03E-04
	SAGWO	3.59E-11	2.66E-10	9.32E-09	3.41E-12	1.39E+00	**2.82E-10**	3.23E+00	**5.28E-11**
1000	mDE-bES	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**2.27E-08**	2.03E-08	**6.09E-08**	9.15E-09
	MCSO	2.70E+01	1.34E+01	7.29E+02	1.80E+01	3.84E+03	2.30E+03	5.46E+04	3.87E+02
	JOA	1.07E-08	3.92E-08	4.25E-08	9.62E-09	4.42E-02	2.03E-03	2.31E+00	1.34E-03
	SAGWO	2.55E-10	2.81E-10	5.94E-10	3.54E-11	8.06E+00	**8.13E-11**	2.06E+01	**1.92E-11**

Table 6.14: Comparison of the average, median, maximum and minimum objective function values of (F17-F18) obtained with mDE-bES, MCSO, DPSOEP and SAGWO

Dim.	Algorithm	Benchmark Function							
		F17				F18			
		Avg.	Median	Max.	Min.	Avg.	Median	Max.	Min.
50	mDE-bES	3.21E-02	1.05E-01	**4.10E+00**	5.89E-02	9.33E-04	1.51E-04	2.05E-04	3.02E-05
	MCSO	1.23E+00	5.34E+01	2.80E+02	5.45E+00	4.15E+01	7.26E+00	9.60E+02	2.39E-01
	JOA	3.21E-01	2.74E+00	6.74E+00	1.74E-02	6.14E-04	**3.02E-05**	9.71E-01	1.63E-07
	SAGWO	**2.95E-02**	**4.27E-07**	1.65E+01	**9.74E-08**	**3.41E-06**	7.12E+00	**3.45E-07**	**6.74E-09**
100	mDE-bES	1.92E+00	2.05E+01	4.25E+03	1.22E+00	1.60E+01	4.14E+00	1.39E+02	9.71E+00
	MCSO	9.36E+02	9.35E+01	7.76E+02	2.34E+00	4.28E+02	3.74E+02	4.70E+03	6.41E+01
	JOA	1.91E-01	2.22E-01	**6.74E+00**	9.71E-02	**9.75E-03**	1.79E-01	**1.74E+00**	3.21E-04
	SAGWO	**2.04E-03**	**1.33E-01**	**6.74E+00**	**9.19E-03**	5.74E+00	**2.74E-06**	1.90E+00	**1.63E-07**
200	mDE-bES	4.18E+01	1.65E+00	3.07E+02	9.17E+00	**9.59E-03**	5.42E-02	2.51E-01	6.72E-03
	MCSO	2.95E+02	1.55E+00	3.45E+02	2.35E+00	1.22E+02	5.85E+02	2.05E+02	2.10E+01
	JOA	**6.87E-02**	2.32E-01	8.84E+00	3.74E-02	5.21E-02	9.59E-03	**1.40E-01**	9.36E-06
	SAGWO	9.71E+00	**4.27E-06**	**5.54E-01**	**7.32E-07**	5.43E+00	**9.42E-10**	3.74E+00	**6.41E-12**
500	mDE-bES	1.56E+02	7.00E+00	1.37E+01	6.21E-02	1.13E-01	8.74E-03	5.85E+00	6.09E-04
	MCSO	3.47E+03	6.15E+01	3.45E+00	1.60E-01	5.45E+02	2.96E+01	6.74E+03	7.12E+00
	JOA	1.11E-03	1.64E-01	2.65E+00	3.12E-02	**5.27E-05**	1.66E-04	**2.12E-01**	6.42E-06
	SAGWO	**4.40E-06**	**2.95E-02**	**1.22E-01**	**6.32E-04**	2.82E-04	**1.71E-06**	7.12E-01	**2.56E-09**
1000	mDE-bES	3.12E+00	6.41E+00	1.11E+01	3.45E-01	**1.71E-02**	**2.34E-02**	8.13E-01	8.60E-03
	MCSO	6.45E+02	1.51E+01	6.45E+03	1.74E+00	2.34E+02	7.29E+03	1.11E+04	6.42E+00
	JOA	**3.45E-04**	9.36E-03	3.41E+00	**8.31E-05**	1.92E-02	3.74E-02	**1.61E-02**	1.01E-03
	SAGWO	6.42E-03	**6.47E-03**	**7.12E-01**	3.74E-04	2.54E+01	3.41E+00	3.41E+01	**6.21E-05**

Table 6.15: Comparison of the average, median, maximum and minimum objective function values of (F19) obtained with mDE-bES, MCSO, DPSOEP and SAGWO

Dimension	Algorithm	Benchmark Function			
		F19			
		Avg.	Median	Max.	Min.
50	mDE-bES	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	MCSO	5.26E+01	3.59E+00	2.87E+02	0.00E+00
	JOA	3.51E-07	2.95E-07	5.38E-07	1.54E-11
	SAGWO	5.66E-11	6.54E-12	2.19E-10	3.01E-13
100	mDE-bES	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	MCSO	3.12E+02	6.47E+01	2.15E+03	7.82E+00
	JOA	3.14E-10	5.67E-10	3.45E-10	6.71E-11
	SAGWO	3.74E-11	6.74E-12	1.65E-10	9.74E-14
200	mDE-bES	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	MCSO	9.74E+01	6.41E+02	6.41E+03	6.45E+00
	JOA	5.32E-10	6.21E-11	3.45E-09	3.54E-12
	SAGWO	6.45E-12	3.45E-13	7.54E-12	9.63E-14
500	mDE-bES	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	MCSO	1.19E+02	5.74E+02	3.36E+04	5.47E+00
	JOA	1.11E-10	3.21E-11	5.97E-08	6.85E-12
	SAGWO	4.65E-13	3.65E-12	4.12E-10	7.21E-14
1000	mDE-bES	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	MCSO	7.50E+03	5.66E+02	1.17E+05	3.67E+00
	JOA	2.78E-09	9.74E-10	2.74E-07	3.92E-12
	SAGWO	9.12E-10	3.64E-12	2.12E-09	8.21E-14

F6 - Shifted Ackley's Function

The experimental result of Shifted Ackleys Function test function, fitness value is 0 SAGWO attains the best value for varying dimensions viz., 500 and 1000 dimensions it provides better result when compared to other algorithms as shown in Table 6.8. Table 6.16 portrays that the SAGWO achieves better success rate compared to other algorithms. Though mDE-bES competes with SAGWO but it fails to attain the better success rate as well in average best objective and average standard deviations for both 500 and 1000 dimensions as shown in table 6.18. The radar representation of the success rate for both high dimensionality 500 and 1000 are shown in figure 6.1 and 6.2.

F7 - Shifted Schwefel's Problem 2.22

The experimental result of Shifted Schwefels Problem 2.22 test function has global minimum as 0 which has been attained by mDE-bES and SAGWO for all dimensions it provides better result when compared to other algorithms as shown in Table 6.9. Table 6.16 portrays that the SAGWO achieves better success rate compared to other algorithms. Though mDE-bES competes with SAGWO but it fails to attain the better success rate as well in average best objective and average standard deviations for both 500 and 1000 dimensions as shown in table 6.18. The radar representation of the success rate for both high dimensionality 500 and 1000 are shown in figure 6.1 and 6.2.

F8 - Shifted Schwefels Problem 1.2

The experimental results of Shifted Schwefels Problem 1.2 shows that for this test function, fitness value is 0 which has been attained by DPSOEP for all dimensions it provides better result when compared to other algorithms as shown in table 6.9. SAGWO algorithm identifies the best position and too achieve the better best value compared to other two algorithms. Table 6.16 portrays that the DPSOEP achieves better success rate compared to other algorithms. Though DPSOEP competes with SAGWO, it both performs better and attains the better success rate as well in average best objective and average standard deviations for both 500 and 1000 dimensions as shown in table 6.18. The radar representation of the success rate for both high dimensionality 500 and 1000 are shown in figure 6.1 and 6.2.

F9 - Shifted Extended F10

The experimental result of Shifted Extended F10 shows that for this test function, fitness value is 0 which has attained nearby optimum by SAGWO compared to mDE-bES, MCSO and DPSOEP. In case of 200 dimensions SAGWO slightly lags in DPSOEP for best and worst value but it proves its efficacy over average best objective and median values as shown in table 6.10. Table 6.16 portrays that the SAGWO achieves better success rate compared to other algorithms. Though DPSOEP competes with SAGWO but it fails to attain the better success rate as well in average best objective and average standard deviations for both 500 and 1000 dimensions as shown in table 6.18. The radar representation of the success rate for both high dimensionality 500 and 1000 are shown in figure 6.1 and 6.2.

F10 - Shifted Bohachevsky

The experimental results of Shifted Bohachevsky shows that for this test function, fitness value is 0 which has been attained by mDE-bES and SAGWO for all dimensions it provides better result when compared to other algorithms as shown in table 6.10. Table 6.16 portrays that the SAGWO achieves better success rate compared to other algorithms. Though mDE-bES competes with SAGWO, it fails to attain the better success rate as well in average best objective and average standard deviations for both 500 and 1000 dimensions as shown in table 6.18. The radar representation of the success rate for both high dimensionality 500 and 1000 are shown in figure 6.1 and 6.2.

F11 - Shifted Schaffer

The experimental result of Shifted Schaffer shows that for this test function, fitness value is 0 which has been attained nearby optimum by SAGWO compared to mDE-bES, MCSO and DPSOEP. In case of 200 dimensions SAGWO slightly lags in DPSOEP for best and worst value but it proves its efficacy over average best objective and median values as shown in table 6.11. Table 6.17 portrays that the SAGWO achieves better success rate compared to other algorithms. Though DPSOEP competes with SAGWO, it fails to attain the better success rate as well in average best objective and average standard deviations for both 500 and 1000 dimensions as shown in table 6.19. The radar representation of the success rate for both high dimensionality 500 and 1000 are shown in figure 6.1 and 6.2.

F12 Hybrid Function (NS-F9)

The experimental result of Hybrid Function (NS-F9) shows that for this test function, fitness value is 0 which has been attained nearby optimum by SAGWO compared to mDE-bES, MCSO and DPSOEP. SAGWO proves its efficacy over average best objective and median values as shown in table 6.11. Table 6.17 portrays that the SAGWO achieves better success rate compared to other algorithms. SAGWO attains the better success rate as well in average best objective and average standard deviations for both 500 and 1000 dimensions as shown in table 6.19. The radar representation of the success rate for both high dimensionality 500 and 1000 are shown in figures 6.1 and 6.2.

F13 - Hybrid Function (NS-F9)

The experimental result of Hybrid Function (NS-F9) shows that for this test function, fitness value is 0 which has been attained nearby optimum by SAGWO compared to mDE-bES, MCSO and DPSOEP. SAGWO proves its efficacy over average best objective and median values as shown in table 6.12. Table 6.17 portrays that the SAGWO achieves better success rate compared to other algorithms. SAGWO attains the better success rate as well in average best objective and average standard deviations for both 500 and 1000 dimensions as shown in table 6.19. The radar representation of the success rate for both high dimensionality 500 and 1000 are shown in figures 6.1 and 6.2.

F14 - Hybrid Function (NS-F9)

The experimental result of Hybrid Function (NS-F9) shows that for this test function, fitness value is 0 which has attained nearby optimum by SAGWO compared to mDE-bES, MCSO and DPSOEP. Though mDE-bES competes with the SAGWO, it fails to compete over the high dimensionality thus the SAGWO proves its efficacy over average best objective and median values as shown in table 6.12. Table 6.17 portrays that the SAGWO achieves better success rate compared to other algorithms. SAGWO attains the better success rate as well in average best objective and average standard deviations for both 500 and 1000 dimensions as shown in table 6.19. The radar representation of the success rate for both high dimensionality 500 and 1000 are shown in figures 6.1 and 6.2.

F15 - Hybrid Function (NS-F10)

The experimental results of Hybrid Function (NS-F10) shows that for this test function, fitness value is 0 which has been attained by mDE-bES for all dimensions it provides better result when compared to other algorithms as shown in table 6.13. Table 6.17 portrays that the SAGWO achieves better success rate compared to other algorithms. Though mDE-bES competes with SAGWO, it fails to attain the better success rate as well in average best objective and average standard deviations for both 500 and 1000 dimensions as shown in table 6.19. The radar representation of the success rate for both high dimensionality 500 and 1000 are shown in figure 6.1 and 6.2.

Table 6.16: Comparison of the Average Best Objective (ABO), Average Standard Deviation (ASD) and Success Rate (SR) for 500 Dimension benchmark problems obtained with mDE-bES, MCSO, DPSOEP and SAGWO.

Fun.	Algorithm											
	mDE-bES			MCSO			DPSOEP			SAGWO		
	ABO	ASD	SR	ABO	ASD	SR	ABO	ASD	SR	ABO	ASD	SR
F1	0.00E+00	0.00E+00	100	1.88E-10	1.21E-09	92	0.00E+00	0.00E+00	100	**0.00E+00**	0.00E+00	100
F2	4.94E+01	3.21E+00	80	1.87E+01	1.77E+01	76	2.43E-05	2.01E-05	88	**5.87E-14**	4.32E-15	92
F3	4.78E+01	3.43E+01	88	1.53E+05	2.38E+02	68	1.27E-13	3.45E-12	92	**0.00E+00**	0.00E+00	100
F4	2.97E-01	1.64E-01	84	2.19E+02	1.67E+00	68	**3.18E-13**	2.51E-14	88	1.97E-04	1.23E-05	92
F5	3.79E-05	3.55E-04	88	2.11E-02	1.92E-02	76	7.18E-06	6.34E-07	88	**0.00E+00**	0.00E+00	100
F6	2.16E-11	1.69E-12	92	2.10E-05	1.74E-05	84	3.29E-03	2.45E-03	84	**0.00E+00**	0.00E+00	100
F7	**0.00E+00**	0.00E+00	100	1.49E-02	1.27E-02	84	3.44E-11	6.65E-11	92	**0.00E+00**	0.00E+00	100
F8	1.84E-01	2.45E-02	88	1.66E+02	3.63E+01	80	**0.00E+00**	0.00E+00	100	1.14E-13	9.72E-14	96
F9	4.62E+01	3.79E+00	80	2.04E+03	7.51E+02	64	7.22E-03	3.64E-05	84	**5.78E-06**	9.54E-08	96
F10	**0.00E+00**	0.00E+00	100	4.14E+01	8.68E-03	76	5.99E-03	3.41E-04	88	**0.00E+00**	0.00E+00	100

Table 6.17: Comparison of the ABO, ASD and SR for 500 Dimension benchmark problems obtained with mDE-bES, MCSO, DPSOEP and SAGWO.

Fun.	Algorithm											
	mDE-bES			MCSO			DPSOEP			SAGWO		
	ABO	ASD	SR	ABO	ASD	SR	ABO	ASD	SR	ABO	ASD	SR
F11	4.43E-05	6.87E-07	88	6.54E+00	8.31E-02	80	2.45E-05	9.71E-05	84	**1.64E-11**	3.96E-10	96
F12	1.72E-04	7.12E-01	92	7.16E+01	3.01E+00	84	9.71E-04	2.08E-06	92	**3.92E-08**	8.54E-09	96
F13	1.27E+01	3.65E-01	88	2.71E+03	2.85E+00	80	**8.63E-04**	5.25E-04	96	1.76E-03	5.14E-05	92
F14	2.45E-06	3.54E-07	84	2.31E+01	5.78E-01	88	1.19E-05	6.74E-06	92	**1.67E-09**	3.84E-08	88
F15	**0.00E+00**	0.00E+00	100	1.02E+01	3.58E+00	92	1.32E-08	2.45E-06	96	3.59E-11	3.54E-09	92
F16	**2.46E-06**	6.84E-07	92	7.00E+02	6.54E+00	88	1.39E-03	6.51E-02	84	1.39E+00	6.97E+00	88
F17	1.56E+02	9.54E+00	88	3.47E+03	4.52E+01	92	1.11E-03	5.98E-04	92	**4.40E-06**	9.54E-07	96
F18	1.13E-01	2.54E-03	84	5.45E+02	3.84E+00	88	**5.27E-05**	6.84E-06	96	2.82E-04	1.31E-05	96
F19	**0.00E+00**	0.00E+00	100	1.19E+02	3.51E+00	88	1.11E-10	3.51E-10	92	4.65E-13	5.34E-13	96

Table 6.18: Comparison of the ABO, ASD and SR for 1000 Dimension benchmark problems.

Fun.	Algorithm											
	mDE-bES			MCSO			DPSOEP			SAGWO		
	ABO	ASD	SR	ABO	ASD	SR	ABO	ASD	SR	ABO	ASD	SR
F1	**0.00E+00**	0.00E+00	100	2.12E-11	3.54E-10	92	**0.00E+00**	0.00E+00	100	**0.00E+00**	0.00E+00	100
F2	4.32E+01	2.47E+00	84	1.85E+01	6.54E+00	80	4.98E-04	9.47E-03	88	**9.43E-14**	6.45E-15	92
F3	7.89E+01	1.02E+00	92	7.85E+03	7.12E+00	72	6.99E+00	4.50E-03	92	**0.00E+00**	0.00E+00	100
F4	3.14E+00	8.49E-02	88	2.78E+01	3.82E-01	72	2.47E-01	7.97E-02	84	**0.00E+00**	7.81E-13	92
F5	2.47E-06	3.54E-08	80	4.78E-02	4.19E-03	80	6.31E-06	5.22E-08	88	**3.17E-12**	0.00E+00	100
F6	1.44E-12	1.25E-14	88	4.13E-05	1.74E-06	88	3.45E-02	4.94E-03	84	**0.00E+00**	0.00E+00	100
F7	**0.00E+00**	0.00E+00	100	7.22E-01	3.55E-03	84	5.94E-10	2.07E-12	92	**0.00E+00**	0.00E+00	100
F8	2.66E-02	9.12E-04	88	2.32E+01	3.99E-01	80	**0.00E+00**	0.00E+00	100	5.94E-13	2.55E-14	96
F9	1.27E+01	1.12E+00	80	1.95E+02	7.26E+00	72	3.52E-01	3.64E-05	84	**8.04E-05**	9.54E-08	96
F10	**0.00E+00**	0.00E+00	100	6.31E+02	1.46E+00	76	3.54E-02	2.85E-01	88	**0.00E+00**	0.00E+00	96

Table 6.19: Comparison of the ABO, ASD and SR for 1000 Dimension benchmark problems.

