Human Learning Principles Inspired Particle Swarm Optimization Algorithms

A thesis submitted to
School of Computer Science and Engineering
Nanyang Technological University

by

MUHAMMAD RIZWAN TANWEER

in fulfilment of the requirements
for the degree of Doctor of Philosophy

2016

Acknowledgements

In the name of the Almighty Creator who is the most beneficent and the most merciful. All my prayers are for Him, who has given me the opportunity, courage and strength for performing this work and completing this Thesis.

First of all, I would like to express my deepest gratitude to my supervisor, Prof. Suresh Sundaram for his expert, sincere and valuable guidance and encouragement extended to me for the past years. His critical thinking and analytical insights of the problem has provided me the understanding of the research work. His support and encouragement has helped me greatly in the accomplishment of this research work. He has made my stay at Nanyang Technological University a truly valuable experience. I am also thankful to Prof. Narasimhan Sundararajan for his expert and valuable guidance, time and encouragement stretched towards me. His expert advice and assistance were a valuable contribution to this research work.

I would also like to dedicate this special thanks to my family and friends, especially my parents and my wife, who have always been there for me. This research work would not have been possible without their love, unceasing encouragement and support. I would like to thank my group mates Shirin Dora, Ankit Kumar Das, Abdullah Al-Dujaili and Kartick Subramanian for their help throughout my research work. Their company, support and keen interest in providing a helping hand in my research work are highly valuable for me.

I express my sincere thanks to the Computational Intelligence Laboratory for providing the research facilities. I also acknowledge the School of Computer Science and

Engineering, Nanyang Technological University for the financial support and this precious opportunity of study. Last but not the least, I would like to record my gratitude to all, who directly or indirectly, have lent their helping hand in my work.

Contents

Abstract

These days, the nature of global optimization problems, especially for engineering systems has become extremely complex. For these types of problems, nature inspired search based algorithms are providing much better solutions compared with other classical optimization methods. Among them, the Particle Swarm Optimization (PSO) algorithm has been mostly preferred due to its simplicity and ability to provide better solutions. PSO algorithm simulates the social behaviour of a bird swarm in search of food where the birds are modelled as particles. The limitations associated with PSO have been extensively studied and different modifications, variations and refinements to PSO have been proposed in the literature for enhancing its performance. The idea of utilizing intelligent swarms motivated towards exploring human cognitive learning principles for PSO. As discussed in learning psychology, human beings are known to be intelligent and have good social cognizance. Therefore, any optimization technique employing human-like learning strategies should prove to be more effective. This thesis addresses the use of human learning principles inspired strategies for the PSO algorithm. The major contributions of the thesis are:

- Self-Regulating Particle Swarm Optimization (SRPSO) algorithm.
- Dynamic Mentoring and Self-Regulation based Particle Swarm Optimization (DMeSR-PSO) algorithm.
- Directionally Driven Self-Regulating Particle Swarm Optimization (DD-SRPSO) algorithm.
- Incorporation of a constraint handling mechanism in the structure of the DD-SRPSO algorithm.

The Self-Regulating Particle Swarm Optimization (SRPSO) algorithm is inspired from the human self-learning principles. SRPSO utilizes self-regulation and self-perception based learning strategies to achieve an enhanced exploration and a better exploitation. The self-regulated inertia weights are employed only for the best particle whereas all the other particles perform search employing self-perception of the global best search

direction. The perception is dynamically changed in every iteration for intelligent exploitation. The effect of human learning strategies on the particles has been studied using CEC2005 benchmark problems and the performance has been compared with the state-of-the-art PSO variants. The results clearly indicate that SRPSO converge faster closer to the global optimum with a 95% confidence level.

Further, human beings utilize multiple information processing strategies during the learning process and collaborate with each other for better decision making. Integration of socially shared information processing will further enhance the performance. Therefore, a new algorithm referred to as Dynamic Mentoring and Self-Regulation based Particle Swarm Optimization (DMeSR-PSO) algorithm has been proposed incorporating the concept of mentoring together with the self-regulation. Here, the particles are divided into three groups consisting of mentors, mentees and independent learners. The elite particles are grouped as mentors to guide the poorly performing particles of the mentees group. The independent learners perform search using self-perception based learning strategy of the SRPSO algorithm. Tested on both the unimodal and multimodal CEC2005 benchmark problems the DMeSR-PSO has shown improved convergence than the SRPSO algorithm. Further, the robustness of the algorithm has been tested on CEC2013 problems and eight real-world optimization problems from CEC2011. The results indicate that DMeSR-PSO is significantly better than other PSO variants and other population based optimization algorithms with a 95% confidence level, yielding an effective optimization algorithm for real-world applications.

Both SRPSO and DMeSR-PSO are rotationally variant algorithms and therefore the performances have not been significant on the rotated problems. To overcome this, a directionally updated and rotationally invariant SRPSO algorithm has also been developed named as Directionally Driven SRPSO (DD-SRPSO) algorithm. Here, the poorly performing particles are equipped with complete social perception guidance. Other particles are randomly selected to perform search either by using self-perception based learning strategy of SRPSO or by applying a rotation invariant strategy. The performance of DD-SRPSO tested on rotated problems from CEC2013 proves that DD-SRPSO is significantly better than SRPSO. Its performance, compared with other algorithms on CEC2013 benchmark problems clearly indicates that DD-SRPSO is significantly better than selected algorithms on a wide range of problems.

Further, a new constraint handling mechanism has been incorporated in the DDSRPSO structure referred to as DD-SRPSO with constraint handing mechanism (DDSRPSO-CHM). Next, the application of DD-SRPSO-CHM in optimizing multi-stage launch vehicle configuration has been studied. In a multi-stage launch vehicle configuration, the multiple objectives are converted into a single objective with constraints and these are efficiently handled by DD-SRPSO-CHM. Comparative analysis on the problem suggests that DD-SRPSO is converging faster towards the solution.

By incorporating human-like behaviour in the PSO algorithm, the developed variants have shown a faster convergence closer to the optima over a diverse set of problems indicating that the algorithms are potential choice for complex real-world applications. In the future, the algorithm will be extended for solving multi-objective optimization problems. The equality constraint handling mechanism has already been implemented in the DD-SRPSO algorithm which can be further extended for the inequality constraints. Furthermore, more human learning strategies can be explored for performance enhancement.