Fun.	Algorithm											
	mDE-bES			MCSO			DPSOEP			SAGWO		
	ABO	ASD	SR	ABO	ASD	SR	ABO	ASD	SR	ABO	ASD	SR
F11	3.98E-05	3.31E-03	92	4.75E+01	2.01E-01	72	3.45E-03	7.81E+00	76	**3.17E-10**	4.70E-08	92
F12	4.15E-05	8.46E-04	88	3.87E+01	6.57E+00	80	7.21E-05	8.31E-03	88	**1.55E-06**	4.94E-05	88
F13	6.71E+00	2.10E-02	76	3.67E+03	1.03E+00	68	**7.93E-03**	2.50E+00	84	4.13E+00	5.86E-03	84
F14	3.58E-03	1.28E-04	80	4.24E+01	9.69E-02	84	2.49E-04	3.81E-03	88	**5.97E-09**	1.22E-11	92
F15	**0.00E+00**	2.04E-14	96	2.70E+01	1.14E-02	80	1.07E-08	6.57E-09	92	2.55E-10	4.71E-09	96
F16	**2.27E-08**	8.46E-09	88	3.84E+03	9.71E+00	76	4.42E-02	9.66E-04	88	8.06E+00	2.07E-01	88
F17	3.12E+00	5.16E-02	80	6.45E+02	6.99E+00	72	**3.45E-04**	1.80E-05	88	6.42E-03	4.51E-04	92
F18	**1.71E-02**	8.46E-03	80	2.34E+02	5.35E+00	76	1.92E-02	8.95E-03	84	2.54E+01	1.38E-03	88
F19	**0.00E+00**	6.78E-14	96	7.50E+03	3.54E+00	84	2.78E-09	1.41E-11	92	9.12E-10	9.12E-13	96

F16 - Hybrid Function (NS-F9)

The experimental result of Hybrid Function (NS-F9) shows that for this test function, fitness value is 0. Though it fails to achieve the optimum but it achieves the nearer optimum solution by SAGWO for all dimensions it provides better result when compared to other algorithms as shown in Table 6.13. Table 6.17 portrays that the SAGWO achieves better success rate compared to other algorithms. Though mDE-bES competes with SAGWO, it fails to attain the better success rate as well in average best objective and average standard deviations for both 500 and 1000 dimensions. The radar representation of the success rate for both high dimensionality 500 and 1000 are shown in figure 6.1 and 6.2.

F17 Hybrid Function (NS-F9)

The experimental result of Hybrid Function (NS-F9) shows that for this test function, fitness value is 0 which was attained by SAGWO for all dimensions it provides better result when compared to other algorithms as shown in Table 6.14. SAGWO algorithm identifies the best position and too achieve the better best value compared to other two algorithms. Table 6.14 portrays that the DPSOEP achieves better success rate compared to other algorithms. Though DPSOEP competes with SAGWO both performs better and attains the better success rate as well in average best objective and average standard deviations for both 500 and 1000 dimensions. The radar representation of the success rate for both high dimensionality 500 and 1000 are shown in figure 6.1 and 6.2.

F18 - Hybrid Function (NS-F9)

The experimental result of Hybrid Function (NS-F9) shows that for this test function, fitness value is 0 which was attained by SAGWO for all dimensions it provides better result when compared to other algorithms in terms of best value as shown in Table 6.14. SAGWO algorithm identifies the best position and too achieve the better best value compared to other two algorithms. Table 6.17 portrays that the DPSOEP achieves better success rate compared to other algorithms. Though DPSOEP competes with SAGWO both performs better and attains the better success rate as well in average best objective and average standard deviations for both 500 and 1000 dimensions as shown in table 6.19. The radar representation of the success rate for both high dimensionality 500 and 1000 are shown in figure 6.1 and 6.2.

F19 Hybrid Function (NS-F10)

The Shifted Greiwanks function has the fitness value as 0 which has been attained by SAGWO for small dimensions like 50 and 100 dimensions whereas as SAGWO attains the best value for varying dimensions viz., 50, 100, 200, 500 and for 1000 dimensions it provides better result when compared to other algorithms as shown in table 6.15. Table 6.17 portrays that the SAGWO achieves better success rate compared to other algorithms. Though mDE-bES competes with SAGWO, it fails to attain the better success rate as well in average best objective and average standard deviations for both 500 and 1000 dimensions as shown in table 6.19. The radar representation of the success rate for both high dimensionality 500 and 1000 are shown in figure 6.1 and 6.2.

The overall observation of the success rate for all test functions has been presented in Figure 6.3. It clearly shows that the proposed SAGWO provides better success rate compare to all the algorithms for 500 and 1000 dimensions problems. The SAGWO obtains 96% (approx.) for 500 dimensions whereas mDE-bES achieves 90% and DPSOEP achieves 91% (approx.) and MCSO achieves 81% (approx.). At the same time, when the dimensionality of the problem increases from 500 to 1000 dimensions, the SAGWO algorithm provides better results for all test functions with 94% (approx.) whereas mDE-bES achieves 88% and DPSOEP achieves 88% (approx.) and MCSO achieves 78% (approx.). This success rate conveys that SAGWO algorithm provides better results for all the n independent runs which shows that the algorithm has the capability to adapt to the problem though the scalability of the problem increases.

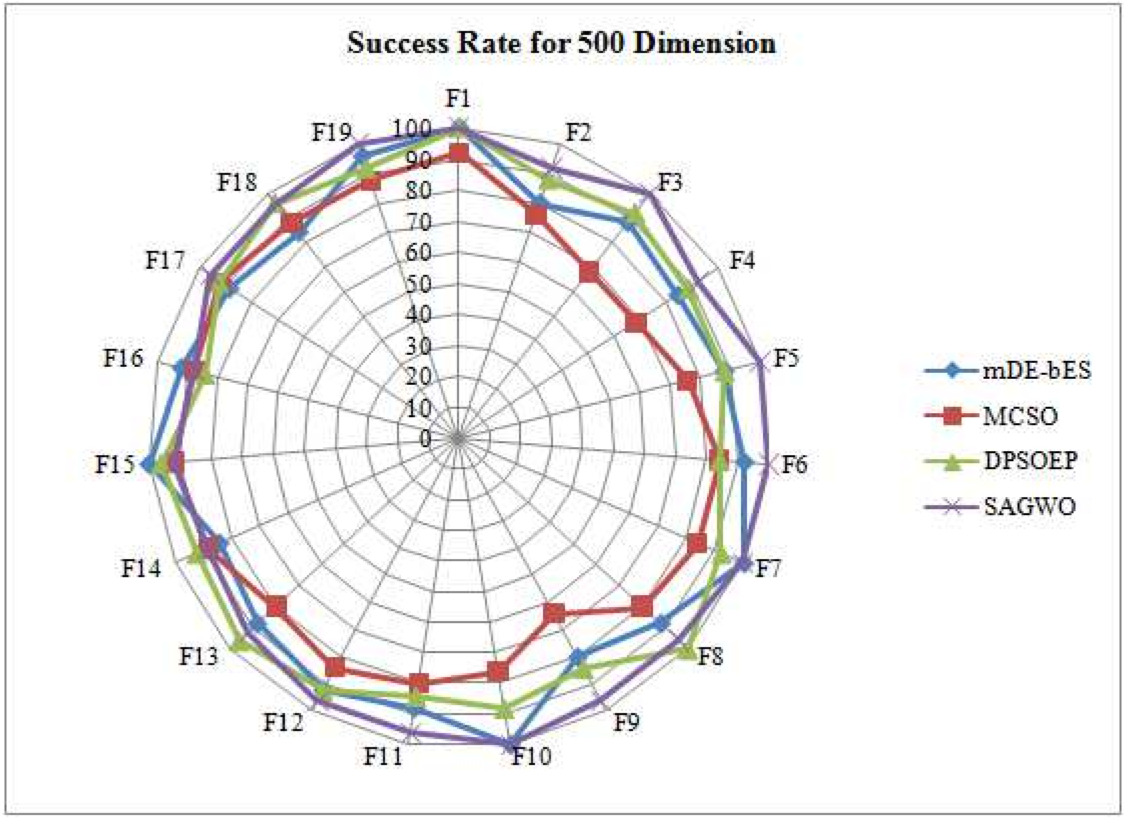

Figure 6.1: Comparison of 500 Dimensional benchmark problems w.r.t Success Rate for mDE-bES, MCSO, DPSOEP and SAGWO.

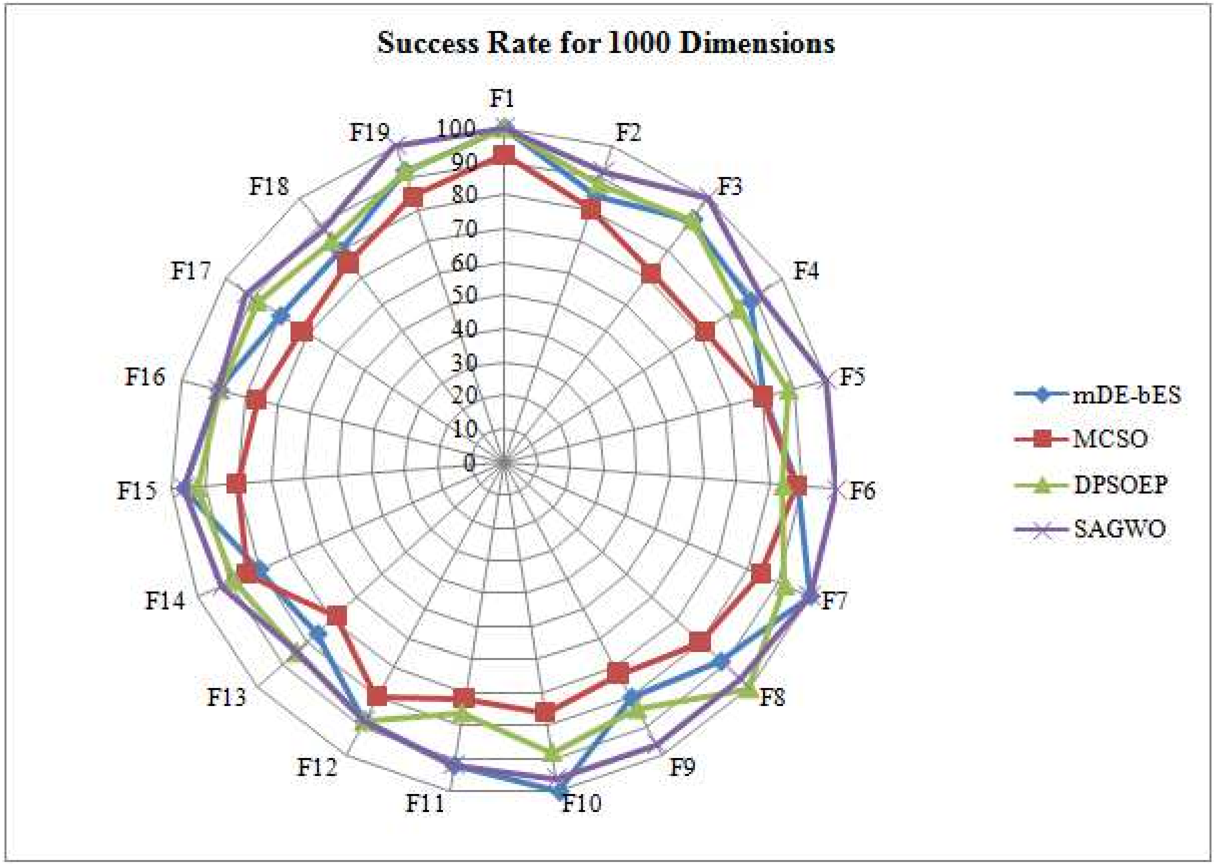

Figure 6.2: Comparison of 1000 Dimensional benchmark problems w.r.t Success Rate for mDE-bES, MCSO, DPSOEP and SAGWO.

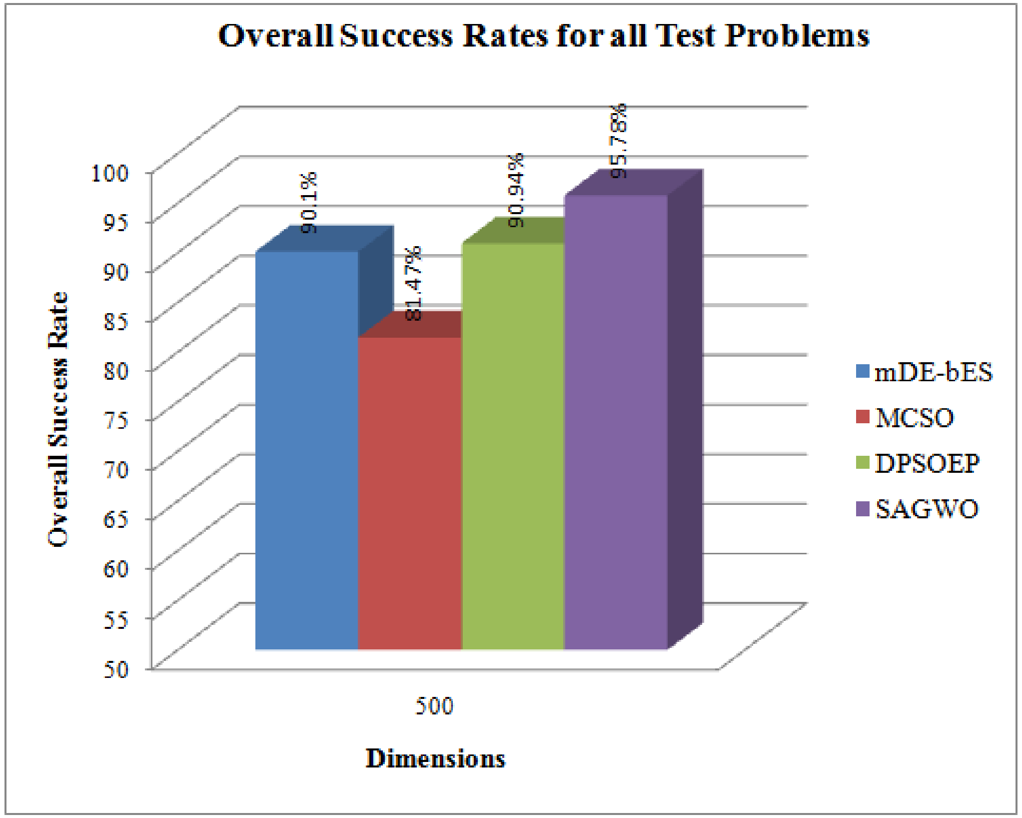

Figure 6.3: Overall success rates of all test problems for 500 and 1000 dimensions

6.3.3.3 Sensitivity Analysis

This section utilizes the functionality of the SAGWO with varying population size and number of subgroups. Moreover, population size plays a vital role in determining the stagnation or premature convergence of the algorithms. Generally, when the population size increases, then the problem of premature convergence and stagnation reduces as much as possible. Increase in the population size has the effect of reducing the convergence velocity. As a result, appropriate size of population has to be determined when dealing with the large-scale problems. The aim of dealing the sensitivity analysis is to study the effects of the population size on the new proposal. In addition to that, the number subgroup aids to identify the best optimum by splitting the population into different groups. This helps to legitimate the local optima stagnation when the dimensionality of the problem increases gradually. Normally, the problem holds high local optima when the size of the dimension of a problem increases. In order to handle it, the algorithm should have the capability to deal with those problems by adapting its population size and number of subgroups.

Table 6.20 shows the results obtained by varying the number of population size and number of groups for high dimensionality problems viz., 500 and 1000 dimensions. The problem with lesser than 500 dimensions can be easily solved by the most of the algorithms but it fails to achieve the optimum when the dimensionality of the problem increases, this analysis helps to prove the proposed capability by handling the large-scale problems. From the Table 6.20, we observed that for the population size 100 and number of subgroups 4 gets best value compared to other variations on the population size and subgroups. In order to observe the population size and subgroups, we selected some of the functions from separable, non-separable and hybridized functions such as F1, F3, F5, F7, F8, F15 and F19.

Based on this observation, the proposed SAGWO algorithm fix the population size as 100 and subgroups as 4 for all the experimentation of benchmark functions.

Table 6.20: Sensitive analysis with varying population size and swarm size

NP	D=500					D=1000				
	50	75	100	125	150	50	75	100	125	150
F1	1.46E-10	3.54E-11	**0.00E+00**	4.62E-11	6.21E-09	5.12E-13	6.12E-14	**0.00E+00**	3.74E-11	6.74E-13
F3	3.45E-06	5.62E-08	**7.54E-09**	4.45E-07	2.14E-05	7.39E-08	5.27E-07	5.14E-08	9.50E-08	**4.19E-08**
F5	7.64E-10	6.08E-10	5.48E-11	2.11E-05	**6.54E-12**	3.11E-10	6.45E-12	**0.00E+00**	4.21E-13	2.12E-11
F7	**5.41E-13**	2.15E-06	1.84E-12	7.26E-12	5.47E-07	5.47E-12	6.12E-10	**3.45E-13**	2.41E-10	6.41E-10
F8	3.54E-03	4.12E-05	**9.12E-07**	1.21E-06	3.97E-06	4.33E-10	7.95E-09	3.11E-08	**5.87E-11**	9.64E-10
F15	2.41E-07	5.68E-05	**6.74E-08**	9.33E-07	9.64E-04	6.45E-10	2.10E-11	7.21E-11	3.74E+10	**6.97E-12**
F19	3.97E-10	1.05E-13	3.51E-08	1.85E-07	**3.97E-11**	2.71E-12	6.44E-12	**7.95E-14**	3.66E-12	4.52E-13
S	2	3	4	5	6	2	3	4	5	6
F1	**3.15E-11**	6.23E-07	5.21E-09	1.74E-07	0.00E+00	9.12E-12	6.74E-13	**5.45E-14**	6.54E-12	6.47E-11
F3	4.56E-07	6.87E-07	**9.74E-09**	8.15E-08	5.14E-06	3.74E-09	2.10E-08	**1.11E-10**	7.79E-10	3.51E-09
F5	3.64E-09	7.34E-10	6.54E-09	3.45E-09	**5.74E-10**	7.12E-10	6.66E-10	5.45E-09	2.34E-10	**2.12E-11**
F7	6.95E-11	3.12E-10	**2.31E-13**	6.54E-11	2.87E-10	6.97E-10	2.21E-12	**9.54E-13**	4.32E-12	7.87E-11
F8	8.71E-05	7.32E-06	3.45E-07	**3.74E-09**	6.74E-08	**1.21E-07**	3.22E-07	2.45E-07	3.97E-07	9.87E-07
F15	5.62E-10	3.45E-11	**1.45E-13**	6.45E-10	7.12E-11	2.87E-11	6.82E-12	7.11E-10	**4.75E-12**	5.74E-10
F19	3.67E-09	8.54E-11	6.74E-12	**7.12E-13**	6.74E-12	3.96E-10	5.21E-11	**5.44E-12**	7.64E-11	8.71E-10

Note: NP Number of Populations, S Number of Sub groups, D Dimension of the problems

6.4 Summary

Experimentation and result analysis steps were utilized for evaluating and projecting the efficacy of the proposed SAGWO approach when compared with mDE-bES, MCSO and DPSOEP. Large scale benchmark functions were taken to evaluate the performance of the proposed SAGWO algorithm. To assess the capability of the algorithm in solving benchmark test functions and attaining global optimum, different performance metrics and various criteria were considered. From the result, it clearly shows that SAGWO has the ability to attain objective value for 96% of the test functions which shows an improvement over other compared algorithm as well as the classical GWO. Based on the success rate and average standard deviation of the test function, we notify that SAGWO outperforms over test functions. In case of sensitivity analysis, the experimentation over the population size and subgroups clearly deliberates that the SAGWO has the capability to handle the local optima and premature convergence. Finally, it has been depicted that SAGWO algorithm has visibly improved the efficiency of algorithm, and it has the capability to search the large space to obtain the optimal solution efficiently compared to other algorithms. This shows that the SAGWO has shown its efficacy in identifying the best suitable point in the search space for the best functional expression in an environment with improved quality of solution.