List of Figures

List of Tables

List of Abbreviations

ABC .... Artificial Bee Colony

ACO .... Ant Colony Optimzation

AIS .... Artificial Immune System

APO .... Artificial Physics Optimization

BBO .... Bio-geography Based Optimization

AIWPSO .... Adaptive Inertia Weight Particle Swarm Optimization

BBPSO .... Bare Bones Particle Swarm Optimization

CD .... Critical Difference

CEC2005 .... Congress on Evolutionary computation 2005

CEC2011 .... Congress on Evolutionary computation 2011

CEC2013 .... Congress on Evolutionary computation 2013

CLPSO .... Comprehensive Learning Particle Swarm Optimization

CMA-ES .... Covariance Matrix Adaptation Evolution Strategy

CMAES-RIS .... CMAES with Re-sampled Inheritance Search

CSO .... Competitive Swarm Optimizer

D .... Dimension

DD-SRPSO .... Directionally Driven Self-Regulating Particle Swarm Optimization

DD-SRPSO-CHM .... DD-SRPSO with Constraint Handling Mechanism

DE .... Differential Evolution

DMeSR-PSO .... Dynamic Mentoring and Self-Regulation based Particle Swarm Optimization

DMSPSO .... Dynamic Multi-Swarm Particle Swarm Optimizer

EAs .... Evolutionary Algorithms

EDA .... Estimation of Distribution Algorithm

ELPSO .... Exemplar based Learning Particle Swarm Optimization Algorithm

FEA .... Flexible Evolutionary Algorithm

FIPS .... Fully informed particle swarm

FM .... Frequency-Modulated

Func. .... Functions

GA .... Genetic Algorithm

GA-MPC .... Genetic Algorithm with Multi-Parent Crossover

GA-TPC .... Genetic Algorithm with Three Parent Crossover

GbSA .... Galaxy based Search Algorithm

G-DE .... Guided Differential Evolution

GPSO .... Global Particle Swarm Optimization

GSA .... Gravitational Search Algorithm

Hy-RCGA .... Hybrid Real-Coded Genetic Algorithm

LJ .... Lennard-Jones

LPSO .... Local Particle Swarm Optimization

M .... Transformation Matrix

NP-hard .... Non Polynomial hard

POP-CMA-ES .... CMA-ES with increasing population

PSMBA .... Power Spectrum optimization inspired by Magnetic Bacteria Algorithm

PSO .... Particle Swarm Optimization

RC-EA .... Real-Coded Evolutionary Algorithm

RCMA .... Real-Coded Memetic Algorithm

SaDE .... Self Adaptive Differential Evolution

SAMODE .... Self-Adaptive Multi-Operator DE

SI .... Swarm Intelligence

SL-PSO .... Social Learning Particle Swarm Optimization Algorithm

SPSO-2011 .... Standard Particle Swarm Optimization 2011 version

SRPSO .... Self-Regulating Particle Swarm Optimization

TNEP .... Transmission Network Expansion Planning

TPLPSO .... Teaching and Peer Learning Particle Swarm Optimization

UPSO .... Unified Particle Swarm Optimization

χPSO .... Constriction factor Particle Swarm Optimization

List of Symbols

$\overrightarrow{X_i}$ .... Particle's Position vector

$\overrightarrow{V_i}$ .... Particle's velocity vector

ω .... Inertia weight

c_1 & c_2 .... Acceleration constants

r_1 & r_2 .... Uniformly distributed Random Numbers

P_{id} .... Personal Best position

P_{gd} .... Global Best position

χ .... Constriction factor

$\otimes$ .... Kronecher product symbol representing point-wise vector multiplication

λ .... Threshold in DMeSR-PSO algorithm

Φ .... Constraints in Constriction Factor PSO

$\overrightarrow{p_m}$ .... Convergence point in the search space

P_c .... Learning Probability

μ .... Mean

σ .... Standard Deviation

p_{id}^{se} .... Particle's perception of the self-cognition

p_{id}^{so} .... Particle's perception of the social-cognition

η .... Constant to control the rate of acceleration

$\Delta\omega$ .... Rate of change of inertia weight

z .... Shifted global optimum

V_{max} .... Maximum Velocity

O .... Big O Notation used for complexity

S_f .... Euclidian distance

S_{ed} .... Fitness difference

M_r .... Mentor Group

M_e .... Mentee Group

I_n .... Independent Learners Group

λ_1 .... Threshold value 1 set as 5%

λ_2 .... Threshold value 2 set as 10%

λ_3 .... Threshold value 3 set as 90%

λ_4 .... Threshold value 4 set as 50%

C_{se} .... Self-confidence signal

C_{Me} .... Mentor-confidence signal

S_{Me} .... Selection of Mentor

β_1 .... Constant to control partial self-cognition

β_2 .... Constant to control partial social-cognition

lP_{kd}^t .... Median of personal best of three particles from the elite group

ε .... Group fraction for DD-SRPSO

β .... Threshold value for DD-SRPSO strategy selection

$\overrightarrow{P_i}$ .... Particle's personal best position vector

$\overrightarrow{L_i}$ .... Local best position vector

$\overrightarrow{G_i}$ .... Centre of Gravity vector

H_i .... Hyper-sphere notation

m .... Total number of equality constraints

p .... Total number of inequality constraints

v_{orbit} .... Orbital velocity

v_{boost} .... Gain in velocity

v_{drag} .... Loss in velocity due to drag force

v_g .... Loss in velocity due to drag Gravity

ΔV .... Total velocity

$m_{payload}$ .... Final mass payload

M_{oi} .... Total vehicle weight at stage i

I_{sp} .... Specific impulse

Chapter 1

Introduction

In recent years, optimization problems have become extremely complex and obtaining the best results has become challenging. In many engineering design applications, several decisions are required to be made for achieving the ultimate goal of either minimizing the required efforts or maximizing the desired benefits. The process of finding the most favourable solutions is often challenging. The area of development of optimization algorithm has been extensively researched and numerous numerical optimization techniques have been developed for achieving the desired optimal solutions [2].

1.1 Background and Motivation

Most of the real world problems are black-box optimization problems. Black-box optimization refers to solving such problems that does not have an analytical formulation. In these problems, the gradient information and problem domain specific knowledge are unknown beforehand for the objective function. Here, the objective function is a mathematical expression that defines the objectives of the problem. The ultimate goal for the optimization algorithm is to evaluate the objectives in such a way that the function is minimized or maximized. The efficiency of the optimization algorithm for evaluating a black-box optimization problem is usually measured in terms of number of evaluations to reach a target value. These problems can be either constrained or unconstrained depending on their characteristics. For these problems, a qualitative measure is not available to identify the maxima or minima of the problem and hence the objective functions are optimized in a black-box interface. The only available solution is to execute the optimization algorithm on the problem and take the performance as a feedback for further experimentation. As indicated in [3], the classical optimization algorithms have failed to solve such problems and therefore this motivated the researchers to develop evolutionary algorithms. Over the past few decades, many powerful population based meta-heuristic optimization algorithms have been developed. These algorithms have effectively provided the optimal/near optimal solutions to those problems where the traditional optimization algorithms have failed miserably [3].

In the past decade, researchers initially introduced evolution inspired algorithms comprehensively discussed in [4, 5] such as Genetic Algorithm (GA) [6], Differential Evolution (DE) [7], Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [8] etc. Another research area introduced various optimization algorithms inspired from the principles of physics such as Artificial Physics Optimization (APO) [9], Galaxy based Search Algorithm (GbSA) [10], Power Spectrum optimization inspired by Magnetic Bacteria Algorithm (PSMBA) [11], Gravitational Search Algorithm (GSA) [12] etc. Further, researchers took inspiration from nature and developed several nature inspired optimization algorithms. These algorithms include Particle Swarm Optimization (PSO) [13], Ant Colony Optimization (ACO) [14], Artificial Bee Colony (ABC) [15], Bio-geography Based Optimization (BBO) [16], Artificial Immune System (AIS) [17] etc. These algorithms are commonly referred to as Swarm Intelligence (SI) algorithms because these algorithms mimic the collective behaviour of a swarm of any creature for optimizing a given problem. Among these research areas, the most notable one in providing good solutions has been the nature inspired optimization algorithms [18].

All these algorithms have either replicated the evolution inspired mechanisms like reproduction, crossover, mutation and selection in their design and implementation or the collective behaviour of a swarm of creatures. Another way of looking into these problems is by utilizing the human learning characteristics. It has been shown in human learning psychology that humans are effective learners, better planners and good decision makers [1]. Humans are superior to other creatures as they possess self-regulation characteristics to learn from the environment in a better way and plan accordingly. This has motivated us to explore the use of human learning principles for achieving better optimized solutions.

Studies in human learning psychology have suggested that humans adopt a self-regulatory mechanism in their learning process for effective learning [19]. Self-regulation controls the learning process through proper planning and selection of appropriate learning strategies. It enables the learners to perform the task with higher order thinking such as planning, monitoring and evaluating. The self-regulatory mechanism in human learning has been presented as a general framework in the Nelson and Naren's model [1]. This framework consists of a meta-cognitive component and a cognitive component interlinked with each other through flow of information. This framework describes the learning processes such that human beings regulate their cognitive processes and improve their cognitive skills through development of new strategies and evaluation of acquired information. The pictorial representation of this framework is shown in figure 1.1. There are two layers in the model, the meta-cognitive layer consisting of the dynamic model of the lower layer called the cognitive component. The layers communicate with each other using the flow of information signals. Based on this information, the meta-cognitive component decides the future states of the cognitive component and generates appropriate control actions.

Further, learning in human beings is not limited to self-cognizance since humans perform learning using multiple information processing strategies [1]. Collaborative and cooperative socially shared information helps in attaining the maximum gain from the environment. In human psychology, a conscientious human is described as one who effectively utilizes his self and social cognizance, regulates his strategies, monitors his performance, effectively performs information sharing and makes better decisions [20].

Hence, socially shared information among humans provides them better understanding of the environment and guides them towards the desired goals. According to [21], human beings transfer and gain knowledge from each other through the process of teaching and learning. There are different learning mechanisms in socially shared interactions adopted by humans to gain the maximum benefit such as exemplar based learning, peer learning etc. In exemplar based learning, humans learn by following a role model and often the exemplar is not an individual but a group of individuals. The learning environment here is one way where the exemplars are providing the information and learners are following them. Peer learning is a successful learning technique adopted in human beings especially by the students where learners learn from each other. A combination of several socially shared information processing principles is provided by the mentoring based learning. Here, humans do not completely imitate each and every social influence instead they perform self-reflection to identify the proper information for adopting in their learning process. This method allows an individual to perform self-reflection of his learning and identify the proper information for adopting a particular strategy [22]. Guidance on the other hand, is another highly influential learning strategy adopted among elite and weak learners. Here, the elite learners guide the weak individuals and raise them towards a better performance.

Human cognitive psychology can be explored for applying the learning mechanism in an optimization algorithm. Further, there are several avenues, such as individual learning, mentor based learning, group based learning (mentoring and guiding) etc that can be explored. Motivated from this, the study is directed towards development of algorithms that can optimize any problem utilizing the human self and social cognition i.e. possesses a human-like self-regulative mechanism for better decision making. If one closely studies any of the available optimization algorithms, they do not possess self-regulatory characteristics. Applying human learning principles in the algorithm will provide a self-regulative mechanism to the algorithm for effective search.

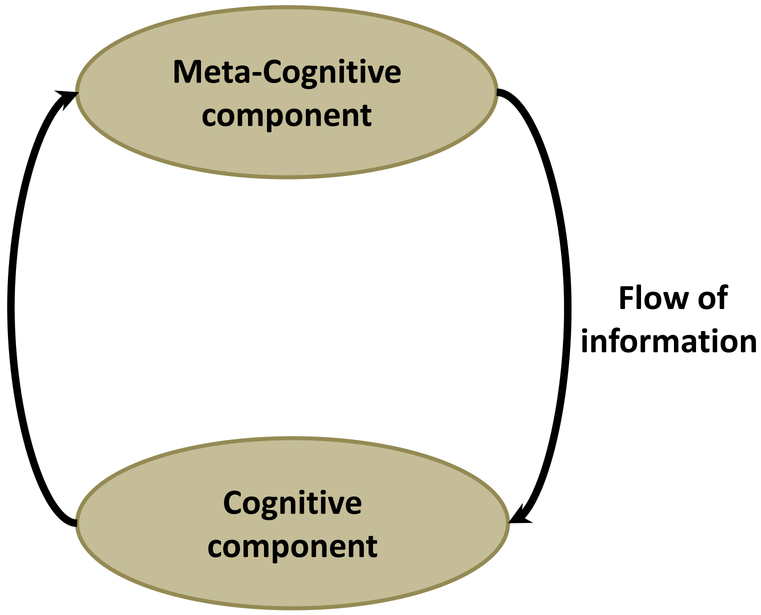

Figure 1.1: Nelson and Naren's Basic Framework of Human Cognitive Processes [1]

Among the nature inspired optimization algorithms, a computationally effective and simple algorithm is the Particle Swarm Optimization (PSO) introduced in 1995 by Kennedy and Ebenhart [13]. PSO is a population based optimization algorithm. PSO has become an immediate choice for optimization technique due to its reduced memory requirements and ease of implementation as well as its extraordinary performance for providing optimum solutions on numerous benchmark problems and real-world applications. It has been successfully implemented on many real-world problems [23, 24, 25, 26]. PSO is derived from the collaborative swarm behavior for search of food by birds flocking and fish schooling [27, 28]. Each member in a swarm updates its search patterns using its own experience i.e. exploration and other members' experiences i.e. exploitation and share information with each other throughout the searching process to ensure that all of them move towards the optimum solution [27]. PSO is an effective algorithm for providing good and competitive performance [29] at a very low computational cost [4]. In the recent past, PSO has been extensively researched whereby researchers focused on development of new strategies in the PSO algorithm [30, 31, 32, 33, 34, 35] and most recently [36, 37, 38, 39, 40, 41, 42, 43, 44, 45]. It has been seen that different variants of PSO have provided much promising convergence characteristics.

Further, researchers have trusted the PSO algorithm for solving different real-world applications. In the thesis, the staging problem of multi-stage launch vehicle optimization has been addressed. Launch vehicle staging design is a complex and computationally intensive optimization application [46]. The placement of satellite at appropriate orbits is highly dependent on vehicle configurations such that the total vehicle weight is minimized and the ratio of the payload to lift off weight is maximized. The launch configuration optimization has been carried out in [47]. In [48], an effective multi-objective PSO algorithm has been proposed for successfully performing population classification in fire-evacuation operations. The algorithm has been reportedly applied to real-world fire evacuation operation at China and has provided successful fire evacuation. In [49], a hybrid Adaptive Radial Basis Function-PSO (ARBF-PSO) has been proposed for financial forecasting purposes. The ARBF-PSO algorithm has been applied to forecast foreign exchange rates whereby it has provided better results on tested currencies with statistical accuracy and trading efficiency. Recently, a Self-Learning PSO (SLPSO) algorithm has been applied to vehicle routing problem [50] whereby the algorithm has provided the best routes by successfully handling the constraints. Similarly, multiple non-linear objects tracking for real-time applications have been successfully performed by Weight Adjusted PSO (WAPSO) algorithm with robustness [51]. In medical field, PSO technique together with Fuzzy C-Means clustering referred to as PSO-FCM algorithm has been utilized to perform image segmentation for MRI images [52]. Recently, several real-world applications have been successfully solved by PSO [53, 54, 55, 56, 57, 58].