Chapter 7

SAGWO for Real Time Applications

7.1 Introduction

Self adaptive grey wolf optimization is an enriched algorithm formulated to handle the large scale optimization problems which influences or regulates the activities of the grey wolf in the same environment. This algorithm was computationally modeled and described in the chapter 4 and 5. Self adaptive grey wolf plays an important role in adapting their behaviors for scalability problems which leads to determine the global optimal solutions by eradicating the local optimal stagnation. Further, neighborhood search mechanism and position repulsion operators are incorporated to improve the quality of the solutions and to attain the same in a minimum computational time.

To assess the efficiency of the proposed SAGWO algorithm, large scale test functions have been chosen initially and it was analyzed in chapter 6. In addition to that, the real world problems viz., the economic load dispatch and localization problems are chosen to analyze the performance of SAGWO over existing algorithms. Economic Load Dispatch (ELD) is one of the general problems in power systems that considered as the large scale problem in case of the number of generating units in the system increases gradually. In general, solving this problem with multiple constraints is quite complex as well as finding the optimal solution is crucial task.

In addition to that, localization problem is considered as one of the challenging problem in wireless sensor networks. In case of identifying the position of the sensor nodes in dense network is crucial because inaccurate position might lead the network to failure state. At the same time, the increase in number of unknown nodes will change the networks to quite complex and non-trivial task. The proposed SAGWO has self-adaptability and complexity over solving the complex and large scale problems which easily attain an optimal solution.

7.2 Economic Load Dispatch (ELD) Problem

Economic load dispatch (ELD) problem is a well-established complex optimization problem which includes the non-convexity, non-linearity and high-dimensionality. The complexity of the problem arises due to the design specifications and operation constraints of the generating units such as the turning holds, transmission losses, restricted activity zones, esteem point impacts, and various fuel alternatives. The objective of this problem is to identify the optimal combination of the output power of all the online generating units that minimize the total fuel cost, while satisfying various constraints generated on the system and units.

In reality, most of the thermal generating units are provided with the different fuel like coal, natural gas and oil. In this situation, the function of ELD problem is altered from single quadratic function into non-smooth quadratic function. In addition to that, value point effects are quite important to obtain the accurate cost. In practical, considering both the value point effects and different fuel options at the same time for large scale ELD with hundreds of generating units would pose a crucial task because of discontinuous values and local optima. In such case, solving this problem using the traditional optimization techniques is quite complex.

In this work, a novel optimization technique namely SAGWO algorithm is applied to solve the large scale ELD problems by considering both multiple fuel options and value-point effects simultaneously. As per the discussion in chapter 6, we notified that the SAGWO is quite efficient to solve the complex and high dimensionality problems through efficient search operators, namely neighborhood search mechanism, position repulsion mechanism and global best oscillation techniques. In addition to that, this algorithm overcomes the issues of local optima stagnation and premature convergence by balancing the exploration and exploitation process.

Objective Function The objective of the ELD problem is to reduce the total fuel cost of power generation by compensating the equality and inequality constraints imposed on the system and units. In practical, the ELD cost function with both multiple fuel options and value point effects is given in Eq. 7.1:

$$f(P_i) = \sum_{d=1}^{D} F(P_{i,d}) + |\sum_{d=1}^{D} P_{i,d} - PW_{demand} - PW_{loss}| \times W \tag{7.1}$$

where, w denotes the penalty coefficient is fixed as 0.55 for all the test instances, PW_{demand} represents the power demand and PW_{loss} transmission power loss.

$$F(P_i) = \begin{cases} a_{i1} + b_{i1}P_i + c_{i1}P_i^2 + |e_{i1}sin[f_{i1}(P_i^{min} - P_i)]|, & P_i^{min} \leq P_i \leq P_{i1} \\ a_{i2} + b_{i2}P_i + c_{i2}P_i^2 + |e_{i2}sin[f_{i2}(P_i^{min} - P_i)]|, & P_{i1} \leq P_i \leq P_{i2} \\ . & . \\ . & . \\ . & . \\ a_{ik} + b_{ik}P_i + c_{ik}P_i^2 + |e_{ik}sin[f_{ik}(P_i^{min} - P_i)]|, & P_{i(k-1)} \leq P_i \leq P_i^{max} \end{cases} \tag{7.2}$$

where $a_{ik}, b_{ik}, c_{ik}, e_{ik}$ and f_{ik} are cost coefficient of the ith generator using the fuel type k. P_i^{max} - Maximum power output of the i^{th} generating unit.

7.2.1 Experimental Setup

For ELD, the test case systems are taken from [Chiang 2007] that contains the number of units, min and max of generation power for every unit along with the different fuel options and the cost coefficients. The proposed SAGWO algorithm undergoes for the six different types of test case systems namely 10-640 units with value point effects and different fuel options. The optimal results achieved by SAGWO algorithm is compared with the other state-of-art of meta-heuristics algorithms proposed in the literature. On the other hand to validate the efficiency of the proposed algorithm namely 320 and 640-unit systems are taken as large scale instance which has huge local optima and high dimensionality.

The parameters used in the SAGWO for experiment and its corresponding values are reported in table 6.1. Initial population size for the SAGWO is set as 100 and the maximum number of generations is considered as 1000. In addition to that, the penalty coefficient to handle the constraints is fixed as 0.55. Furthermore, each test case has been repeated for 100 trails in order to minimize the statistical errors. The cost and the power output in all tables and figures are represented as \$/h and MW.

7.2.2 Performance Metrics

In order to analyze the performance of the proposed SAGWO algorithm, comparisons are made with other algorithms namely CSO, OGWO and LFA respectively. Six performance metrics were used for evaluating the experimental results, which are mentioned in this section [Meng et al. 2016; Chiang 2005; Park et al. 2010].

Total Number of Function Evaluations (TFE):The total number of function evaluations (TFE) is determined as number of function evaluations required for a single run is given in Eq. 7.3.

$$TFE = \varepsilon \star PopulationSize \star MaximumnumberofIterations \tag{7.3}$$

where ε represented as the number of objective evaluations performed by an algorithm when the population size is set to be 1 and number of generations is set to be 1. The function evaluation may vary only if the algorithm stagnates in local optima for certain number of generations otherwise the proposed SAGWO algorithm has $\varepsilon = 1$.

7.2.3 Experimental Analysis

This section discuss about the efficiency of the proposed SAGWO algorithm and compared with OGWO [Pradhan et al. 2017], LFA [Kheshti et al. 2017] and CSO [Cheng and Jin 2015] algorithms by considering the performance metrics that are discussed in section 7.2.2. The test instances are taken from the [Chiang et al. 2005] for the experiment are categorized into three classifications namely small scale instance (10-unit system), medium scale instance (40, 80 and 160-unit system) and large scale instance (320 and 640-unit system) and the experimental setup are same for all the instances. Table 7.1 7.8 and figure 7.1 - 7.12 reveal the performance of the SAGWO in case of 10 to 640-unit systems.

7.2.3.1 Small Scale Test Instance: 10-unit systems

Small test instance deals with the small scale system namely 10-unit systems with different fuel options and value point effects are used respectively. This instance has been most commonly used as the benchmark test instance by all meta-heuristic algorithms in order to analyze the performance of the same. The total demand for this instance is fixed as the 2700 MW and the transmission loss has not been considered. The best solution obtained by SAGWO algorithm and other compared algorithms for 10-unit system is represented in Table 7.1. The table 7.1 results clearly shows that all the algorithms provides the final power output while analyzing the total fuel cost, the proposed SAGWO algorithm provides minimum fuel cost compare to other algorithms.

Table 7.1: Best power output for 10-unit system with load demand of 2700 MW

Units	**Fuel type**	**SAGWO**	**OGWO**	**LFA**	**CSO**
1	2	217.6954	218.7524	219.5247	219.1274
2	1	211.7024	211.1034	211.0346	211.3547
3	1	280.6461	280.5341	279.3421	279.2218
4	3	239.3122	239.4516	239.4476	239.6071
5	1	279.3475	279.8249	280.6521	280.2478
6	3	239.9715	239.7213	239.3525	239.4217
7	1	287.9571	287.6304	287.2463	287.5616
8	3	239.8233	239.2145	240.0527	240.271
9	3	427.6024	427.8247	428.2961	428.0624
10	1	275.9421	275.9426	275.0512	275.1246
Total power output		2700	2700	2700	2700
Total cost		622.863	623.828	623.621	626.754

The performance analysis of the proposed SAGWO algorithm and other algorithms are given in table 7.2. From the table 7.2, proposed SAGWO algorithm provides better results for all the performance metrics namely minimum, maximum, and average fuel cost. At the same time the proposed algorithm provides best results for all 100 trails which in result provide the minimum standard deviation compare to other algorithms. While analyzing the success rate, the proposed SAGWO algorithm deliberates the better results compare to other comparison algorithms. Furthermore, the proposed SAGWO requires only minimum number of function evaluations in order to obtain the best result for the test instance 1.

The convergence curves provided in the figure 7.1 clearly shows that the proposed SAGWO algorithm needs only 200 iterations to converge towards the best solution. The best result obtained by running the system repeatedly for 100 times and the best solution obtained for the every trial is presented in figure 7.2 . It clearly shows that in case of repeated simulation SAGWO provides centralized results which are nearby best costs and only minimum number simulations gives the little bias. The

Table 7.2: Comparison of the statistical analysis over 100 trails

Units	**Algorithm**	**Total Generation Cost($/h)**			**Std**	**SR(%)**	**Time(S)**
		Min	**Avg**	**Max**			
10-Unit	OGWO	623.828	624.521	625.412	0.3245	76	10.06
	LFA	623.621	624.012	624.854	0.2475	88	9.42
	CSO	626.754	627.423	628.529	0.5312	86	11.02
	SAGWO	622.863	622.8637	622.8721	0.1234	91	9.04

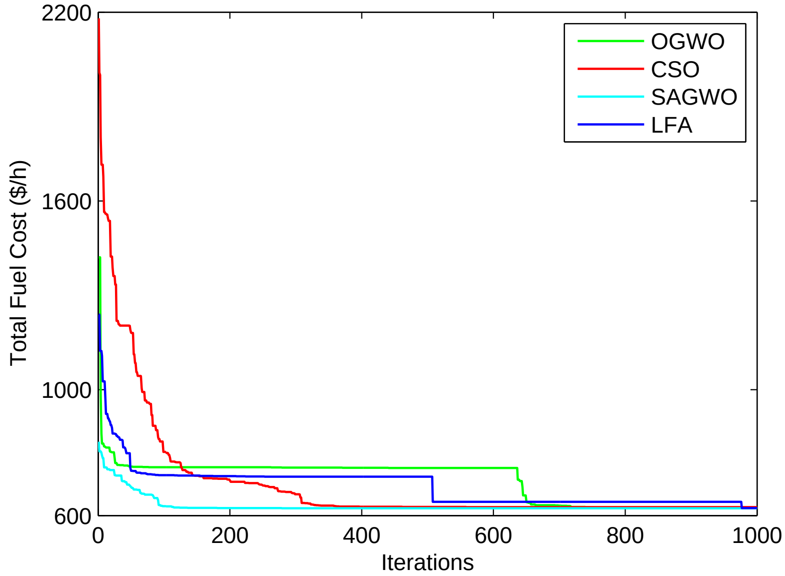

Figure 7.1: The convergence curve of SAGWO in small scale test instance (10-unit systems)

result of obtained SAGWO deliberates that it has the good stability performance and has the collective intelligence to tackle the local optimal solutions.

7.2.3.2 Medium Scale Test Instance-1: 40-unit systems

This section consists of the 40-unit system, which replica of small scale test instance for 4 times respectively. As the size of the generating units increases, more local optima gradually increase for the ELD problem with different fuel options and value-point effects. In order to deal this problem, an efficient optimization algorithm with more searching capability to eradicate the local optima struck as well as the premature convergence problem. The result obtained by SAGWO for 40-unit system with load demand of 10800 MW is provided in the table 7.3. The table 7.3 presents the output power of the each generating units and its total fuel cost for generating the same. The result for 40-unit systems utilizes the differ-

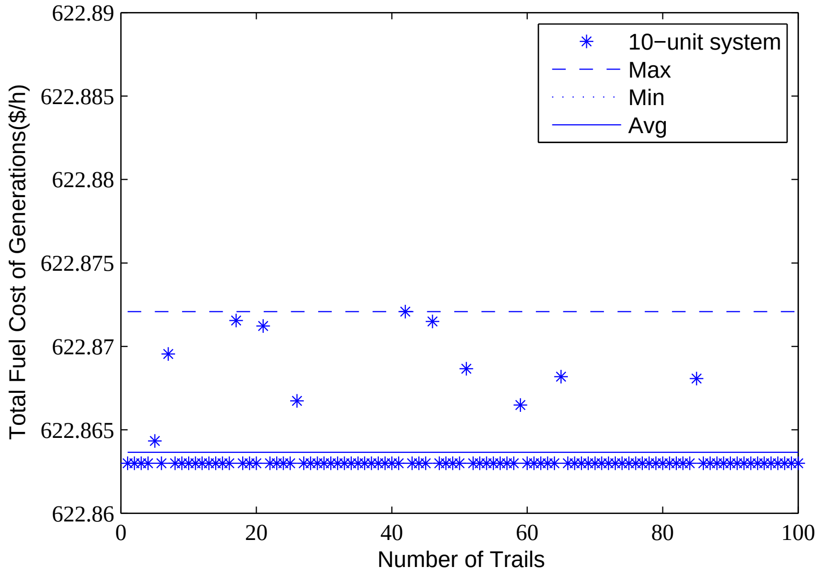

Figure 7.2: Distribution of the minimum costs obtained over 100 trails for small scale test instance (10-unit systems)

ent fuel options which in result provide the minimum fuel cost of 2495.714 $h to generate the desired power output of 10800 MW.

The convergence curves of the different algorithms are given in figure 7.3, it clearly deliberates that the proposed SAGWO provides the faster convergence compare to other algorithms. Initially, the algorithm generates the arbitrary values for every agent and by using the efficient operators the search agent attains the best position which is a result of faster convergence rate. Thus, the proposed algorithm requires only 175 iterations to identify the best solution as well as proves its efficiency and sustains the best place compared to other algorithms. The results of the distribution of the minimum costs obtained for the SAGWO algorithm over 100 trials are given in the figure 7.3. It clearly shows that most of the minimum costs repeatedly obtain the best cost and only few trials exceeds the average costs. This shows that SAGWO has better stability for solving the ELD problems with 40-unit systems with multiple fuel options.

From the table 7.5, the proposed SAGWO algorithm obtains the minimum total fuel costs for 40-unit system is 2495.7145$/h. In addition to that, SAGWO outperforms other algorithms in terms of providing the minimum cost, maximum cost and average cost. The minimum cost is selected from one of the minimum best cost obtained over 100 trails. Though LFA algorithm competes with proposed SAGWO, it fails to converge as much as faster.

Table 7.3: Best power output for 40-unit system with load demand of 10800 MW

Unit	**PO***	**Fuel type**	**Unit**	**PO***	**Fuel type**	**Unit**	**PO***	**Fuel type**	**Unit**	**PO***	**Fuel type**
1	218.0374	2	11	217.5274	2	21	215.3741	2	31	219.3568	2
2	212.2742	1	12	211.3712	1	22	210.6375	1	32	210.2577	1
3	278.5374	1	13	278.6274	1	23	281.4617	1	33	280.5775	1
4	239.6276	3	14	239.3826	3	24	240.9354	3	34	240.2456	1
5	280.1228	1	15	280.7341	1	25	282.6723	1	35	275.2471	1
6	238.5159	3	16	238.5247	3	26	239.7234	3	36	239.8347	3
7	289.7357	1	17	288.6852	1	27	288.3712	1	37	285.0324	1
8	240.3842	3	18	238.5375	3	28	238.3845	3	38	241.0733	3
9	430.5043	3	19	432.4374	3	29	434.2795	3	39	428.2175	3
10	274.3826	1	20	273.6371	1	30	272.1567	1	40	274.5721	1
	Total power output:			10801 MW			**Total cost:**		2495.71$/h		

PO* - Power Output,

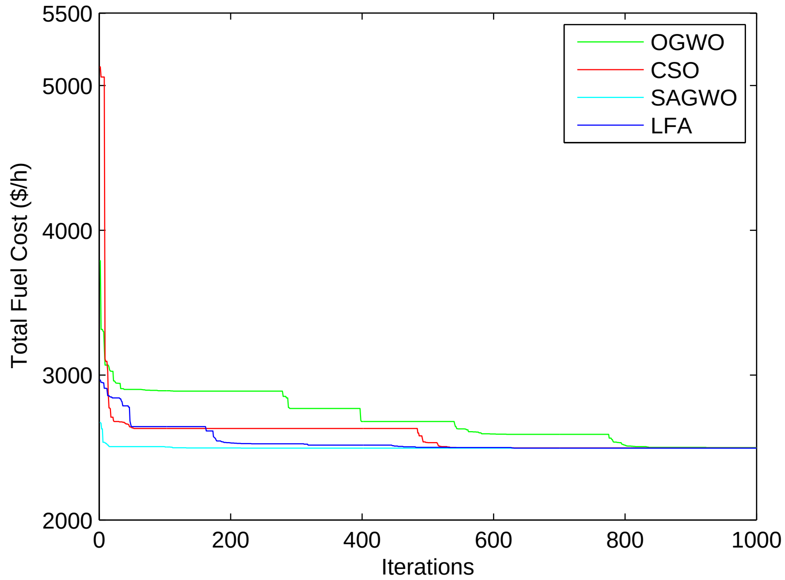

Figure 7.3: The convergence curve of SAGWO in Medium test instance-1 (40-unit systems)

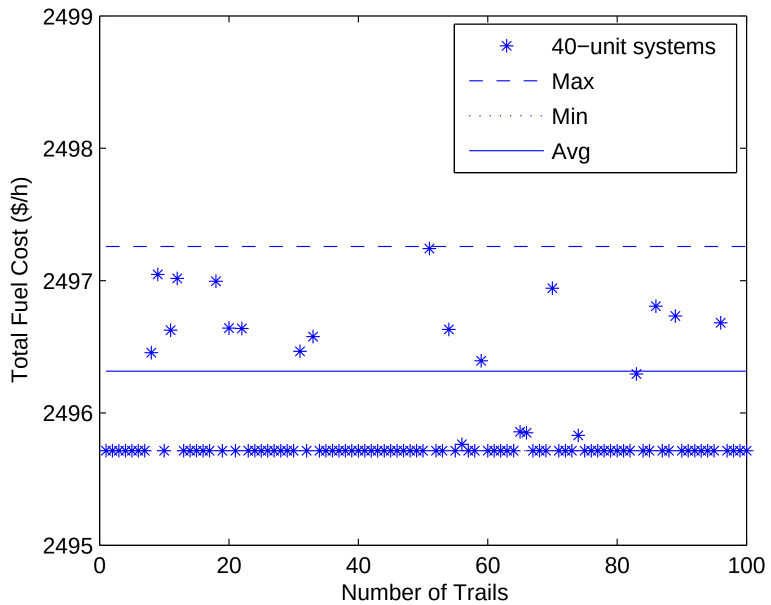

Figure 7.4: Distribution of the minimum costs obtained over 100 trails for Medium test instance-1 (40-unit systems)

While analyzing the success rate of the SAGWO provides the repeated best solution and achieves the better success rate compared to other algorithms. LFA algorithm provides best solution for 82 trails whereas SAGWO provides best solution for 86 trails. In addition to that, the number of function evaluations processed for the proposed algorithm is lesser among the others except the LFA algorithm because the number of function evaluations increases in case of handling the efficient neighborhood search mechanism in SAGWO for particular cases. In terms of computational time LFA algorithm provides the solution within the minimum time period whereas in case of analyzing the total cost, SAGWO provides the minimum which result conveys that the SAGWO is efficient in providing the best solution compare to LFA algorithm.