It is evident that the performance of PSO can be further improved by incorporating human intelligence, creativity, productivity, decision making, planning and regulation. Therefore, PSO has been adopted in this thesis as a candidate for integrating human learning principles inspired learning mechanism in the algorithm.

1.2 Objectives

The primary objectives of this thesis is to incorporate human learning principles described in the human learning psychology into a nature inspired population based optimization algorithm and assess the impact on the algorithm's performance. Among nature inspired algorithms, PSO is simpler and efficient therefore, we have taken it as a candidate to integrate the concept. The main objectives are discussed below:

- To integrate human self learning principles described in the human learning psychology in PSO that will introduce a self-regulatory mechanism to the basic PSO framework. As discussed earlier in the previous section, the best planners employ self-regulation and use self-perception of their past experiences to adjust their future directions of search. This results in effective search and provide better outcomes. Hence, the main objective of the thesis is to implement human self-regulation in the PSO algorithm.

- To explore more human social learning principles and integrate the concepts in the PSO algorithm. It is well-known that human beings possess intelligence and have good social cognizance and they perform learning using multiple information processing strategies. Taking an inspiration from a classroom environment and exploring different learning techniques adopted by the learners i.e. How humans interact with each other? How they mentor others? How they guide others? etc. gives a clear picture of human social interactions. Human self-learning principles incorporated in PSO might not work better on complex problems due to the lack of social interactions among particles. Socially shared information will help the particles to acquire a better learning scheme that can lead them towards potentially better solutions. Hence, another main contribution of the thesis is to develop a human social learning principles inspired PSO algorithm.

- To make the algorithm rotationally invariant for tackling the rotated characteristics of the complex problems. PSO is a rotationally variant algorithm that performs search separately along each dimension. On the other hand, the rotated characteristics of a problem demand such an algorithm that can perform effective search on non-separable dimensions. To address this issue it is necessary to incorporate rotationally invariant characteristics in the PSO algorithm.

- To integrate a constraint handling mechanism in the structure of the PSO algorithm. Most of the real world problems are governed by constraints and the PSO algorithm is designed to solve unconstraint optimization problems. Therefore, it is necessary to design an algorithm that can efficiently address the constraints of any problem.

1.3 Major Contributions of this Thesis

In this thesis, motivated from human cognition a novel scheme of simultaneously applying multiple strategies for updating the particles' velocity in the PSO algorithm have been proposed. The major contributions of this thesis are:

- **Development of a Self-Regulating Particle Swarm Optimization (SRPSO) algorithm:** A self-regulated learning mechanism in the form of self-regulation and self-perception has been incorporated in the PSO algorithm. Since, the particles employ self-regulated search and perception, the algorithm is referred to as Self Regulating Particle Swarm Optimization (SRPSO) algorithm. The main components of the SRPSO algorithm are:

  - *Self-regulated inertia weight* strategy for the best particle. Here, the particle which achieves the global best at any given time, remains in the exploration mode without interacting with other particles and it also adaptively adjusts its learning strategies by self-regulating its inertia weight. Hence, the best particle explores effectively and provides better convergence towards the global optimum.

- *Self-perception for the selection of direction from global best* strategy for the other particles. Here, the particles make self-perception for the selection of directions from the global best value to update their velocities. This helps in better convergence as updating is not dependent on all the global best directions where some of the directions might take the particle away from the global optimum.

The effect of these strategies on convergence is studied using the CEC2005 benchmark problems [59]. The results obtained from SRPSO and a statistical analysis for the same clearly proves that SRPSO is significantly better in providing accurate solutions on the benchmark problems as compared to other state-of-the-art PSO algorithms.

- **Development of a Dynamic Mentoring and Self-Regulation based Particle Swarm Optimization (DMeSR-PSO) algorithm:** Human socially shared information processing scheme has been implemented in the PSO algorithm. The proposed algorithm employs the process of dynamic mentoring and self-regulation that allows the particles to improve their search capabilities by intelligently utilizing both the self and social cognizance. Here, the particles are divided into three groups consisting of mentors, mentees and independent learners. The main components of the DMeSR-PSO algorithm are:

  - *The mentors group:* The particles in this group consists of the best particle and the particles closer to it in terms of fitness difference and Euclidian distance. The best particle performs search using the same self-regulated inertia weight strategy of SRPSO. The other particles in this group perform belief based search where the particles will have a full self-belief and partial social-belief. Therefore, the particles will perform search with a strong influence of their own experiences.

- *The mentees group:* The poorly performing particles are grouped as mentees and are mentored for performance improvement. These particles utilize either the self-belief search strategy or the social-cognition based search strategy.
- *The independent learners group:* The remaining particles are grouped as independent learners. The self-perception for the selection of direction from the global best strategy of SRPSO has been adopted by the particles in this group.

The effect of these strategies on convergence is studied using benchmark problems. The results indicate that DMeSR-PSO is significantly better than other PSO variants and other population based optimization algorithms with a 95% confidence level, yielding an effective optimization algorithm for real-world applications.

- **Development of a Directionally Driven Self-Regulating Particle Swarm Optimization (DD-SRPSO) algorithm:** SRPSO and DMeSR-PSO algorithms are both rotationally variant algorithms. The complexities associated with real world problems also include rotated characteristics whereby the objective functions cannot be optimized separately along each dimension. Therefore, a directionally updated and rotationally invariant SRPSO algorithm has been developed named as Directionally Driven SRPSO (DD-SRPSO) algorithm. The main components of the DD-SRPSO algorithm are:

  - *Directional update strategy:* In this strategy, a group of elite particles from the top performing particles are selected to update the search directions of the group of poorly performing particles. The velocity update for poorly performing particles is derived from the group of elite particles. The centroid of personal best performance of three elite particles has been selected as the personal best of poorly performing particles. Also, the group of poorly performing particles will always follow the global best direction.

  - *Rotationally invariant strategy:* The rotationally invariant strategy makes the particles capable of tackling problems with rotational characteristics. In this strategy, a center of gravity $\overrightarrow{G_i}$ is calculated around three points; a) particle's current position, b) particle's personal best position and c) local best position of the swarm. Then, a hyper-sphere is generated centered at $\overrightarrow{G_i}$. Finally, a random point is selected in the hyper-sphere as the new current position of the particle.

The effect of these strategies on convergence is studied using benchmark problems. The experimental studies clearly indicate that DD-SRPSO is significantly better than other PSO variants and evolutionary algorithms on a wide range of problems.

- **Constraint handling technique:** Most of the real world applications have constraints associated with them. For example, the economic decision applications are subjected one or series of constraints. Any manufacturing firm will take decisions of maximizing the profit subjected to the constraints of production capabilities of the firm. Similarly, another application area is the ground traffic management at an airport. The airport traffic management will take decisions of minimizing the waiting time of aircrafts subjected to the constraints of number of taxiways, runways and holding areas. Therefore, a new constraint handling technique has been implemented in DD-SRPSO and the algorithm is referred to as DD-SRPSO with constraint handing mechanism (DD-SRPSO-CHM). Here, equality constraint handling mechanism has been incorporated within the structure of the DD-SRPSO algorithm in such a way that during the updation process the particles will never violate the constraints. The proposed constraint handling mechanism has been studied in the optimization of multi-stage launch vehicle configuration. In the staging problem, there are certain equality constraints that are required to be met for maximum outcome. The DD-SRPSO-CHM algorithm achieves the maximum payload whereby the DD-SRPSO algorithm fails to achieve the same. Further, the comparative analysis suggests that DD-SRPSO-CHM is faster in solving the staging problem.

1.4 Organization of this Thesis

The rest of the thesis is organized as follows:

Chapter 2 presents the literature review on particle swarm optimization algorithm, consisting of the basic PSO and also the state-of-the-art PSO variants. The state-of-the-art PSO variants are classified into four categories, namely; parameter selection, neighbourhood topology, learning strategy update and hybrid version and the different variants of the algorithm available in each category are discussed in detail.

Chapter 3 presents the SRPSO algorithm. The chapter starts with a brief overview of the human learning principles followed by its incorporation in the PSO framework. All the incorporated strategies are discussed in detail and a new self regulating PSO algorithm is presented with an analysis of each strategy. The proposed algorithm is compared with six state-of-the-art PSO variants on twenty five CEC2005 benchmark functions and the experimental setup and results are provided. This chapter also presents a detailed analysis of the performance comparison and statistical significance of SRPSO. Finally, the computational complexity analysis of the selected PSO variants compared to the SRPSO algorithm is also provided in this chapter.

In **chapter 4** human socially shared information inspired DMeSR-PSO algorithm is presented. First, an overview of some human social learning principles is provided highlighting the importance of mentoring based learning followed by the process of mentoring based learning. Next, the incorporation of mentoring process in PSO is discussed together with the learning strategies. Then, an analysis of the impact of learning strategies on the performance is presented. Finally, a comparative analysis of the performance of DMeSR-PSO algorithm compared to SRPSO and eight other evolutionary algorithms is provided.

Chapter 5 introduces the Directionally Driven SRPSO algorithm. First, a brief overview is provided followed by the detailed description of the DD-SRPSO algorithm. The chapter includes details about learning strategies, selection of particles for the strategies and an analysis of the impact on convergence. Performance of DD-SRPSO has been compared with well-known PSO variants and other evolutionary algorithms on more complex CEC2013 benchmark functions and the detailed comparison together with a rank based and statistical analysis is provided. Finally, a comparative analysis of SRPSO, DMeSR-PSO and DD-SRPSO is also included in this chapter.

Chapter 6 provides the performance evaluation of SRPSO, DMeSR-PSO and DD-SRPSO on practical optimization problems. This chapter contains brief description of the selected practical problems and provides the comparative performance analysis. Next, the DD-SRPSO-CHM and its application in optimizing multi-stage launch vehicle configuration have been discussed. The chapter also includes the details about multi-stage launch vehicle and a comparative performance analysis on the problem.

Chapter 7 summarizes the conclusions of the research and provides some recommendations for the future works.

Finally, the list of publications and Appendices are included at the end of the thesis.

Chapter 2

Comprehensive Review of Existing PSO Algorithms

Particle swarm optimization (PSO) is derived from the collaborative swarm behavior for search of food by insects, birds flocking, and fish schooling. The model was first introduced by James Kennedy and Russell Eberhart in 1995 [13] and has attracted the interest of researchers due to its simplicity, high-performance, flexibility and much lower computation cost. Each member in a swarm learns from its own experience and other members' experience and updates its search patterns. This behavior is represented using mathematical model where the members are referred as particles. These particles provide a potential solution in the search space. The particles search along the entire space by moving with a certain velocity to find the global best position. Each particle collaborates with the other particles and share its experiences with all of them. The particles adjust their movement and position by updating their fitness values and velocities using their own experience and other particles' experiences. The location of the food is used as the global optimum in the PSO algorithm. In this chapter, the standard PSO algorithm and some state-of-the-art variants of the algorithm are discussed.

2.1 The Standard Particle Swarm Optimization Algorithm

Particle swarm optimization [13] is a search optimization technique used to find the optimal solution motivated from the behaviour of bird flocking. Each swarm has a population of particles as its members, which are initialized along the D-dimensional search space, having random positions.

Each particle i consists of a D-dimensional position vector $\overrightarrow{X_i} = [X_{i1}, X_{i2}, X_{i3}, \cdots, X_{iD}]$ and velocity vector $\overrightarrow{V_i} = [V_{i1}, V_{i2}, V_{i3}, \cdots, V_{iD}]$. The two equations used in the algorithm for updating the velocity and position of the particles are:

$$V_{id}^{t+1} = V_{id}^{t} + c_1 r_1 (P_{id}^{t} - X_{id}^{t}) + c_2 r_2 (P_{gd}^{t} - X_{id}^{t}) \tag{2.1}$$

$$X_{id}^{t+1} = X_{id}^{t} + V_{id}^{t+1} \tag{2.2}$$

where id represents the d^{th} dimension of i^{th} particle and V_{id} and X_{id} are the velocity and position respectively of the corresponding particle. t and $t+1$ represents the current and next iteration respectively. P_{id} represents the personal best position of particle i and P_{gd} represents the global best position i.e. the best position among all the particles. c_1 and c_2 represents the acceleration constants and r_1 and r_2 are the random numbers distributed uniformly within the range $[0, 1]$. The three terms in the velocity update equation (1) defined above describe the local behaviors followed by the particles and these terms are momentum, cognitive component and the social component respectively. The first term serves as the memory of the previous flight directions, the second term is used for keeping a track of the previous best position and the third term is used for interaction with the neighbours.

However, the primary PSO algorithm represented by Equations (1) and (2) does not work desirably, due to the lack of adjusting strategy for the trade-off between exploration and exploitation capabilities of PSO. Therefore, to achieve the desirable output an inertia weight was introduced in PSO's velocity update equation [30]. For updating the velocity of particles, the previous velocities are multiplied by a parameter called inertia weight to balance the exploration and exploitation capabilities. During the entire run of the search process, the inertia weight is commonly decreased linearly in order to ensure that the particles are focused towards exploration at initial stages and later focus on exploitation. The corresponding velocity update equation is:

$$V_{id}^{t+1} = \omega V_{id}^{t} + c_1 r_1 (P_{id}^{t} - X_{id}^{t}) + c_2 r_2 (P_{gd}^{t} - X_{id}^{t}) \tag{2.3}$$

It has been proved that the inclusion of ω in the velocity update equation provides much better solutions [30, 60]. Therefore, researchers have always used and considered the update equations represented by (3) and (2) as the standard PSO velocity and position update equations respectively. The velocity update process and the displacement of the particles in a two dimensional search space is shown in Figure 2.1. In the figure, the weighted influence of the three components to find the new velocity and displacement of a particle is clearly demonstrated. Particle's velocity due to the inertia, velocity factor due to the best local (P_i) and velocity factor due to the best global (P_g) are summed up together to determine the final velocity $V(t+1)$ and then the final displacement $X(t+1)$ is evaluated. Next we present the pseudo-code of the basic PSO algorithm as shown in algorithm 1.