7.2.3.3 Medium Scale Test Instance-2: 80-unit systems

In case of medium test instance 2 deals with the 80-unit system which are 8 times duplication of the 10-unit systems with multiple fuel options and value-point effects. The load demand fixed for the 80-unit system is 21600 MW and the results of the 80-unit systems are shown in table 7.4. This shows the power output of all the 80 generators and its total fuel cost utilized for operating the system. The total fuel cost of the 80-unit systems obtained by the SAGWO is 4986.474$/h.

The convergence curves of the different algorithms are given in figure 7.5, it clearly deliberates that the proposed SAGWO provides the faster convergence compare to other algorithms. The proposed algorithm utilizes the minimum iterations to identify the best solution as well as proves its efficiency and sustains the best positions compare to other algorithms. The convergence of the SAGWO is much efficient than other meta-heuristics algorithms. The results of the distribution of the minimum costs obtained for the SAGWO algorithm over 100 trails are given in the figure 7.6. It clearly shows that most of the minimum costs repeatedly obtain the best cost and only few trails are exceeds the average costs. This shows that SAGWO has better stability for solving the ELD problems with 40-unit systems with multiple fuel options.

The comparative results of the proposed algorithm and other algorithms for 80-unit systems are provided in table 7.5. From the obtained results, it clearly shows that SAGWO provides better results in terms of minimum cost, average cost and maximum cost. In addition to that, it provides the minimum standard deviation which means the algorithm provides better results for 100 trails of simulations. While analyzing the success rate and number of function evaluations SAGWO provides better success rate as well minimum number of function evaluations for the obtaining the best solution. In terms of computational efficiency, SAGWO obtains the best solution within a reasonable time period.

Table 7.4: Best power output for 80-unit system with load demand of 21600 MW

Unit	**PO***	**Fuel type**	**Unit**	**PO***	**Fuel type**	**Unit**	**PO***	**Fuel type**	**Unit**	**PO***	**Fuel type**
1	217.3475	2	21	213.5578	2	41	216.534	2	61	215.1125	2
2	210.2496	1	22	211.2686	1	42	210.9638	1	62	211.0352	1
3	277.2487	1	23	276.2486	1	43	275.8535	1	63	278.0565	1
4	239.6374	3	24	239.5787	3	44	239.5582	3	64	238.0052	3
5	272.2595	1	25	273.9377	1	45	276.6552	1	65	276.0785	1
6	238.9347	3	26	239.8892	3	46	239.5882	3	66	238.2586	3
7	285.5374	1	27	284.7535	1	47	285.7535	1	67	282.3286	1
8	238.3579	3	28	238.8966	3	48	239.8653	3	68	239.4253	3
9	423.2886	3	29	423.2773	3	49	419.2568	3	69	422.0468	3
10	272.4674	1	30	272.9671	1	50	267.2358	1	70	272.3644	1
11	217.2375	2	31	216.347	2	51	217.8682	2	71	212.6862	2
12	209.3179	1	32	211.5667	1	52	210.2635	1	72	210.2935	1
13	277.8314	1	33	278.934	1	53	278.6538	1	73	278.6728	1
14	238.9347	3	34	238.0347	3	54	239.6524	3	74	238.9242	3
15	273.6278	1	35	275.3175	1	55	272.1353	1	75	272.7572	1
16	238.2582	3	36	239.1238	3	56	240.5835	3	76	239.7956	3
17	285.2528	1	37	283.6347	1	57	284.8652	1	77	283.5847	1
18	238.8852	3	38	238.3013	3	58	238.4245	3	78	238.7521	3
19	423.7578	3	39	423.5317	3	59	423.8056	3	79	420.9356	3
20	269.2558	1	40	269.3427	1	60	489.1254	1	80	268.0457	1
Total power output:				21600 MW			**Total cost:**			4986.474$/h	
PO* - Power Output,											

Table 7.5: Comparison of the statistical analysis over 100 trails for Medium scale Test Instance

Units	Algorithm	Total Generation Cost($/h)			Std	SR(%)	Time(S)
		Min	Avg	Max			
40-Unit	OGWO	2499.986	2500.214	2503.563	0.6451	76	31.54
	LFA	2491.9688	2493.1419	2493.9716	0.5234	88	26.47
	CSO	2495.7888	2496.9341	2497.132	0.2484	91	22.12
	SAGWO	2495.715	**2496.396**	**2497.321**	**0.2643**	**96**	**21.49**
80-Unit	OGWO	4994.642	4996.457	4999.934	0.8912	74	45.31
	LFA	4988.517	4990.6001	4991.9812	0.8416	86	36.29
	CSO	4990.9267	4991.2948	4992.0014	0.3074	90	53.12
	SAGWO	**4986.476**	**4987.341**	**4989.425**	**0.2924**	**95**	**35.08**
160-Unit	OGWO	9986.457	9987.637	9989.0216	0.4678	73	67.12
	LFA	9980.2096	9984.9959	9988.7855	1.9379	87	41.57
	CSO	9984.2438	9984.9163	9986.364	0.40321	90	162.29
	SAGWO	**9984.241**	**9985.675**	**9986.424**	**0.2831**	**93**	**41.03**

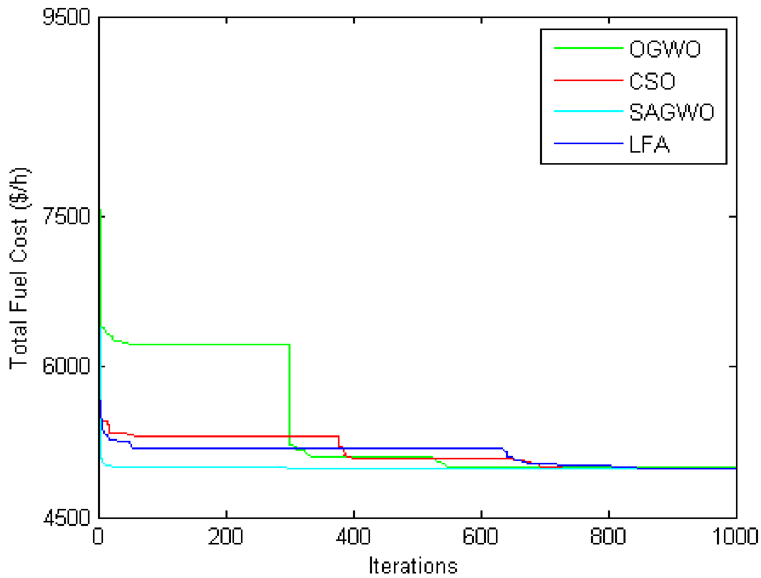

Figure 7.5: The convergence curve of SAGWO in Medium test instance-2 (80-unit systems)

7.2.3.4 Medium Scale Test Instance-3: 160-unit systems

In case of medium scale test instance (160-unit systems) with multiple fuel options and value-point effects is considered. This instance is the 16-times duplication of the 10-unit system and it has to generate the load demand of 43200 MW by utilizing the minimum fuel cost. The 160-unit systems have 160 generators and

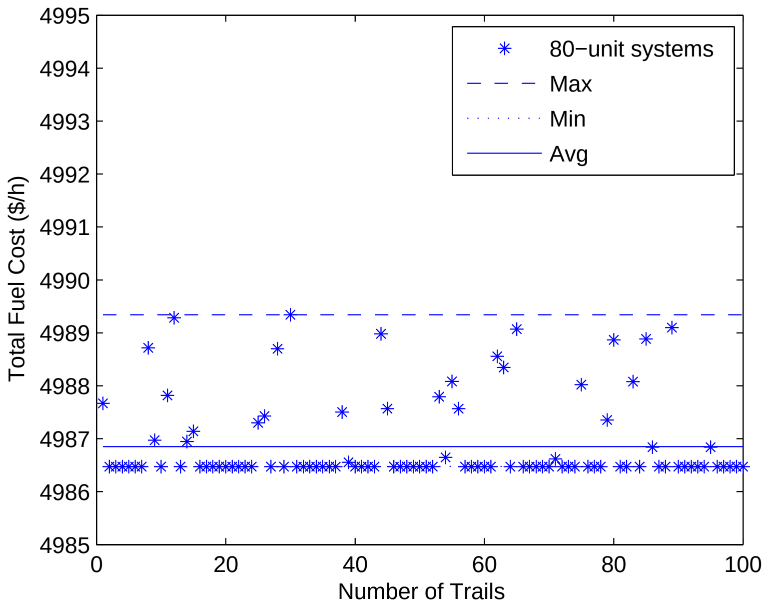

Figure 7.6: Distribution of the minimum costs obtained over 100 trails for Medium test instance-2 (80-unit systems)

each generator produce the different power output by utilizing the different fuel options. The power output obtained by different generators by SAGWO for 160-unit systems is presented in the table 7.6 and 7.7. This shows that the system has attained the desired power output and utilized minimum fuel for generation of power.

The comparative results of the other algorithms with its performance factors are presented in the table 7.5. This shows that the SAGWO is quite efficient in terms of the maximum cost, average cost and standard deviation, while obtained the same minimum cost for maximum number of independent runs. From the comparison we noticed that the SAGWO algorithm is much effective for the ELD problem and deliberates the effectiveness in terms of solution quality and robustness.

In order to understand the much difference between these algorithms more apparently, figure 7.7 illustrates the comparison of convergence rates of the respective algorithm are chosen from one of the 100 trails. From figure 7.7, we notified that the proposed algorithm obtains the similar result as of CSO, while better than the LFA and OGWO approaches. In this case, most of the algorithm identifies the best solution but fails to provide the minimum fuel cost. It observed that the SAGWO algorithm provides the best fuel cost and proves its efficacy and flexibility over solving the 160-unit system.

Table 7.6: Best power output for 160-unit system with load demand of 43200 MW

Unit	**PO***	**Fuel type**	**Unit**	**PO***	**Fuel type**	**Unit**	**PO***	**Fuel type**	**Unit**	**PO***	**Fuel type**
1	215.5582	2	21	213.2557	2	41	216.8527	2	61	220.8821	2
2	209.1275	1	22	213.8427	1	42	215.175	1	62	212.7257	1
3	280.2554	1	23	279.0458	1	43	278.8857	1	63	284.9765	1
4	240.2386	3	24	238.7285	3	44	241.8745	3	64	240.8535	3
5	278.9527	1	25	279.5252	1	45	278.7572	1	65	273.9372	1
6	240.7585	3	26	240.8575	3	46	239.7727	3	66	238.0527	3
7	283.3568	1	27	288.8582	1	47	289.9252	1	67	287.8756	1
8	236.9838	3	28	237.0525	3	48	238.7538	3	68	242.8527	3
9	422.2427	3	29	433.8528	3	49	422.9282	3	69	420.9378	3
10	278.1578	1	30	267.9578	1	50	269.8975	1	70	267.9638	1
11	214.7728	2	31	218.7784	2	51	213.8324	2	71	216.8556	2
12	209.2677	1	32	212.5828	1	52	211.9275	1	72	214.9678	1
13	279.0568	1	33	278.9275	1	53	276.8574	1	73	277.9785	1
14	237.0457	3	34	238.9757	3	54	235.8275	3	74	235.3758	3
15	270.8582	1	35	269.7882	1	55	282.7385	1	75	275.9687	1
16	240.9628	3	36	237.8955	3	56	239.93	3	76	239.9675	3
17	287.5281	1	37	284.8577	1	57	275.6575	1	77	290.9583	1
18	240.2586	3	38	238.8751	3	58	237.8757	3	78	239.9277	3
19	439.5558	3	39	424.8457	3	59	425.8534	3	79	425.5762	3
20	276.9287	1	40	276.9375	1	60	280.2475	1	80	270.7551	1

PO* - Power Output,

Table 7.7: Best power output for 160-unit system with load demand of 43200 MW

Unit	**PO***	**Fuel type**	**Unit**	**PO***	**Fuel type**	**Unit**	**PO***	**Fuel type**	**Unit**	**PO***	**Fuel type**
81	218.5355	2	101	214.6528	2	121	220.8682	2	141	219.2358	2
82	213.5867	1	102	209.6828	1	122	210.2862	1	142	209.5585	1
83	279.8525	1	103	282.6864	1	123	281.9272	1	143	273.8257	1
84	239.7257	3	104	237.9528	3	124	238.8542	3	144	238.8587	3
85	272.9375	1	105	281.4275	1	125	280.8427	1	145	274.8428	1
86	239.8927	3	106	241.2757	3	126	241.8542	3	146	238.8582	3
87	284.8453	1	107	283.9625	1	127	285.6824	1	147	282.8427	1
88	237.8772	3	108	242.2585	3	128	238.9582	3	148	240.8385	3
89	418.8527	3	109	425.9458	3	129	428.2868	3	149	424.9625	3
90	269.2757	1	110	279.2875	1	130	274.9276	1	150	276.228	1
91	217.7572	2	111	219.2857	2	131	220.8572	2	151	216.9251	2
92	210.5275	1	112	212.9458	1	132	214.8885	1	152	212.2866	1
93	274.0457	1	113	281.0285	1	133	274.8247	1	153	280.6588	1
94	239.2556	3	114	238.2867	3	134	239.7568	3	154	424.9527	3
95	283.5352	1	115	269.2025	1	135	273.8427	1	155	276.4585	1
96	236.3545	3	116	238.6952	3	136	238.8271	3	156	241.9428	3
97	283.8686	1	117	288.2481	1	137	291.5425	1	157	292.9284	1
98	239.5854	3	118	239.9642	3	138	237.6585	3	158	237.1782	3
99	423.8682	3	119	417.084	3	139	437.5427	3	159	417.8282	3
100	268.5274	1	120	277.0589	1	140	268.8275	1	160	273.8247	1
Total power output:			43200 MW			**Total cost:**			9984.2438$/h		

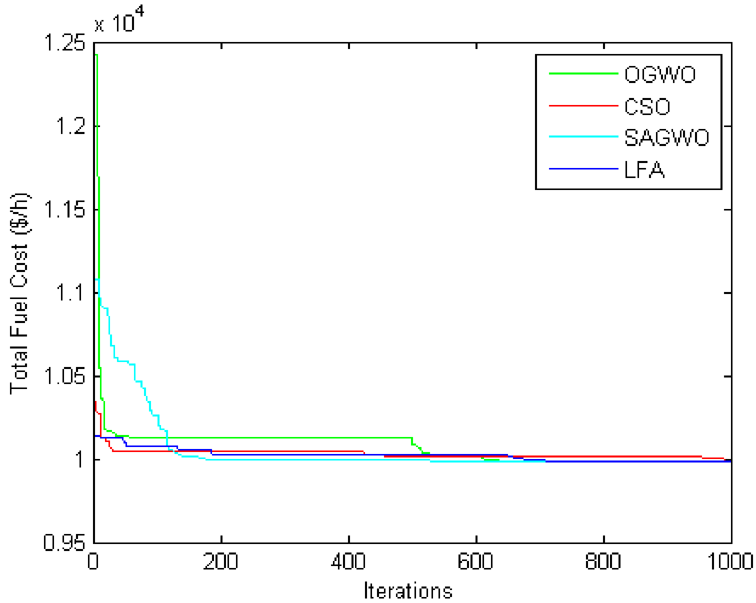

Figure 7.7: The convergence curve of SAGWO in Medium test instance-3 (160-unit systems)

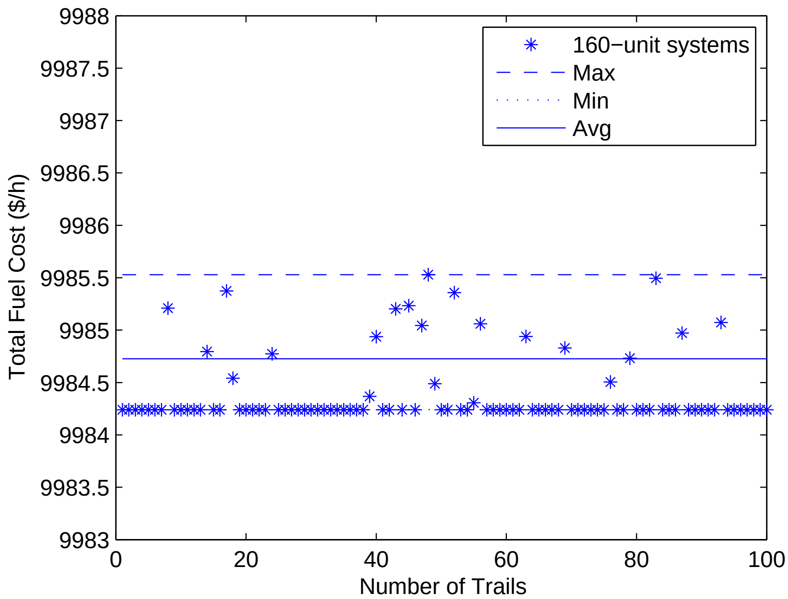

Figure 7.8: Distribution of the minimum costs obtained over 100 trails for Medium test instance-3 (160-unit systems)

Finally, the figure 7.8 depicts distribution of minimum fuel cost obtained over 100 independent runs. This clearly conveys that the proposed algorithm provides repeated best solution for maximum number of independent runs and only for the minimum number trails deviates from the best solution but it obtains the solution within the maximum cost.

7.2.3.5 Large Scale Test Instance-1: 320-unit systems

In order to evaluate the effectiveness of the proposed algorithm two large scale test instance has been considered. First test instance deals with 320-unit systems which is a 32-times duplication of the 10-unit systems. The load demand of this system is set as 86400 MW. As the number of generating units increases gradually then it leads to the multimodal problem which has drastic local optima and it is quite difficult to solve using the traditional algorithms. So far, only few algorithms are used for testing the more than 200-units system with different fuel options. In this point of view, we considered this test instance for evaluating the efficiency of the proposed algorithm. The comparative results of the proposed algorithm and other algorithm is presented in the table 7.8. The result obtained over the 100 independent runs in order to reduce the statistical errors and to understand the efficiency of the proposed one with other approaches. This results shows that the proposed algorithm depicts its efficiency in terms of maximum cost, average cost and standard deviation, while the algorithm provides the best solution for maximum number of trails. Though the LFA algorithm competes with the proposed algorithm in terms of the success rate, while considering the computational efficiency of the proposed algorithm is better compare to other optimization algorithms.

Table 7.8: Comparison of the statistical analysis over 100 trails for large scale Test Instance

Units	Algorithm	Total Generation Cost($/h)			Std	SR(%)	Time(S)
		Min	Avg	Max			
320-Unit	OGWO	19972.64	19976.98	19979.64	3.64	79	506.51
	LFA	19965.95	19969.34	19971.54	1.25	89	375.12
	CSO	19969.93	19972.25	19974.98	0.89	80	412.52
	SAGWO	19964.67	19966.87	19969.56	0.76	92	373.27
640-Unit	OGWO	39968.75	39971.64	39976.97	5.45	74	732.45
	LFA	39957.77	39969.28	39974.66	4.0725	86	596.36
	CSO	39964.06	39968.03	39974.18	1.9075	84	678.41
	SAGWO	39954.54	39964.24	39969.45	0.8374	90	594.54

In order to analyze the efficiency among the other algorithms more visually, the comparison of one arbitrary run chosen over the 100 independent runs of the convergence curves of the appropriate approaches are shown in figure 7.9. From the figure 7.6, we clearly notify that the proposed algorithm obtains the best solution within the 230 iterations whereas other algorithm fails to converge towards the

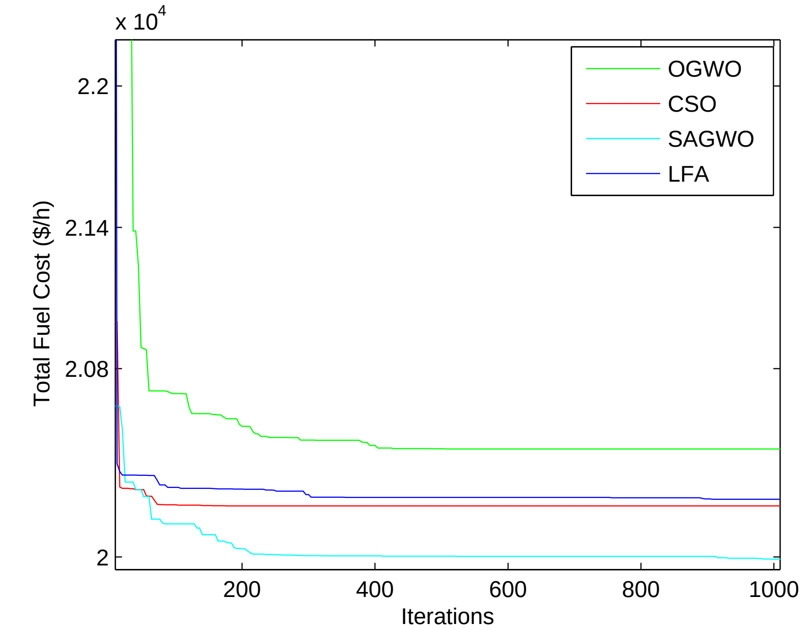

Figure 7.9: The convergence curve of SAGWO in Large scale test instance-1 (320-unit systems)

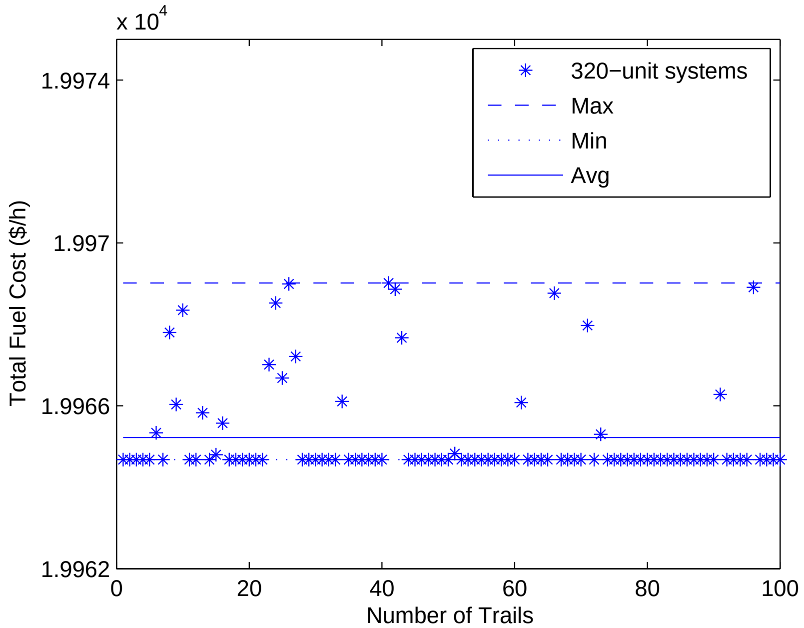

Figure 7.10: Distribution of the minimum costs obtained over 100 trails for large test instance-1 (320-unit systems)

minimum cost. The OGWO algorithm has been stagnated in local optima in about 400 iterations, while the result of LFA and CSO are better than the OGWO, and CSO outperforms the LFA and OGWO. Furthermore, the proposed SAGWO has proven its efficiency on both local search ability and also on accelerated convergence rate of the algorithm.

The distribution of the minimum costs obtained over 100 independent runs for large scale test instance 1 is presented in the figure 7.10. It depicts that the proposed algorithm provides the repeated best solution for all the maximum number of trails and only for few trails the algorithm provides the little bias. This shows that the proposed algorithm has the efficient capability to solve the problem as well as it provides the best solution for maximum trails.

7.2.3.6 Large Scale Test Instance-2: 640-unit systems

This is a 640-unit system considering both the multiple fuel options and value point effects and its a duplication of 10-unit systems for 64-times. It is clear that the sharp increase of units number will provides much more non-convexity and non-linearity into the ELD problem due to high local optima caused by multiple fuel options and value-point effects. The load demand of the 640-unit system is set as 172800 MW.

The comparison results of the each algorithm over 100 independent runs are given in table 7.8. This results shows that the proposed algorithm obtains the better results than all of other approaches in terms of minimum cost, maximum cost, average cost and standard deviation. While analyzing the success rate, proposed algorithm provides the better success rate by providing repeated solution for nearly 86 times among the 100 independent runs, whereas LFA algorithm provides the 82 times repeated solution which is lesser than the proposed algorithm and the SAGWO takes only minimum number function evaluations to obtain the best solution. Finally, while analyzing the computational efficiency in terms of the computational time, the proposed algorithm obtains the best solution within a reasonable time period.

In order to understand the performance of the proposed algorithm among the other algorithm more visually, figure 7.11 provides the convergence curves of the respective algorithm among one of the arbitrary chosen trail from the 100 independent trails. It clearly conveys that the proposed algorithm requires only 300 iterations to obtain the best solution whereas other algorithm stagnates in certain iterations due to soaring of the local optima. Finally, the distribution of the best solution over 100 independent trails is shown in figure 7.12. It depicts that the proposed algorithm provides best cost for maximum number of trails and only for few trails it deviates from the best solution.

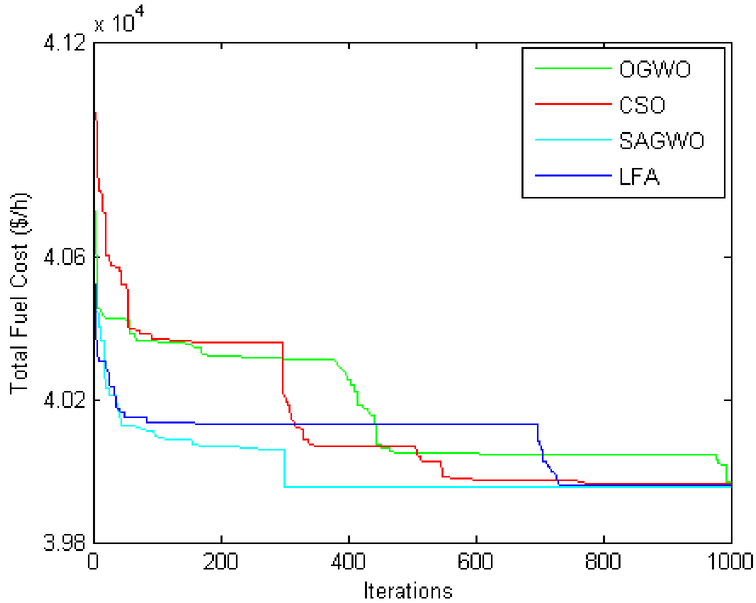

Figure 7.11: The convergence curve of SAGWO in Large scale test instance-2 (620-unit systems)

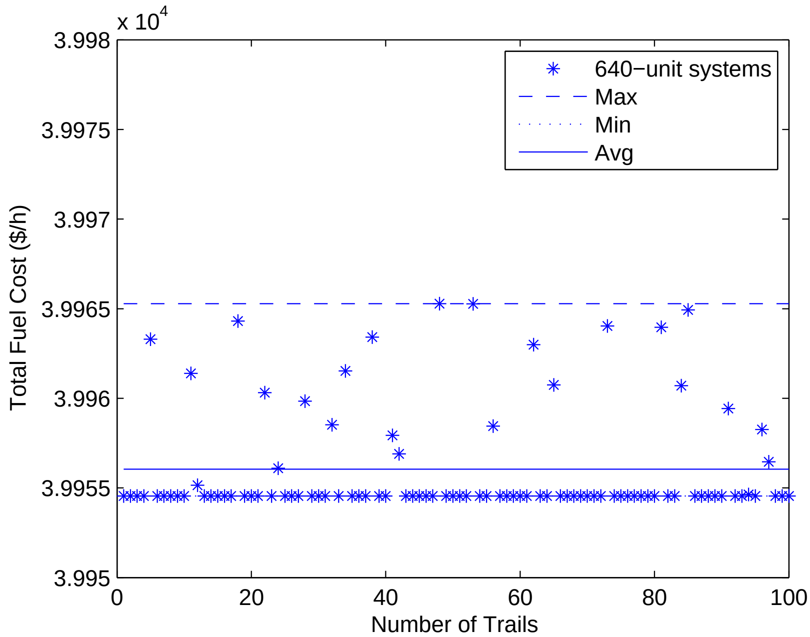

Figure 7.12: Distribution of the minimum costs obtained over 100 trails for large test instance-2 (640-unit systems)

From the above simulations and results for the three test instance cases of ELD problems, it can be summarized that the proposed algorithm has superior global search ability and robustness for nonlinear ELD problems. Furthermore, the proposed SAGWO is more effective than the meta-heuristic algorithms reported in this work as well as it overcomes the issues of the generic GWO and the parameters of the proposed algorithm are adaptive and effective to solve the large scale optimization problems.

7.3 Localization Problem

The definition of a localization system among sensor nodes is a fundamental issue for many applications of wireless sensor networks (WSNs). Because sensor networks may be deployed in inaccessible terrains or disaster relief operations, the position of sensor nodes may not be predetermined. Thus, a localization system is required in order to provide position information to the nodes. The importance of localization information arises from several factors, many of which are related only to WSNs. These factors include the identification and correlation of gathered data, node addressing, management and query of nodes localized in a determined region, evaluation of nodes density and coverage, energy map generation, geographic routing, object tracking, and other geographic algorithms. All of these factors make localization systems a key technology for the development and operation of WSNs. For large scale wireless sensor network, traditional optimization algorithms fail to approximate the accurate positions of the sensor network due to higher error rate on position estimation.

Apparently, a sharp increase of unknown sensor nodes in WSN will introduce more non-linearity and non-convexity into WSN. In addition to that, it increases the high local optima into localization problem which makes the traditional algorithm quite difficult to solve. In order to eradicate these issues an efficient optimization with global search ability and novel mechanism is required to avoid the local optima stagnation and faster convergence towards the optimum solution. The proposed SAGWO has efficient search operators and adaptive parameters which encounter the issues of the generic GWO and redefined in identifying the superior solution. It is clear that SAGWO performs better by sustaining the population of personal best solution from generation to generation.

7.3.1 Problem Formulation

WSN node localization problem formulates using the single hop range based distribution technique to estimate the position of the unknown node coordinates (X, Y) with the aid of anchor nodes (position of known nodes) coordinates (x, y). Anchor nodes are provided with GPS device, so it has the capability to automatically determine its position. Most of the nodes in the WSN are not equipped with GPS due to high cost. To measure coordinates of N unknown nodes, the procedure followed is given below:

Step 1: Randomly Initialize the N unknown nodes and M anchor nodes within the communication range (R). Anchor nodes measure their position and communicate their coordinates to their neighbors. For all iteration, the node which settles at the end termed as reference node and this node will act as anchor node further.

Step 2: Three or more anchor nodes within the communication range of a node is considered as localized node.

Step 3: Neighboring anchor node aids to measure the location of localized node. Distance measurements are distracted due to environmental consideration, to eradicate it Gaussian noise n_i is incorporated with the actual distance d_i.

$$d_i = \sqrt{(X - x_i)^2 + (Y - y_i)^2} \tag{7.4}$$

The node estimates its distance from its anchor as $\widehat{d_i} = [d_i + n_i]$, the noise n_i is generated within the range of $d_i \pm d_i(\frac{P_n}{100})$ where P_n denotes the percentage of noise in estimated distance. Whereas (X, Y) is the coordinates of unknown nodetarget node and (x_i, y_i) is the coordinates of the i^{th} anchor node in the neighborhood.

Step 4: The optimization problem is formulated to minimize the error of localization problem. Each localizable target hub runs SAGWO calculation freely to restrict itself by discovering its position coordinates (x, y). The target capacity of restriction issue can be planned as taken after:

$$f(x, y) = min(\sum_{i=1}^{M} |d_i - \widehat{d_i}|) \tag{7.5}$$

Where, M is the number of anchor nodes within the transmission range (R), of the target node.

Step 5: The localization error is characterized as the interval between the original and evaluated areas of an obscure node which is figured as the mean of square root of interval of evaluated node coordinates (X_i, Y_i) and the original node arranges (ex_i, ey_i) for $i = 1, 2, N_L$ (N_L is the quantity of confined nodes) as demonstrated as follows:

$$E_L = \frac{\sum_{i=M+1}^{N} \sqrt{(X_i - ex_i)^2 + (Y_i - ey_i)^2)}}{(N_L) \times R} \tag{7.6}$$

Step 6: Repeat the step 2 to 5 until all unknown/target nodes get localized or no more nodes can be localized. Localization Error (E_L) and Number of non-Localized nodes (N_{NL}) aid to identify the performance of the localization algorithm. The Number of non-Localized nodes (N_{NL}) is identified based on the difference between the total number nodes and the number of nodes localized. The performance of the algorithm is better if it obtains minimum the value of N_{NL} and E_L.

7.3.2 Experimental Setup

In this section, the point by point assessment of the SAGWO is exhibited. For correlation, two algorithms are utilized thus they are as firstly, the generic GWO and second Modified Bat algorithm (MBA) are used. The algorithmic parameter of SAGWO for analyzing the localization problem is same as given in section 6.4. The deployment area of WSN is considered as the 300m * 300m with varying number of target sensor nodes. The sensor nodes are arbitrarily distributed in the simulation area, whereas the anchor nodes might vary from $\sum_{i=1}^{10} i \times 10$. For SAGWO, the population size is fixed as 100 and the maximum numbers of iterations are fixed as 100 and remaining parameters are chosen as per the section 6.4. For GWO, the parameter linearly decreases in the interval of [2 to 0] and the C parameter linearly increases from 0 to 2. For MBA, the initial values for parameters pulse rate (r) and loudness (A) are assigned as 0.5 and 0.2 ms, respectively.

7.3.3 Performance Metrics

In order to analyze the performance of the SAGWO algorithm, comparisons are made with other algorithms namely GWO and MBA respectively. The performance factors to analyze the performance are as given as follows:

Total Localization Error (E_L)*:* The total localization error is measured after the position of all localizable target nodes N_L is determined. It is computed as the mean of square of the distance between the projected node coordinates (X_i, Y_i)

and the real node coordinates (ex_i, ey_i) which is expressed as follows.

$$E_L = \frac{\sum_{i=M+1}^{N} \sqrt{(X_i - ex_i)^2 + (Y_i - ey_i)^2)}}{(N_L) \times R} \tag{7.7}$$

Number of Non-Localized nodes (N_{NL})*:* Number of non-localized nodes means the number of nodes are not estimated after the successful generations. It is computed as follows:

$$N_{NL} = N - N_L \tag{7.8}$$

Where N is the total number of target nodes or unknown nodes and N_L denotes the number of localized nodes. The minimum values of E_L and N_{NL} make the localization more efficient.

7.3.4 Experimental Analysis

This section discusses about the efficiency of the proposed SAGWO algorithm and compare with GWO, and MBA algorithms by considering the performance metrics that are discussed in section 7.3.3.

7.3.4.1 Node Localization

In this section, each localized node implements three algorithms namely SAGWO, GWO [Mirjalili et al. 2014] and MBA [Goyal and Patterh 2016] to determine the position. In SAGWO, constant parameter of guided search is fixed as 0.5 and the guided probability is chosen as 0.8 which guides the algorithm to converge well and makes the search agent to learn among the surroundings. The Lévy step size is fixed as 0.5 to explore search space and to eradicate the local optima struck. The localization of target nodes using SAGWO is presented in figure 7.13. In GWO, the coefficient parameter a linearly decreases from 2 to 0 and c linearly increases from 0 to 2. The localization of target nodes using GWO is depicted in figure 7.15. In MBA, the initial values for parameters pulse rate (r) and loudness (A) are assigned as 0.5 and 0.2 ms, respectively. The localization of target node using MBA is depicted in figure 7.14. The figure 7.13 - 7.15, shows that anchor nodes, target nodes, and the position estimated by the algorithm SAGWO, GWO and MBA.