The process of information flow between the particles is performed through a defined topology for the neighbourhood. The collaboration among the particles in basic PSO as described in [61] is performed using a global star topology, where all the particles are attracted simultaneously to the best particle as shown in figure 2.2(Reproduced as in [62]). Here, the best particle is considered nearer to the global optima and it is assumed that it will converge fast.

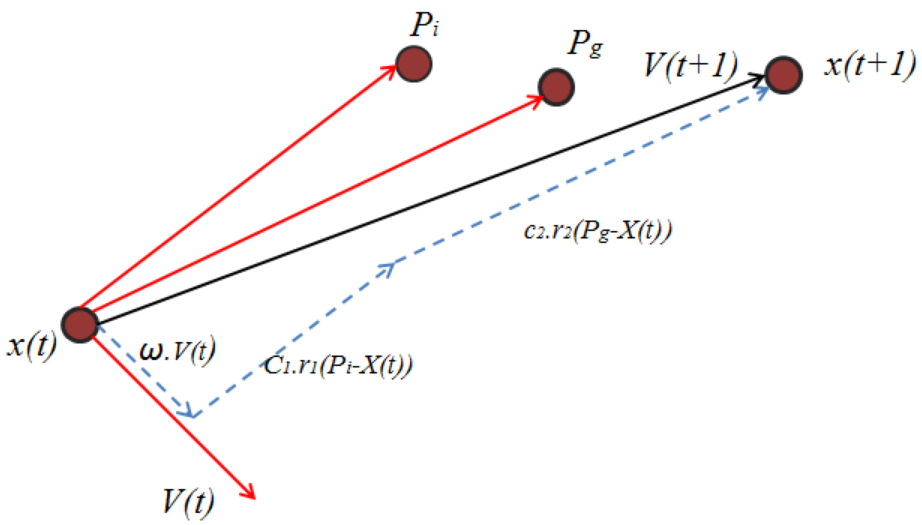

Figure 2.1: The PSO update process

Initialization:
for *each particle i* **do**
  Randomly initialize position of each particle X_i in the search range (X_{min}, X_{max})
  Randomly initialize velocity of each particle V_i
end
The PSO Loop:
while $(success = 0 \text{ and } t \leq max\_iterations)$ **do**
  Calculate the fitness values for each particle;
  Find the personal best position of each particle;
  Find the Particle with the best fitness value;
  Assign this value to global best;
  for $i = 1 : Swarmsize$ **do**
    for $j = 1 : Dimension$ **do**
      Update the velocity using equation (1.3);
      Update the position of each particle using equation (1.2);
    end
  end
end

Algorithm 1: The Basic PSO Algorithm

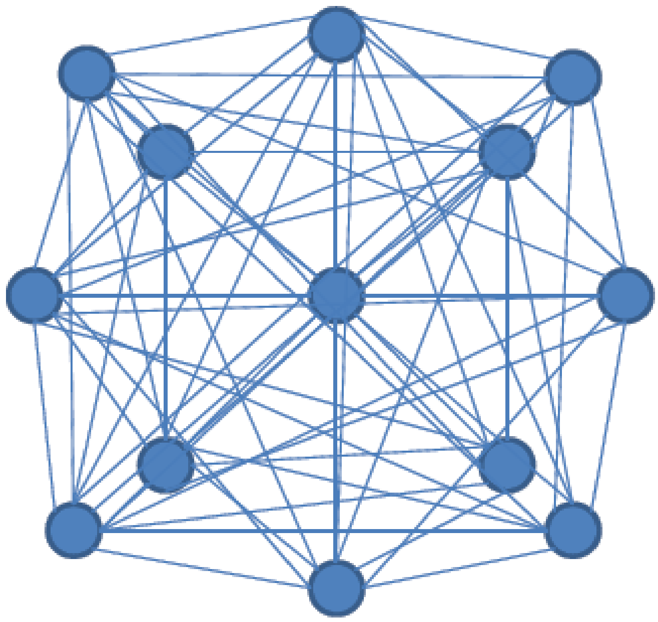

Figure 2.2: PSO Neighbourhood Topology

2.2 A survey on PSO Variants

From the time of initial introduction of PSO, there had been several variants proposed by the researchers and these variants can be broadly categorized into four groups namely;

1. Parameter selection,
2. Neighbourhood topology,
3. Hybrid version and
4. Learning strategy update.

2.2.1 Parameter Selection

Intelligent selection of the parameters is very important in providing the optimal solution. As soon as the inertia weight ω was initially introduced [30], it became an overwhelming research area and has been extensively researched for its setting procedures. The variants of inertia weight setting include fixed [30], linearly decreasing [60], linearly increasing [63], simulated annealing [64], Gaussian [65], random [66], non-linear [67], exponential [68], adaptive [69, 70] and fuzzy adaptive [71, 72]. A recent comparative study [73] has proved that the linearly decreasing inertia weight is the most appropriate setting because it is simplest and most efficient.

Acceleration coefficients had also been extensive researched for providing better performances. The coefficients were originally introduced and have been widely used as constant values [74] but researchers have also investigated the time varying acceleration coefficients [34, 75] and adaptive acceleration coefficients [76, 77] to provide more efficient computational behaviour. Recently, the research is focused towards simultaneously tuning the inertia weight and acceleration coefficients for velocity update. The areas include adaptive parameters [78, 79] and different parameters for all the particles [80, 81]. The simultaneous tuning approach has successfully enhanced the accuracy and reliability for convergence towards the global optimum but it has introduced complexity in the algorithm as more parameters are required to be tuned.

Among many variants, PSO with constriction factor [31] has been cited widely in the literature. The working principle of the same is presented next.