Table 7.9: Comparison results of SAGWO, GWO and MBA with varying Anchor nodes

Anchor Node	**Minimum Localization Error**			**Localized Nodes**		
	MBA	**GWO**	**SAGWO**	**MBA**	**GWO**	**SAGWO**
10	0.59	0.52	0.45	230	340	440
20	0.55	0.45	0.39	280	380	500
30	0.51	0.43	0.37	320	420	530
40	0.49	0.41	0.33	370	480	550
50	0.46	0.39	0.31	400	510	580
60	0.43	0.38	0.3	440	580	650
70	0.4	0.37	0.28	480	630	700
80	0.38	0.35	0.26	510	680	753
90	0.36	0.33	0.23	550	700	804
100	0.34	0.3	0.19	560	720	880

Table 7.10: Localized nodes with varying transmission range

Transmission Range	**Localized Nodes**		
	MBA	**GWO**	**SAGWO**
10	388	493	580
15	427	551	660
20	489	607	720
25	537	679	750
30	586	732	790
35	663	783	850
40	758	820	880

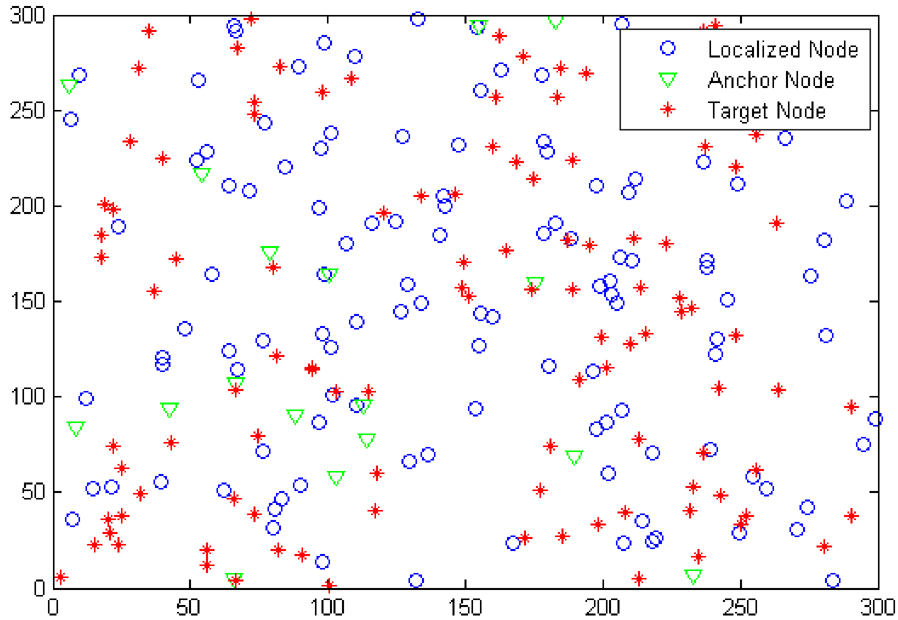

Figure 7.13: Node Localization using GWO

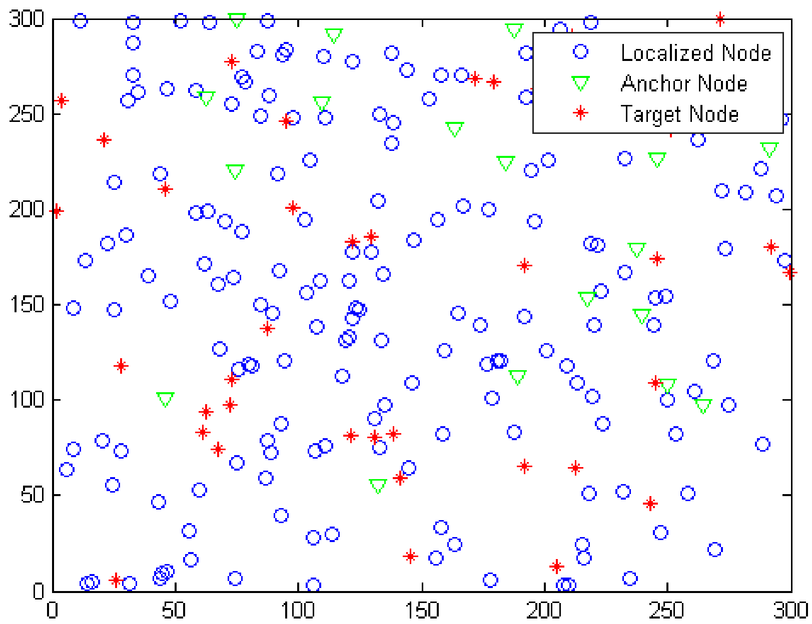

Figure 7.14: Node Localization using SAGWO

The results of SAGWO, MBA and GWO based localization are summarized in table 7.10 shows that all the algorithms utilized here have performed fairly well in localization problem. The outcome of P_n, percentage noise in estimation of distance is apparent on localization accuracy. The percentage noise P_n is used as 2

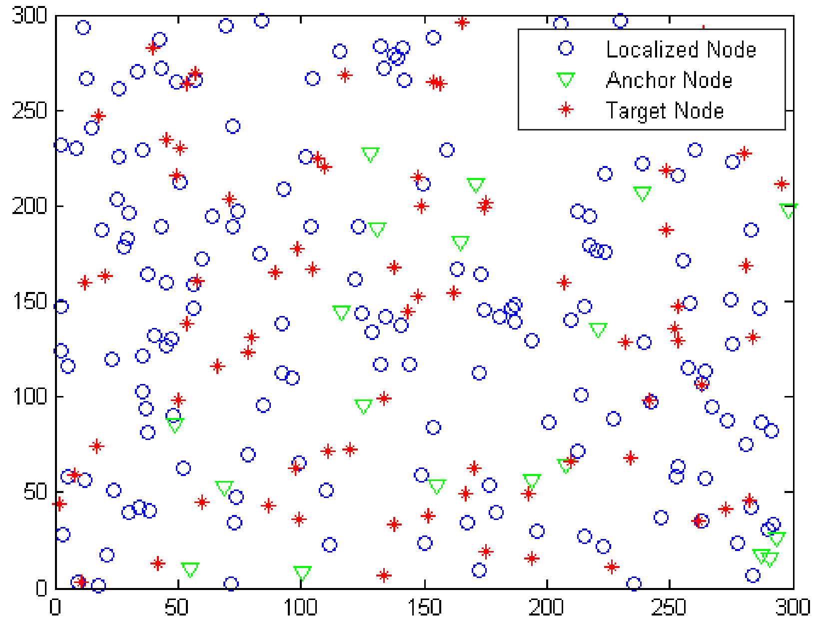

Figure 7.15: Node Localization using MBA

for average localization error for all algorithms. The total localization error E_L and number of non localized nodes $N_N L$ for SAGWO is lesser than that for GWO and MBA, this shows that SAGWO performs well and obtains the optimal solution. Furthermore, the computing time utilized for SAGWO is also significantly less than that for GWO and MBA. Mostly, the localization errors are affected using some of the critical parameters viz., number of anchor nodes, transmission range and number of iterations of optimization algorithms.

7.3.4.2 Effect of Anchor Node Density

Mostly the number of non-localized nodes and localization error are reduced as the number of anchor node increases. Moreover, it is crucial to estimate the position of nodes if adequate number of anchor nodes ($N \geq 3$) is not presented. The density of anchor nodes plays a major role in order to improve the performance of the localization algorithm. A minimum number of anchor nodes provide the reduced efficiency over the localization algorithm. Apparently, the estimation of number of localized nodes mainly depends on the number of anchor nodes for SAGWO, GWO and MBA as given in Figure 7.16-7.17. The figure 7.16 and 7.17 clears that the proposed SAGWO algorithm provides better results compare to generic GWO algorithm and MBA. The results are achieved through varying the number anchor nodes in order to reduce the minimum localization error as well as to improve the number of localized nodes in the WSN.

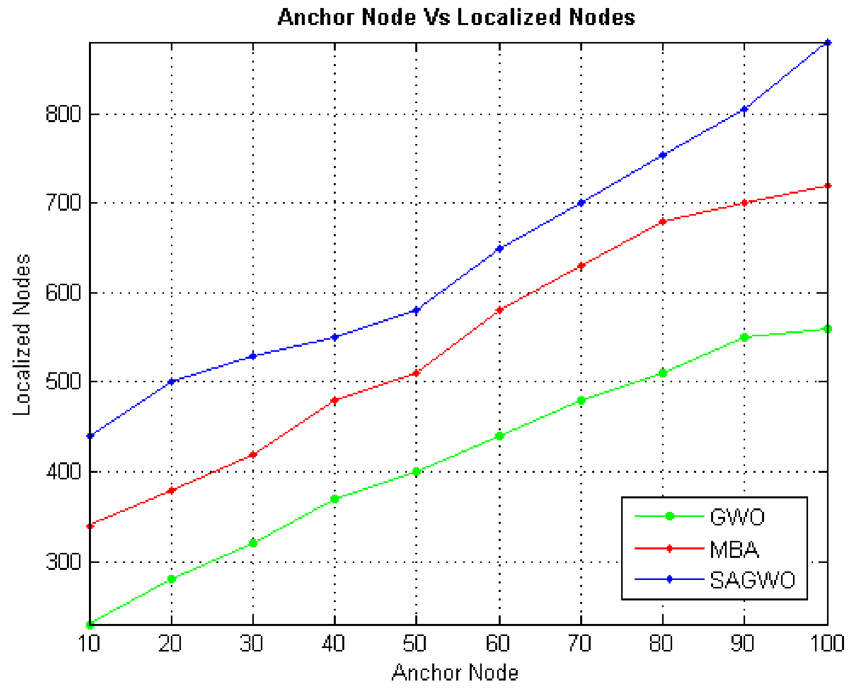

Figure 7.16: Localized Nodes with varying Anchor nodes

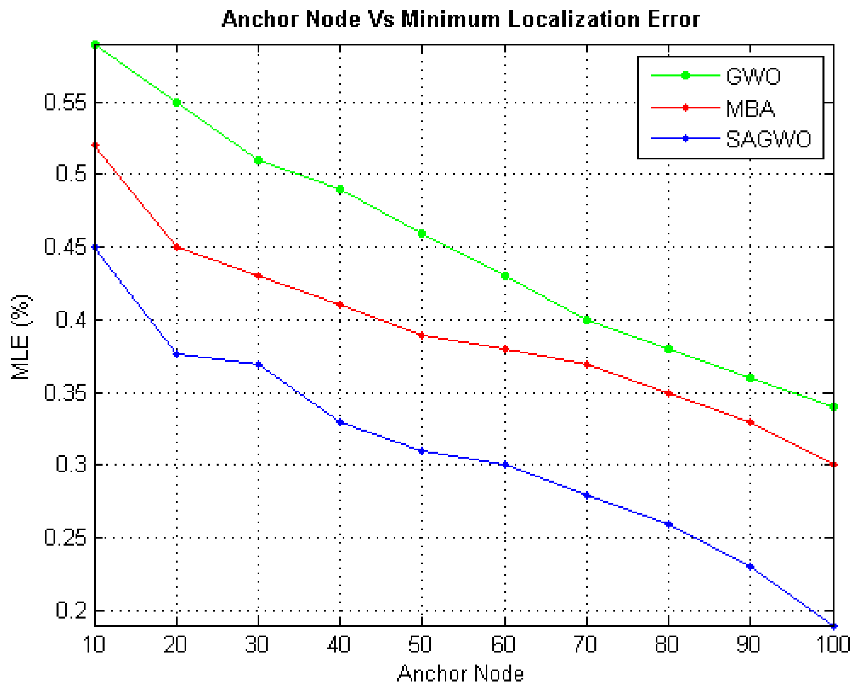

Figure 7.17: Minimum Localization Error with Varying Anchor nodes

7.3.4.3 Effect of Transmission Range

The performance of estimating the localized nodes gradually increases as the increase of transmission range over anchor nodes. The increase in transmission range reflects that anchor node range which in result number of localized nodes are identified are to be high. The performance of localized nodes depends on the transmission range for SAGWO, GWO and MBA are shown in figure 7.18. The result conveys that the minimum transmission range identifies the limited number of localized nodes whereas in maximum transmission range leads to identify the maximum number of localized nodes. In addition to that, the Gaussian noise also plays a vital role that it affects the localization accuracy in practical. As the noise ratio increases then the E_L increases which affect the accuracy of localized nodes. In order to compensate the localization accuracy the entire experiment is performed by fixing the noise as $P_n = 2$. The accuracy of identifying localized nodes increase with the increase in number of generations as given in figure 7.19. As the number of generation increases, then the accuracy of prediction over the localized nodes are to be high.

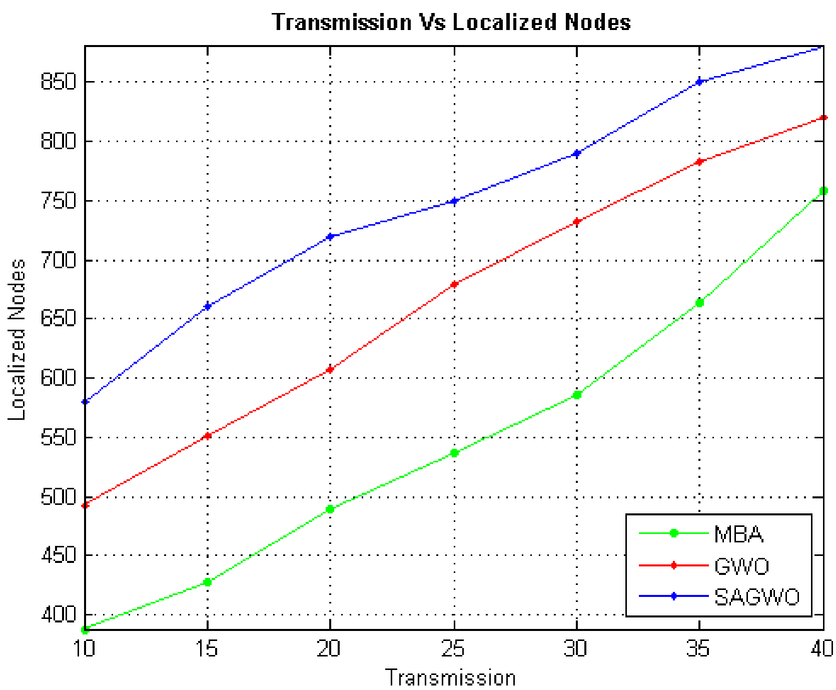

Figure 7.18: Localized Nodes with varying transmission ranges of Anchor nodes

The figure 7.18 clearly shows that the SAGWO identifies maximum number of localized nodes in both the cases of increase in transmission range and number of iterations. The increase in transmission range identifies the as much as possible number of localized nodes compared to less transmission range. The proposed SAGWO estimates the localized nodes better than the other two algorithms. This shows that the proposed algorithm has effective search capability and superior in providing the best solution.

7.3.4.4 Effect of Number of Iterations

The increase in number of iterations used to analyse the accuracy of localizing the sensor nodes. This increase in number of reference nodes which are already estimated or localized by the anchor nodes. Moreover, this process reduces the probability of the flip ambiguity problem. In addition to that, if an unknown node has more number of reference nodes in iteration $k+1$ than in iteration k, then the time for localization nodes is to be increased. The increase in number of iteration might increase the computational time period but it decreases the localization error by identifying the maximum number of localized nodes as shown in figure 7.19. Figure 7.19 apparently shows that localization error reduces with the increase in number of iterations.

From the overall observations notified that the SAGWO based node localization improves the localization accuracy with decrease in localization error. Localization of nodes based on SAGWO, GWO and MBA by varying the number of anchor nodes is provided in the Table 7.7. The SAGWO based localization algorithms offers the minimum localization error as well as identifies more number of localized nodes whereas in GWO approximates the position in minimum computation time but it offers maximum location error. Almost all the optimization algorithm performs well in determining the location of nodes in WSN. SAGWO provides better localization accuracy to determine the position than GWO and MBA in terms of minimized localization error.

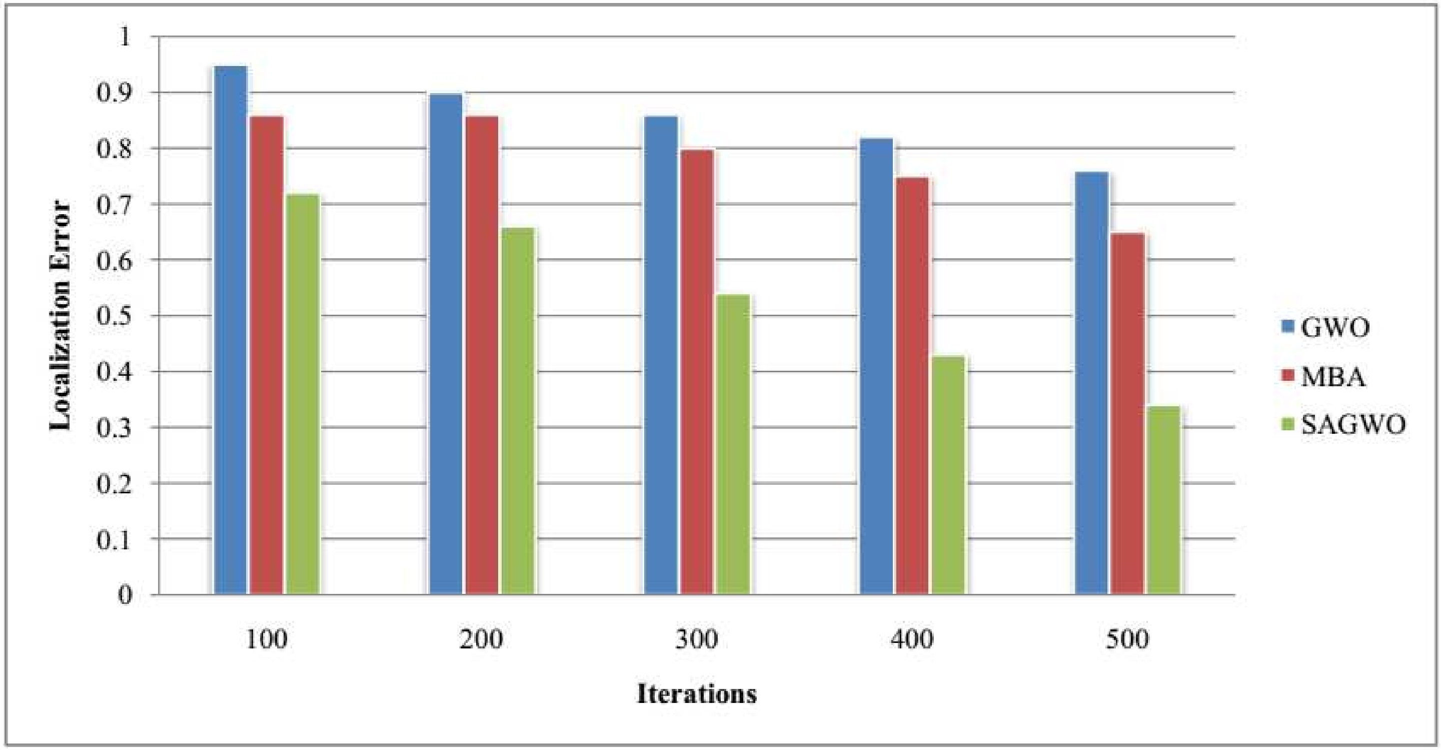

Figure 7.19: Localization Error versus Number of Iterations

7.4 Summary

The assessment carried out in this chapter, is aid to analyze the performance level of SAGWO with other state-of-art meta-heuristic algorithms. To determine the efficiency and stability on handling real-time applications Economic Load Dispatch (ELD) problem and Localization problem in WSN has been chosen for the experiment. The experimental results are analyzed on different test instances namely Small scale, medium scale and large scale test instance along with the performance factors. From the obtained results of ELD, we notify that the proposed approaches shows its superiority over the standard performance metrics as well as the success rate and total number of function evaluations. In case of Localization problem, SAGWO is utilized to estimate the location of unknown sensor nodes using three assessment criterias namely varying the number of anchor density, transmission range and number of iterations. SAGWO outperforms in its superiority over all the test instances with increase in efficiency of generic GWO as well as provides the repeated best solution for n independent runs. Finally, an observation on ELD and Localization problems conveys that SAGWO performs better than state-of-art meta-heuristic algorithms.

Chapter 8

Conclusions and Future Works

This chapter is the concluding part of the thesis and also proposes some suggestions towards which the present work can be further extended. Section 8.1 brings out the overall conclusions of the research work carried out in this thesis and in section 8.2 suggestions regarding the future research directions and possible extensions of the work presented in the thesis are made.

8.1 Conclusion

The key of this research is to model a new variant in GWO that mimic generic functionality of grey wolf and by utilizing the environmental factors that influencing its performance. This phase was embedded into classical GWO in the exploration and exploitation phase and this research was motivated from the issues faced b generic GWO algorithm in the aspect of performance and convergence towards the large scale optimization problems. The main contribution of this thesis are described as follows.

In this thesis, an attempt has been made to solve the large scale optimization problem using self adaptive grey wolf optimization algorithm. They are:

A modified variant of GWO with the name of SAGWO is introduced, it utilizes the fast solving speed of Neighborhood Guided search and Position repulsion mechanism and maintaining the proper balance between exploration and exploitation. The exploitation of SAGWO is carried out using the Neighborhood Guided search mechanism and multi-swarm approach. The exploration of SAGWO is determined using the Position Repulsion mechanism and in addition to that the global best oscillation technique used to oscillate the global best solution to attain global optimum.

A comparative performance study is carried out for the large scale benchmark functions using SAGWO and other state-of-art meta-heuristic algorithms. The comparative study of state-of-art meta-heuristic algorithms for large scale real-time problems viz., Economic Load Dispatch (ELD) and Localization problem with standard performance metrics are carried out.

The Proposed SAGWO algorithm efficiency was determined by testing with the large scale benchmark function, Economic Load Dispatch (ELD) problem and Localization Problem. The first testing was performed on the large scale benchmark function with varying dimensionality and this functions has uni-modal, multi-modal, separable and non-separable characteristics. Existing meta-heuristic algorithms such as mDE-bES, MCSO and JOA were considered for comparing SAGWO approach. Second testing was performed with ELD problem that suits into functionality of SAGWO by determining the optimum solution. For this problem benchmark tests such as small scale test system '10 unit', medium scale test system '40-160 unit' and large scale test system '320 and 640 unit' which were categorized based on the number of generating units. Existing approach such as OGWO, LFA and IODPSO were experimented for comparing SAGWO performance for ELD. Third testing was conducted using Localization problem, which was considered as another real-time problem in which the objective is to locate the unknown sensor by reducing its localization error. Existing approach such as MBA and GWO were also experimented for comparing SAGWO performance for Localization problem. The experimental results were evaluated using the standard statistical performance metrics like best value attained, success rate, standard deviation, etc. The experimentation results clearly conveys that SAGWO could yeild better result than the existing approaches. The overall results of SAGWO notifies the importance of modified variants in large scale benchmark problems.

8.2 Scope for future work

Research is an iterative and continuous procedure. The work presented in the thesis focuses on the solving large scale optimization problem using SAGWO approach. There are several directions in which this work could be expanded. Some of the suggestions for future work in this direction are:

Efficient modification can be done in the implementation of the proposed SAGWO algorithm that may improves its performance on solving other domain state-of-art real-time problems. An implementation of the algorithms may be designed for a small adjustment over the parameters. The implementation of proposed exploitation and exploration mechanism on other state-of-art meta-heuristic algorithms will give promising and better results.

Further, the position adjustment of search agent using three features such as present environment area, Euclidean distance, and secondary search agent structure. More features need to be explored for more accurate prediction of global optimum. In this thesis, a modified variant are proposed for large scale optimization problem and solved for some set of problems. Further, new problems are available and they need to be explored for accurate and fast predictions.

References

[1] Nija Mani, Ashish Mani, et al. Solving combinatorial optimization problems with quantum inspired evolutionary algorithm tuned using a novel heuristic method. *arXiv preprint arXiv:1612.08109*, 2016.

[2] Ali Wagdy Mohamed. Solving large-scale global optimization problems using enhanced adaptive differential evolution algorithm. *Complex & Intelligent Systems*, 3(4):205–231, 2017.

[3] Monirehalsadat Mahmoudi and Xuesong Zhou. Finding optimal solutions for vehicle routing problem with pickup and delivery services with time windows: A dynamic programming approach based on state–space–time network representations. *Transportation Research Part B: Methodological*, 89: 19–42, 2016.

[4] Hong-Fei Teng, Yu Chen, Wei Zeng, Yan-Jun Shi, and Qing-Hua Hu. A dual-system variable-grain cooperative coevolutionary algorithm: satellite-module layout design. *IEEE transactions on evolutionary computation*, 14 (3):438–455, 2010.

[5] Chao Wang and Jinghuai Gao. High-dimensional waveform inversion with cooperative coevolutionary differential evolution algorithm. *IEEE Geoscience and Remote Sensing Letters*, 9(2):297–301, 2012.

[6] Anbo Meng, Hanwu Hu, Hao Yin, Xiangang Peng, and Zhuangzhi Guo. Crisscross optimization algorithm for large-scale dynamic economic dispatch problem with valve-point effects. *Energy*, 93:2175–2190, 2015.

[7] Scott Kirkpatrick, C Daniel Gelatt, and Mario P Vecchi. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983.

[8] John H Holland. Genetic algorithms. *Scientific american*, 267(1):66–73, 1992.

[9] Marco Dorigo and Gianni Di Caro. Ant colony optimization: a new meta-heuristic. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, volume 2, pages 1470–1477. IEEE, 1999.

[10] Rainer Storn and Kenneth Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997.

[11] Riccardo Poli, James Kennedy, and Tim Blackwell. Particle swarm optimization. *Swarm intelligence*, 1(1):33–57, 2007.

[12] Dervis Karaboga and Bahriye Basturk. Artificial bee colony (abc) optimization algorithm for solving constrained optimization problems. In *International fuzzy systems association world congress*, pages 789–798. Springer, 2007.

[13] Xin-She Yang and Suash Deb. Engineering optimisation by cuckoo search. *International Journal of Mathematical Modelling and Numerical Optimisation*, 1(4):330–343, 2010.

[14] Xin-She Yang. Firefly algorithm, levy flights and global optimization. In *Research and development in intelligent systems XXVI*, pages 209–218. Springer, 2010.

[15] Seyedali Mirjalili, Seyed Mohammad Mirjalili, and Andrew Lewis. Grey wolf optimizer. *Advances in engineering software*, 69:46–61, 2014.

[16] Ilhem BoussaïD, Julien Lepagnot, and Patrick Siarry. A survey on optimization metaheuristics. *Information Sciences*, 237:82–117, 2013.

[17] Mohammad Nabi Omidvar, Ming Yang, Yi Mei, Xiaodong Li, and Xin Yao. Dg2: A faster and more accurate differential grouping for large-scale blackbox optimization. *IEEE Transactions on Evolutionary Computation*, 21(6): 929–942, 2017.

[18] Yuan Sun, Michael Kirley, and Saman K Halgamuge. Quantifying variable interactions in continuous optimization problems. *IEEE Transactions on Evolutionary Computation*, 21(2):249–264, 2017.

[19] Ralf Salomon. Re-evaluating genetic algorithm performance under coordinate rotation of benchmark functions. a survey of some theoretical and practical aspects of genetic algorithms. *BioSystems*, 39(3):263–278, 1996.

[20] Ian T Jolliffe. Principal component analysis and factor analysis. In *Principal component analysis*, pages 115–128. Springer, 1986.

[21] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10 (5):1299–1319, 1998.

[22] Aleksandr Filippovich Timan. *Theory of approximation of functions of a real variable*. Elsevier, 2014.

[23] René Descartes. Discourse on method; and, meditations on first philosophy. 1993.

[24] Dimitri P Bertsekas. *Nonlinear programming*. Athena scientific Belmont, 1999.

[25] Janez Brest and Mirjam Sepesy Maučec. Self-adaptive differential evolution algorithm using population size reduction and three strategies. *Soft Computing*, 15(11):2157–2174, 2011.

[26] Zhenyu Yang, Ke Tang, and Xin Yao. Scalability of generalized adaptive differential evolution for large-scale continuous optimization. *Soft Computing*, 15(11):2141–2155, 2011.

[27] Prabhujit Mohapatra, Kedar Nath Das, and Santanu Roy. A modified competitive swarm optimizer for large scale optimization problems. *Applied Soft Computing*, 59:340–362, 2017.

[28] Shi-Zheng Zhao, Jing J Liang, Ponnuthurai N Suganthan, and Mehmet Fatih Tasgetiren. Dynamic multi-swarm particle swarm optimizer with local search for large scale global optimization. In *Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on*, pages 3845–3852. IEEE, 2008.

[29] K Tang, X Li, PN Suganthan, Zh Yang, and Th Weise. Benchmark functions for the cec 2010 special session and competition on large-scale global optimization: Nature inspired computation and applications laboratory, university of science and technology of china. *Applicat. Lab., Univ. Sci. Technol. China, Hefei, China, Tech. Rep*, 2009.

[30] Mohammad Nabi Omidvar, Yi Mei, and Xiaodong Li. Effective decomposition of large-scale separable continuous functions for cooperative co-evolutionary algorithms. In *Evolutionary Computation (CEC), 2014 IEEE Congress on*, pages 1305–1312. IEEE, 2014.

[31] Ran Cheng and Yaochu Jin. A competitive swarm optimizer for large scale optimization. *IEEE transactions on cybernetics*, 45(2):191–204, 2015.

[32] Sedigheh Mahdavi, Mohammad Ebrahim Shiri, and Shahryar Rahnamayan. Cooperative co-evolution with a new decomposition method for large-scale optimization. In *Evolutionary Computation (CEC), 2014 IEEE Congress on*, pages 1285–1292. IEEE, 2014.

[33] Jianchao Fan, Jun Wang, and Min Han. Cooperative coevolution for large-scale optimization based on kernel fuzzy clustering and variable trust region methods. *IEEE Transactions on Fuzzy Systems*, 22(4):829–839, 2014.

[34] Mostafa Z Ali, Noor H Awad, Ponnuthurai N Suganthan, Rehab M Duwairi, and Robert G Reynolds. A novel hybrid cultural algorithms framework with trajectory-based search for global numerical optimization. *Information Sciences*, 334:219–249, 2016.

[35] José García-Nieto and Enrique Alba. Restart particle swarm optimization

with velocity modulation: a scalability test. *Soft Computing*, 15(11):2221–2232, 2011.

[36] Guohua Wu, Rammohan Mallipeddi, Ponnuthurai Nagaratnam Suganthan, Rui Wang, and Huangke Chen. Differential evolution with multi-population based ensemble of mutation strategies. *Information Sciences*, 329:329–345, 2016.

[37] Wei-jie Yu and Jun Zhang. Multi-population differential evolution with adaptive parameter control for global optimization. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 1093–1098. ACM, 2011.

[38] Xiangtao Li, Shijing Ma, and Yunhe Wang. Multi-population based ensemble mutation method for single objective bilevel optimization problem. *IEEE Access*, 4:7262–7274, 2016.

[39] Jiaxing Chen, Guomin Cui, and Huanhuan Duan. Multipopulation differential evolution algorithm based on the opposition-based learning for heat exchanger network synthesis. *Numerical Heat Transfer, Part A: Applications*, 72(2):126–140, 2017.

[40] Antonio LaTorre de la Fuente and José María Peña Sánchez. *A framework for hybrid dynamic evolutionary algorithms: multiple offspring sampling (MOS)*. PhD thesis, Universidad Politécnica de Madrid, 2009.

[41] Laura Lanzarini, Victoria Leza, and Armando De Giusti. Particle swarm optimization with variable population size. In *International Conference on Artificial Intelligence and Soft Computing*, pages 438–449. Springer, 2008.

[42] Sedigheh Mahdavi, Shahryar Rahnamayan, and Kalyanmoy Deb. Opposition based learning: A literature review. *Swarm and Evolutionary Computation*, 2017.

[43] Ivan Zelinka. Somaself-organizing migrating algorithm. In *New optimization techniques in engineering*, pages 167–217. Springer, 2004.

[44] Gaoji Sun, Ruiqing Zhao, and Yanfei Lan. Joint operations algorithm for large-scale global optimization. *Applied Soft Computing*, 38:1025–1039, 2016.

[45] Kalyanmoy Deb et al. Investigating predator-prey algorithms for multi-objective optimization. 2005.

[46] Moumita Pradhan, Provas Kumar Roy, and Tandra Pal. Grey wolf optimization applied to economic load dispatch problems. *International Journal of Electrical Power & Energy Systems*, 83:325–334, 2016.

[47] Sen Zhang and Yongquan Zhou. Grey wolf optimizer based on powell local optimization method for clustering analysis. *Discrete Dynamics in Nature*

and Society, 2015, 2015.

[48] M Madhiarasan and SN Deepa. Elman neural network with modified grey wolf optimizer for enhanced wind speed forecasting. *Circuits and Systems*, 7(10):2975–2995, 2016.

[49] Mehak Kohli and Sankalap Arora. Chaotic grey wolf optimization algorithm for constrained optimization problems. *Journal of Computational Design and Engineering*, 2017.

[50] An Song, Qiang Yang, Wei-Neng Chen, and Jun Zhang. A random-based dynamic grouping strategy for large scale multi-objective optimization. In *Evolutionary Computation (CEC), 2016 IEEE Congress on*, pages 468–475. IEEE, 2016.

[51] Mitchell A Potter. *The design and analysis of a computational model of cooperative coevolution*. PhD thesis, Citeseer, 1997.

[52] Mitchell A Potter and Kenneth A De Jong. A cooperative coevolutionary approach to function optimization. In *International Conference on Parallel Problem Solving from Nature*, pages 249–257. Springer, 1994.

[53] Mitchell A Potter and Kenneth A De Jong. Cooperative coevolution: An architecture for evolving coadapted subcomponents. *Evolutionary computation*, 8(1):1–29, 2000.

[54] Olivier Barrière and Evelyne Lutton. Experimental analysis of a variable size mono-population cooperative-coevolution strategy. In *Nature Inspired Cooperative Strategies for Optimization (NICSO 2008)*, pages 139–152. Springer, 2009.

[55] Xian-Bin Cao, Hong Qiao, and John Keane. A low-cost pedestrian-detection system with a single optical camera. *IEEE Transactions on Intelligent Transportation Systems*, 9(1):58–67, 2008.

[56] Xiao Mingming, Zhang Jun, Cai Kaiquan, Cao Xianbin, and Tang Ke. Cooperative co-evolution with weighted random grouping for large-scale crossing waypoints locating in air route network. In *Tools with Artificial Intelligence (ICTAI), 2011 23rd IEEE International Conference on*, pages 215–222. IEEE, 2011.

[57] Yong Liu, Xin Yao, Qiangfu Zhao, and Tetsuya Higuchi. Scaling up fast evolutionary programming with cooperative coevolution. In *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on*, volume 2, pages 1101–1108. Ieee, 2001.

[58] Frans Van den Bergh and Andries Petrus Engelbrecht. A cooperative approach to particle swarm optimization. *IEEE transactions on evolutionary*

computation, 8(3):225–239, 2004.

[59] Mohammed El-Abd. A cooperative approach to the artificial bee colony algorithm. In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pages 1–5. IEEE, 2010.

[60] Zhenyu Yang, Ke Tang, and Xin Yao. Large scale evolutionary optimization using cooperative coevolution. *Information Sciences*, 178(15):2985–2999, 2008.

[61] Zhenyu Yang, Ke Tang, and Xin Yao. Self-adaptive differential evolution with neighborhood search. In *Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on*, pages 1110–1116. IEEE, 2008.

[62] Zhenyu Yang, Ke Tang, and Xin Yao. Multilevel cooperative coevolution for large scale optimization. In *Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on*, pages 1663–1670. IEEE, 2008.

[63] Mohammad Nabi Omidvar, Xiaodong Li, Zhenyu Yang, and Xin Yao. Cooperative co-evolution for large scale optimization through more frequent random grouping. In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pages 1–8. IEEE, 2010.

[64] Zhenyu Yang, Jingqiao Zhang, Ke Tang, Xin Yao, and Arthur C Sanderson. An adaptive coevolutionary differential evolution algorithm for large-scale optimization. In *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*, pages 102–109. IEEE, 2009.

[65] Yuanfang Ren and Yan Wu. An efficient algorithm for high-dimensional function optimization. *Soft Computing*, 17(6):995–1004, 2013.

[66] Tapabrata Ray and Xin Yao. A cooperative coevolutionary algorithm with correlation based adaptive variable partitioning. In *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*, pages 983–989. IEEE, 2009.

[67] Mohammad Nabi Omidvar, Xiaodong Li, and Xin Yao. Smart use of computational resources based on contribution for cooperative co-evolutionary algorithms. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 1115–1122. ACM, 2011.

[68] Karsten Weicker and Nicole Weicker. On the improvement of coevolutionary optimizers by learning variable interdependencies. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, volume 3, pages 1627–1632. IEEE, 1999.

[69] Wenxiang Chen, Thomas Weise, Zhenyu Yang, and Ke Tang. Large-scale

global optimization using cooperative coevolution with variable interaction learning. In *International Conference on Parallel Problem Solving from Nature*, pages 300–309. Springer, 2010.

[70] Cara MacNish and Xin Yao. Direction matters in high-dimensional optimisation. In *Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on*, pages 2372–2379. IEEE, 2008.

[71] Abdel-Rahman Hedar and Ahmed Fouad Ali. Genetic algorithm with population partitioning and space reduction for high dimensional problems. In *Computer Engineering & Systems, 2009. ICCES 2009. International Conference on*, pages 151–156. IEEE, 2009.

[72] Abdel-Rahman Hedar, Bun Theang Ong, and Masao Fukushima. Genetic algorithms with automatic accelerated termination. *Department of Applied Mathematics and Physics, Kyoto University, Tech. Rep*, 2, 2007.

[73] Wei Chu, Xiaogang Gao, and Soroosh Sorooshian. A new evolutionary search strategy for global optimization of high-dimensional problems. *Information Sciences*, 181(22):4909–4927, 2011.

[74] Yu Wang and Bin Li. A restart univariate estimation of distribution algorithm: sampling under mixed gaussian and lévy probability distribution. In *Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on*, pages 3917–3924. IEEE, 2008.

[75] Weishan Dong, Tianshi Chen, Peter Tino, and Xin Yao. Scaling up estimation of distribution algorithms for continuous optimization. *IEEE Transactions on Evolutionary Computation*, 17(6):797–822, 2013.

[76] Yu Wang, Jin Huang, Wei Shan Dong, Jun Chi Yan, Chun Hua Tian, Min Li, and Wen Ting Mo. Two-stage based ensemble optimization framework for large-scale global optimization. *European Journal of Operational Research*, 228(2):308–320, 2013.