- A constriction factor was also introduced in PSO [31] to control the magnitude of all velocities. The work was investigated on the multidimensional version of dynamic system defined as

$$v(t+1) = v(t) + \Phi y(t) \tag{2.4}$$

$$y(t+1) = -v(t) + (1-\Phi) y(t) \tag{2.5}$$

The above equations were made equivalent to 1-dimensional version of the PSO model using the following assumptions

$$y(t) = \frac{\Phi_1 p_{id} + \Phi_2 p_{gd}}{\Phi_1 + \Phi_2} - x(t) \quad and \quad \Phi = \Phi_1 + \Phi_2 \tag{2.6}$$

Then further modifications were performed to the dynamics of the system and five different coefficients were introduced in equations 2.4 and 2.5 where the PSO system was discussed as a special case of the general system defined by

$$v(t+1) = \alpha v(t) + \beta \Phi y(t) \tag{2.7}$$

$$y(t+1) = -\gamma v(t) + (\delta - \eta \Phi) y(t) \tag{2.8}$$

where $\Phi > 0$ and the coefficients $\{\alpha, \beta, \gamma, \delta, \eta\}$ are used for controlling the velocity explosion and convergence of the system. Further, a generalized PSO model known as a class 1" system was introduced where the coefficients were controlled as $\alpha = \beta = \gamma = \eta$ and $\delta = 1$ to provide the following system

$$v(t+1) = \chi(v(t) + \Phi y(t)) \tag{2.9}$$

$$y(t+1) = -\chi v(t) + (1 - \chi \Phi) y(t) \tag{2.10}$$

Finally, by substituting equation 2.6 in equations 2.9 and 2.10 the PSO variant with a constriction factor was introduced as

$$v(t+1) = \chi(v(t) + \Phi_1 r_1 (p_b - x(t)) + \Phi_2 r_2 (p_g - x(t))) \tag{2.11}$$

$$x(t+1) = x(t) + v(t+1) \tag{2.12}$$

where $\chi = \frac{2k}{2 - \Phi - \sqrt{\Phi(\Phi - 4)}}$ and $\Phi = \Phi_1 + \Phi_2$ subjected to the constraints $\Phi \geq 4$ and $k \in [0, 1]$.

A mathematical proof for the convergence towards the global optima was shown in the work. The PSO variant having a constriction factor is more likely to provide better solutions for selected problems than the basic PSO algorithm. The constriction factor PSO (χ PSO) has been widely used by researchers for comparative analysis of their algorithm.

2.2.2 Neighbourhood Topology

Neighbourhood topology is another crucial factor for the performance of PSO because the flow-rate of the best solution among the particles is decided through their connectivity. The neighbourhood topology used in the initial PSO as described in [61] is the ring topology where each particle has two neighbours. Initially, different topologies like fully connected, wheel and Von Neumann [62] were studied for information flow and it was recommended that the Von-Neumann topology where each particle is connected to four neighbours provides better solutions. Researchers investigated several other topologies for selection of neighbourhood among the particles [82, 83, 84, 85] aiming towards enhancing exploration capabilities of the algorithm. Some of these algorithms have recently developed with enhanced convergence characteristics [86, 87, 88]. Among different variants, the fully informed particle swarm [32] and dynamic multi-swarm particle swarm optimization [89] have been widely cited by researchers for comparative analysis of their work. Next, the working principles of both the variants are discussed.

- In [32] a novel information flow process was introduced for updating the position of each particle. The strategy named as Fully informed particle swarm (FIPS) used a weighted sum of all the neighbouring particles for updating the position of a given particle. The main advantage of the strategy is that a particle's new position is evaluated using the information of all the topological neighbours unlike the traditional PSO where only the best neighbouring particle is used for update process and the information from the remaining neighbours is unused. The condensed form of the formula proposed in χPSO for velocity update equation is given by

$$v_i(t+1) = \chi(v_i(t) + \overrightarrow{\Phi}_k \otimes (\overrightarrow{p}_m - x_i(t))) \tag{2.13}$$

where $\otimes$ performs point-wise vector multiplication, $\overrightarrow{\Phi}_k$ is the vector of Φ values of all neighbours and $\overrightarrow{p}_m$ is the convergence point in the search space.

A novel form of calculation of $\overrightarrow{\Phi}_k$ and $\overrightarrow{p}_m$ is introduced in fully informed PSO and the strategy is given by

$$\overrightarrow{\Phi}_k = \overrightarrow{U}[0, \frac{\Phi_{max}}{N}] \ \ \forall k \in N \tag{2.14}$$

$$\overrightarrow{p}_m = \frac{\sum_{k \in N} W(k) \overrightarrow{\Phi}_k \otimes \overrightarrow{p}_k}{\sum_{k \in N} W(k) \overrightarrow{\Phi}_k} \tag{2.15}$$

where $\overrightarrow{U}[0, \frac{\Phi_{max}}{N}]$ is a function that returns a vector whose position are randomly generated within the range $[0, \frac{\Phi_{max}}{N}]$ using the uniform distribution, N is the set of neighbours of a given particle, $\overrightarrow{\Phi}_k$ is the best position found by particle k and W is the weight of the neighbour. This novel strategy where all the neighbours contribute to the velocity update of a given particle, has given better results and converges faster to the global optima for the selected problems.

- Another novel approach of dynamically changing neighbourhood topology for each particle was introduced in [89] named as Dynamic Multi-Swarm Particle Swarm Optimizer (DMSPSO). The strategy involves random selection of small swarms with small neighbourhood in the early stages to provide better exploration and then dynamically increasing the neighbourhood by regrouping the swarms to incorporate social interaction and perform better exploitation in the later stages of the search process. The velocity update strategy used in the algorithm consists of two stages where 1^{st} stage is from 0 to 90% of maximum iteration and 2^{nd} stage is 90%+ of maximum iteration. In the first stage, the following velocity update strategy is applied

$$V_{id}^{t+1} = \omega V_{id}^t + c_1 r_1 (pbest_{id}^t - X_{id}^t) + c_2 r_2 (lbest_{kd}^t - X_{id}^t) \tag{2.16}$$

where $pbest_{id}$ is the personal best position of particle i along d dimension and $lbest_{kd}$ is the local optima found by neighbour k along d dimension. During the search process, a randomized regrouping schedule 'R' is implemented to enhance the diversity of the particles by dynamically changing the neighbourhood structures. Therefore, in every R generations, the population is regrouped randomly and starts searching using a new configuration of small sub-swarms. In this way, the information obtained by each sub-swarm is exchanged among the whole swarm. In the second stage, the velocity update equation of the standard PSO algorithm is utilized. The dynamic neighbourhood structure has provided much better performances on complex multimodal problems.

2.2.3 Hybrid Version

Hybridization for improving performance of PSO has been actively researched in which different techniques are combined with PSO. The main focus has been directed towards mitigating the weaknesses of the PSO algorithm by combining it with useful properties of different strategies. The two well known methods for integrating different techniques in PSO are the transition technique and the parallel technique. The former is combination of a technique in PSO and the later is simultaneous running PSO with another technique. The transition technique has proved much better and hence it has been widely used. The techniques include embedding the unique evolutionary operators: selection, crossover and mutation inside PSO [90, 91, 92], differential evolution [93, 94, 95, 96], diversity enhancing mechanism [97, 36], avoidance mechanism [98, 99, 100], combination with krill herd algorithm [101] and combination of multi-crossover with bee colony mechanism [102]. One can have a look into [103, 104] for detailed survey on hybridized PSO variants. Combining PSO with other optimization techniques has provided better performances and insight knowledge about the convergence behaviour but on the expense of complexity.