[77] Yu Wang and Bin Li. Two-stage based ensemble optimization for large-scale global optimization. In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pages 1–8. IEEE, 2010.

[78] Ivan Nesic, Aleksandar Rakicevic, Ana Poledica, and Bratislav Petrovic. Gaussian variable neighborhood search and enhanced genetic algorithm for continuous optimization. *Electronic Notes in Discrete Mathematics*, 39:273–280, 2012.

[79] Viviana Cocco Mariani, Luiz Guilherme Justi Luvizotto, Fábio Alessandro

Guerra, and Leandro dos Santos Coelho. A hybrid shuffled complex evolution approach based on differential evolution for unconstrained optimization. *Applied Mathematics and Computation*, 217(12):5822–5829, 2011.

[80] Jingqiao Zhang and Arthur C Sanderson. Jade: adaptive differential evolution with optional external archive. *IEEE Transactions on evolutionary computation*, 13(5):945–958, 2009.

[81] Tetsuyuki Takahama and Setsuko Sakai. Large scale optimization by differential evolution with landscape modality detection and a diversity archive. In *Evolutionary Computation (CEC), 2012 IEEE Congress on*, pages 1–8. IEEE, 2012.

[82] Chao Wang and J-H Gao. A differential evolution algorithm with cooperative coevolutionary selection operation for high-dimensional optimization. *Optimization Letters*, 8(2):477–492, 2014.

[83] Chao Wang and Jinghuai Gao. A new differential evolution algorithm with cooperative coevolutionary selection operator for waveform inversion. In *Geoscience and Remote Sensing Symposium (IGARSS), 2010 IEEE International*, pages 688–690. IEEE, 2010.

[84] Qinqin Fan and Xuefeng Yan. Self-adaptive differential evolution algorithm with discrete mutation control parameters. *Expert Systems with Applications*, 42(3):1551–1572, 2015.

[85] Yong Wang, Zixing Cai, and Qingfu Zhang. Enhancing the search ability of differential evolution through orthogonal crossover. *Information Sciences*, 185(1):153–177, 2012.

[86] MS Morley and C Tricarico. A comparison of population-based optimization techniques for water distribution system expansion and operation. *Procedia Engineering*, 89:13–20, 2014.

[87] Chun Chen and Lin-Yu Tseng. An improved version of the multiple trajectory search for real value multi-objective optimization problems. *Engineering Optimization*, 46(10):1430–1445, 2014.

[88] Carlos García-Martínez, Francisco J Rodríguez, and Manuel Lozano. Role differentiation and malleable mating for differential evolution: an analysis on large-scale optimisation. *Soft Computing*, 15(11):2109–2126, 2011.

[89] Antony W Iorio and Xiaodong Li. Improving the performance and scalability of differential evolution. In *Asia-Pacific Conference on Simulated Evolution and Learning*, pages 131–140. Springer, 2008.

[90] Jonatan Gomez and Elizabeth Leon. A coevolutionary chromosome encoding scheme for high dimensional search spaces. In *Evolutionary Computation*

(CEC), 2010 IEEE Congress on, pages 1–8. IEEE, 2010.

[91] YOU Xuemei. Differential evolution with a new mutation operator for solving high dimensional continuous optimization problems. *Journal of Computational Information Systems*, 6(9):3033–3039, 2010.

[92] Xiuqin Pan, Yue Zhao, and Xiaona Xu. Adaptive differential evolution with local search for solving large-scale optimization problems. *JOURNAL OF INFORMATION &COMPUTATIONAL SCIENCE*, 9(2):489–496, 2012.

[93] Shahryar Rahnamayan, Hamid R Tizhoosh, and Magdy MA Salama. Opposition-based differential evolution algorithms. In *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, pages 2010–2017. IEEE, 2006.

[94] XZ Gao, X Wang, SJ Ovaska, and K Zenger. A hybrid optimization method of harmony search and opposition-based learning. *Engineering Optimization*, 44(8):895–914, 2012.

[95] Shahryar Rahnamayan and G Gary Wang. Center-based sampling for population-based algorithms. In *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*, pages 933–938. IEEE, 2009.

[96] Sedigheh Mahdavi, Mohammad Ebrahim Shiri, and Shahryar Rahnamayan. Metaheuristics in large-scale global continues optimization: A survey. *Information Sciences*, 295:407–428, 2015.

[97] Borhan Kazimipour, Xiaodong Li, and A Kai Qin. A review of population initialization techniques for evolutionary algorithms. In *Evolutionary Computation (CEC), 2014 IEEE Congress on*, pages 2585–2592. IEEE, 2014.

[98] Sheng-Ta Hsieh, Tsung-Ying Sun, Chan-Cheng Liu, and Shang-Jeng Tsai. Solving large scale global optimization using improved particle swarm optimizer. In *Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on*, pages 1777–1784. IEEE, 2008.

[99] Yuhua Li, Zhi-Hui Zhan, Shujin Lin, Jun Zhang, and Xiaonan Luo. Competitive and cooperative particle swarm optimization with information sharing mechanism for global optimization problems. *Information Sciences*, 293:370–382, 2015.

[100] Manuel Lozano, Daniel Molina, and Francisco Herrera. Editorial scalability of evolutionary algorithms and other metaheuristics for large-scale continuous optimization problems. *Soft Computing*, 15(11):2085–2087, 2011.

[101] Takeshi Korenaga, Toshiharu Hatanaka, and Katsuji Uosaki. Performance improvement of particle swarm optimization for high-dimensional function optimization. In *Evolutionary Computation, 2007. CEC 2007. IEEE*

Congress on, pages 3288–3293. IEEE, 2007.

[102] Bahriye Akay and Dervis Karaboga. A modified artificial bee colony algorithm for real-parameter optimization. *Information Sciences*, 192:120–142, 2012.

[103] Ahmad Nickabadi, Mohammad Mehdi Ebadzadeh, and Reza Safabakhsh. A novel particle swarm optimization algorithm with adaptive inertia weight. *Applied Soft Computing*, 11(4):3658–3670, 2011.

[104] Stephen Chen and Yenny Noa Vargas. Improving the performance of particle swarms through dimension reductionsa case study with locust swarms. In *Evolutionary computation (CEC), 2010 IEEE congress on*, pages 1–8. IEEE, 2010.

[105] Wei Chu, Xiaogang Gao, and Soroosh Sorooshian. Handling boundary constraints for particle swarm optimization in high-dimensional search space. *Information Sciences*, 181(20):4569–4581, 2011.

[106] Iztok Fister Jr, Xin-She Yang, Iztok Fister, Janez Brest, and Dušan Fister. A brief review of nature-inspired algorithms for optimization. *arXiv preprint arXiv:1307.4186*, 2013.

[107] Haluk Gozde and M Cengiz Taplamacioglu. Automatic generation control application with craziness based particle swarm optimization in a thermal power system. *International Journal of Electrical Power & Energy Systems*, 33(1):8–16, 2011.

[108] Hui Wang, Hui Sun, Changhe Li, Shahryar Rahnamayan, and Jeng-Shyang Pan. Diversity enhanced particle swarm optimization with neighborhood search. *Information Sciences*, 223:119–135, 2013.

[109] A Rezaee Jordehi. Enhanced leader pso (elpso): a new pso variant for solving global optimisation problems. *Applied Soft Computing*, 26:401–417, 2015.

[110] Narinder Singh and SB Singh. Hybrid algorithm of particle swarm optimization and grey wolf optimizer for improving convergence performance. *Journal of Applied Mathematics*, 2017, 2017.

[111] Narinder Singh and SB Singh. A modified mean gray wolf optimization approach for benchmark and biomedical problems. *Evolutionary Bioinformatics*, 13:1176934317729413, 2017.

[112] Himani Joshi and Sankalap Arora. Enhanced grey wolf optimization algorithm for global optimization. *Fundamenta Informaticae*, 153(3):235–264, 2017.

[113] Pranjali Diwan and MR Khan. Energy efficient communication for wsns using grey-wolf optimization algorithm. *International Journal Of Engineering*

And Computer Science, 5(12), 2016.

[114] Nitin Mittal, Urvinder Singh, and Balwinder Singh Sohi. Modified grey wolf optimizer for global engineering optimization. *Applied Computational Intelligence and Soft Computing*, 2016:8, 2016.

[115] Dipayan Guha, Provas Kumar Roy, and Subrata Banerjee. Grey wolf optimization to solve load frequency control of an interconnected power system: Gwo used to solve lfc problem. *International Journal of Energy Optimization and Engineering (IJEOE)*, 5(4):62–83, 2016.

[116] Sen Zhang, Yongquan Zhou, Zhiming Li, and Wei Pan. Grey wolf optimizer for unmanned combat aerial vehicle path planning. *Advances in Engineering Software*, 99:121–136, 2016.

[117] Bo Yang, Xiaoshun Zhang, Tao Yu, Hongchun Shu, and Zihao Fang. Grouped grey wolf optimizer for maximum power point tracking of doubly-fed induction generator based wind turbine. *Energy conversion and management*, 133:427–443, 2017.

[118] Abdul Kayom Md Khairuzzaman and Saurabh Chaudhury. Multilevel thresholding using grey wolf optimizer for image segmentation. *Expert Systems with Applications*, 86:64–76, 2017.

[119] Lo Ing Wong, MH Sulaiman, MR Mohamed, and Mee Song Hong. Grey wolf optimizer for solving economic dispatch problems. In *Power and Energy (PECon), 2014 IEEE International Conference on*, pages 150–154. IEEE, 2014.

[120] Attia A El-Fergany and Hany M Hasanien. Single and multi-objective optimal power flow using grey wolf optimizer and differential evolution algorithms. *Electric Power Components and Systems*, 43(13):1548–1559, 2015.

[121] Eid Emary, Hossam M Zawbaa, Crina Grosan, and Abul Ella Hassenian. Feature subset selection approach by gray-wolf optimization. In *Afro-European Conference for Industrial Advancement*, pages 1–13. Springer, 2015.

[122] Vikram Kumar Kamboj, SK Bath, and JS Dhillon. Solution of non-convex economic load dispatch problem using grey wolf optimizer. *Neural Computing and Applications*, 27(5):1301–1316, 2016.

[123] Mingjing Wang, Huiling Chen, Huaizhong Li, Zhennao Cai, Xuehua Zhao, Changfei Tong, Jun Li, and Xin Xu. Grey wolf optimization evolving kernel extreme learning machine: Application to bankruptcy prediction. *Engineering Applications of Artificial Intelligence*, 63:54–68, 2017.

[124] Mostafa Z Ali, Noor H Awad, and Ponnuthurai N Suganthan. Multi-

population differential evolution with balanced ensemble of mutation strategies for large-scale global optimization. *Applied Soft Computing*, 33:304–327, 2015.

[125] PJ García Nieto, E García-Gonzalo, F Sánchez Lasheras, and Francisco Javier de Cos Juez. Hybrid pso–svm-based method for forecasting of the remaining useful life for aircraft engines and evaluation of its reliability. *Reliability Engineering & System Safety*, 138:219–231, 2015.

[126] R Jensi and G Wiselin Jiji. An enhanced particle swarm optimization with levy flight for global optimization. *Applied Soft Computing*, 43:248–261, 2016.

[127] Hüseyin Haklı and Harun Uğuz. A novel particle swarm optimization algorithm with levy flight. *Applied Soft Computing*, 23:333–345, 2014.

[128] Shi-Zheng Zhao, Ponnuthurai Nagaratnam Suganthan, and Swagatam Das. Self-adaptive differential evolution with multi-trajectory search for large-scale optimization. *Soft Computing*, 15(11):2175–2185, 2011.

[129] MF Zaman, Saber M Elsayed, Tapabrata Ray, and Ruhul A Sarker. Evolutionary algorithms for dynamic economic dispatch problems. *IEEE Transactions on Power Systems*, 31(2):1486–1495, 2016.

[130] Anbo Meng, Jinbei Li, and Hao Yin. An efficient crisscross optimization solution to large-scale non-convex economic load dispatch with multiple fuel types and valve-point effects. *Energy*, 113:1147–1161, 2016.

[131] Subham Sahoo, K Mahesh Dash, RC Prusty, and AK Barisal. Comparative analysis of optimal load dispatch through evolutionary algorithms. *Ain Shams Engineering Journal*, 6(1):107–120, 2015.

[132] Sonia Goyal and Manjeet Singh Patterh. Wireless sensor network localization based on cuckoo search algorithm. *Wireless personal communications*, 79(1):223–234, 2014.

[133] Miloud Mihoubi, Abdellatif Rahmoun, Pascal Lorenz, and Noureddine Lasla. An effective bat algorithm for node localization in distributed wireless sensor network. *Security and Privacy*, 1(1):e7, 2018.

[134] N Jayakumar, S Subramanian, S Ganesan, and EB Elanchezhian. Grey wolf optimization for combined heat and power dispatch with cogeneration systems. *International Journal of Electrical Power & Energy Systems*, 74: 252–264, 2016.

[135] Seyedali Mirjalili. How effective is the grey wolf optimizer in training multi-layer perceptrons. *Applied Intelligence*, 43(1):150–161, 2015.

[136] MR Shakarami and I Faraji Davoudkhani. Wide-area power system stabilizer

design based on grey wolf optimization algorithm considering the time delay. *Electric Power Systems Research*, 133:149–159, 2016.

[137] Ben Niu, Yunlong Zhu, Xiaoxian He, and Henry Wu. Mcpso: A multi-swarm cooperative particle swarm optimizer. *Applied Mathematics and computation*, 185(2):1050–1062, 2007.

[138] Guochu Chen and Jinshou Yu. Two sub-swarms particle swarm optimization algorithm. In *International Conference on Natural Computation*, pages 515–524. Springer, 2005.

[139] Zhao-Hua Liu, Jing Zhang, Shao-Wu Zhou, Xiao-Hua Li, and Kan Liu. Coevolutionary particle swarm optimization using ais and its application in multiparameter estimation of pmsm. *IEEE Transactions on cybernetics*, 43 (6):1921–1935, 2013.

[140] Bilal H Abed-alguni and Faisal Alkhateeb. Novel selection schemes for cuckoo search. *Arabian Journal for Science and Engineering*, 42(8):3635–3654, 2017.

[141] Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989.

[142] Qingna Li and Dong-Hui Li. A class of derivative-free methods for large-scale nonlinear monotone equations. *IMA Journal of Numerical Analysis*, 31(4):1625–1635, 2011.

[143] Deyu Tang, Jin Yang, Shoubin Dong, and Zhen Liu. A lévy flight-based shuffled frog-leaping algorithm and its applications for continuous optimization problems. *Applied Soft Computing*, 49:641–662, 2016.

[144] Renato A Krohling and Eduardo Mendel. Bare bones particle swarm optimization with gaussian or cauchy jumps. In *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*, pages 3285–3291. IEEE, 2009.

[145] F Herrera, M Lozano, and D Molina. Test suite for the special issue of soft computing on scalability of evolutionary algorithms and other metaheuristics for large scale continuous optimization problems. *Last accessed: July*, 2010.

[146] Ke Tang, Xın Yáo, Ponnuthurai Nagaratnam Suganthan, Cara MacNish, Ying-Ping Chen, Chih-Ming Chen, and Zhenyu Yang. Benchmark functions for the cec2008 special session and competition on large scale global optimization. *Nature Inspired Computation and Applications Laboratory, USTC, China*, 24, 2007.

[147] Jiajia Chen, Jiehui Zheng, Peterzhe Wu, Luliang Zhang, and Qinghua Wu. Dynamic particle swarm optimizer with escaping prey for solving constrained non-convex and piecewise optimization problems. *Expert Systems with Ap-*

plications, 86:208–223, 2017.

[148] C-L Chiang. Genetic-based algorithm for power economic load dispatch. *IET generation, transmission & distribution*, 1(2):261–269, 2007.

[149] Chao-Lung Chiang. Improved genetic algorithm for power economic dispatch of units with valve-point effects and multiple fuels. *IEEE transactions on power systems*, 20(4):1690–1699, 2005.

[150] Jong-Bae Park, Yun-Won Jeong, Joong-Rin Shin, and Kwang Y Lee. An improved particle swarm optimization for nonconvex economic dispatch problems. *IEEE Transactions on Power Systems*, 25(1):156–166, 2010.

[151] Moumita Pradhan, Provas Kumar Roy, and Tandra Pal. Oppositional based grey wolf optimization algorithm for economic dispatch problem of power system. *Ain Shams Engineering Journal*, 2017.

[152] Mostafa Kheshti, Xiaoning Kang, Zhaohong Bie, Zaibin Jiao, and Xiuli Wang. An effective lightning flash algorithm solution to large scale nonconvex economic dispatch with valve-point and multiple fuel options on generation units. *Energy*, 129:1–15, 2017.

[153] Sonia Goyal and Manjeet Singh Patterh. Modified bat algorithm for localization of wireless sensor network. *Wireless Personal Communications*, 86 (2):657–670, 2016.

List of Publications

International Journal

1. R. Rajakumar, J. Amudhavel, P. Dhavachelvan, and T. Vengattaraman, *"GWO-LPWSN: Grey Wolf Optimization Algorithm for Node Localization Problem in Wireless Sensor Networks"*, Journal of Computer Networks and Communications, Volume 2017, Article ID 7348141, pp. 1-10.

2. R. Rajakumar, D Rajaguru, J Amudhavel, P Dhavachelvan and T Vengattaraman, *"An Energy Efficient Cluster Formation in Wireless Sensor Network using Grey Wolf Optimization"*, Int. J. Ad Hoc and Ubiquitous Computing, 2017. (Accepted). (SCI Indexed)

3. R. Rajakumar, T. Kalaipriyan, D. Rajaguru, P Dhavachelvan, T Vengattaraman, *"A Comprehensive Survey on Particle Swarm Optimization (PSO) and Its Variants to Classical Benchmark Functions"*, Int. J. Ad Hoc and Ubiquitous Computing, 2017. (Accepted). (SCI Indexed)

4. R. Rajakumar, P. Dhavachelvan, T. Vengattaraman, *"Accelerated Grey Wolf Optimization for Global Optimization Problems"*, Arabian Journal of Science and Engineering, springer, 2017. (Communicated) (SCI Indexed).

International Conference

1. Rajakumar, R., P. Dhavachelvan, and T. Vengattaraman, *"A survey on nature inspired meta-heuristic algorithms with its domain specifications"*, In Communication and Electronics Systems (ICCES), International Conference on, pp. 1-6. IEEE, 2016.

2. Raja, B., R. Rajakumar, P. Dhavachelvan, and T. Vengattaraman, *"A survey on classification of network structure routing protocols in wireless sensor networks"*, In Computational Intelligence and Computing Research (ICCIC), 2016 IEEE International Conference on, pp. 1-5. IEEE, 2016.

3. Thirugnanasambandam KalaiPriyan, Rajakumar Ramalingam, Vengattaraman Thirumal, and Sujatha Pothula, *"Greedy based population seeding technique in Ant Colony Optimization algorithm"*, In Communication and Electronics Systems (ICCES), International Conference on, pp. 1-4. IEEE, 2016.