2.2.4 Learning Strategy Update

The parameter tuning and neighbourhood topology selection have provided much better solutions than the basic PSO but there exist some issues that does not guarantee convergence towards global optima in complex multi-modal search space. Hence, the demand of complex real-world problems encouraged the researchers to explore the learning strategies of standard PSO and provide some new efficient learning strategies. A novel algorithm using personal best, global best and another particle's personal best with a maximum fitness-to-distance ratio for updating a particle's velocity is introduced in [105]. Another variant using historical best position together with global best and nearest neighbour is introduced in [106] as an adaptive learning PSO algorithm. Further, many unique adaptive/ evolving strategies have been incorporated in PSO [107, 108, 109]. Human cognition inspired PSO learning strategy is implemented in [110] where velocity update is performed using social component only. Recently, many variants with different learning strategies are introduced for providing better convergence towards global optimum like teaching and peer learning PSO [37], orthogonal learning strategy for PSO [111, 112], ageing mechanism transformed PSO [113], distance based locally informed PSO [114], cellular PSO [115], quantum PSO [116] and a new fitness estimation strategy for PSO [117]. Among different variants, few of the PSO variants have gained noticeable attention due to their novel strategies and promising convergence characteristics like bare bones PSO [118], unified PSO [119] and comprehensive learning PSO [35]. Next, the working principles of the mentioned variants are discussed in detail.

- In the earlier variants of learning strategy update, the PSO algorithm was investigated by eliminating the velocity formula [118] referred to as bare bones particle swarms. The work focussed towards searching for some strategies that can allow PSO to work when some of its traditional features are eliminated. A gaussian distribution based search space is utilized by bare bones PSO where swarm's structure and search history are used for setting the mean and standard deviation. Each particle finds its mean value by investigating in a region where its best value is encountered and fly in the search space by adjusting its variance through neighbourhood distance. Here, a particle's position update rule in the j^{th} component is given by

$$x_{ij}^{t+1} = N(\mu_{ij}^t, \sigma_{ij}^t) \tag{2.17}$$

where N is the normal distribution with

$$\mu_{ij}^t = \frac{p_{ij} - p_{gj}}{2},$$

$$\sigma_{ij}^t = |p_{ij} - p_{gj}|$$

Hence, the strategy of exploration is implemented in the early stages of the algorithm since each particle's best position are far from each other. The exploitation process is allowed in the later stages because of the neighbourhood interaction.

- The individual influence of global and local components in PSO were investigated in [119] and a unification factor was introduced to combine the influences together in an efficient manner. This strategy provides a unified composite version of PSO called as Unified Particle Swarm Optimization (UPSO) where the exploration and exploitation capabilities of global and local variants are combined. The velocity update equations in global and local PSO are defined as

$$G_i(t+1) = \chi[v_i(t) + \Phi_1 r_1(p_i - x_i(t)) + \Phi_2 r_2(p_g - x_i(t))] \tag{2.18}$$

$$L_i(t+1) = \chi[v_i(t) + \Phi_1 r_1(p_i - x_i(t)) + \Phi_2 r_2(p_l - x_i(t))] \tag{2.19}$$

where $G_i(t+1)$ is the velocity update equation for global PSO variant and $L_i(t+1)$ is the corresponding velocity update for the local variant. p_g represents the best position among entire swarm and p_l is the local best position in the neighbourhood. UPSO scheme is the combination of these two search directions in a single equation as

$$U_i(t+1) = uG_i(t+1) + (1-u)L_i(t+1) \tag{2.20}$$

and finally the position is updated using

$$x_i(t+1) = x_i(t) + U_i(t+1) \tag{2.21}$$

where $u \in [0, 1]$ is a parameter called the unification factor, used to determine the influence of global and local components. If $u = 1$, UPSO becomes global PSO variant and if $u = 0$, it becomes local PSO variant. The trade-off between exploration and exploitation of particles has been successfully handled by UPSO and much better convergence is provided on selected problems.

- A novel strategy for providing comprehensive learning capabilities to the particles for solving complex multi-modal functions was proposed in [35] as Comprehensive Learning PSO (CLPSO). This novel strategy updates a given particle's velocity using all other particles' personal best information. Here, each dimension of a given particle can learn from different particles' best dimensions. This helps in retaining a particle's diversity and hence avoiding premature convergence as any of the particle stuck in a local optima can learn from other particles and eventually escape the local optima. The strategy is mathematically represented as

$$V_{id}^{t+1} = \omega V_{id}^{t} + cr(P_{fi(d)}^{t} - X_{id}^{t}) \tag{2.22}$$

where $fi(d)$ defines which particles' personal best is used by particle i which can either be its own dimension or any other particle's dimension. A learning probability P_c is used to decide $fi(d)$ such that

$$P_c = 0.05 + 0.45 * \frac{(exp(\frac{10(i-1)}{p_s-1}) - 1)}{exp(10) - 1} \tag{2.23}$$

where p_s is the swarm size. Random number r is generated in each dimension and if $P_c > r$ the particle learns from another particle's dimension otherwise it learns from its own dimension. CLPSO has been widely accepted because of its simplicity and ease of implementation [120, 121].

2.3 Limitations in Existing PSO Variants

Some PSO variants have gained a lot of attention and acceptance by the research community, due to their successful novel PSO schemes which have exhibited great performance improvements. Comprehensive review of PSO variants is available in [122]. A summary of seven such algorithms used for comparative analysis of the proposed algorithm is provided in Table 2.1. Most of the PSO variants discussed in the literature have provided much better convergence characteristics on selected problems. All the existing variants proposed in the literature have tried to address the limitations of an old variant and successfully overcame them. Almost all of these variants have followed the collective behaviour of a swarm of creatures which do not possess any self-regulatory mechanism. This has always limited the performance of the particles. These limitations can be addressed by exploring the human learning principles for effective search.

In this chapter, a comprehensive survey of the standard PSO algorithm and some state-of-the-art PSO variants available in the literature have been discussed. Further, limitations in the existing PSO variants have also been discussed to highlight the scope of future research in the algorithm. Hence, a new research direction of incorporating human learning principles inspired learning strategies in PSO for achieving better convergence towards the global optimum in multitude different problems is provided in the next chapter.

Table 2.1: Summary of selected State-of-the-art PSO variants Discussed in Literature Review

Publication	Group	Novelty	Performance
Clerc and Kennedy [31]	Parameter Selection	Introduced a new parameter (χ)	Better convergence on selected problems (unimodal and basic multimodal)
Mendes et al. [32]	Neighbourhood Topology	A new method of information flow (All neighbours contribute in velocity update process)	Faster convergence on selected problems (unimodal and basic multimodal)
Liang and Suganthan [89]	Neighbourhood Topology	Dynamically changing neighbourhood	Better convergence on complex multimodal problems
Kennedy [118]	Learning Strategy Update	Gaussian Distribution based search space, Eliminated Velocity update equation, Position update using mean and standard deviation	Better convergence on selected problems (Most notably on Griewank function)
Parsopoulos and Vrahatis [119]	Learning Strategy Update	Combination of Global and Local search direction using unification factor	Better convergence on selected problems (unimodal and basic multimodal)
Liang et al. [35]	Learning Strategy Update	Only the Personal best information is used for velocity update	Better convergence on complex multimodal problems
Cheng and Jin [40]	Learning Strategy Update	A Particle can learn from any other having better fitness value	Better convergence on complex multimodal problems

Chapter 3

Self-Regulating Particle Swarm Optimization Algorithm

In the previous chapter, a complete survey on particle swarm optimization algorithm has been presented. In the past two decades, PSO has been extensive researched for better convergence towards the optimum solution. All the research areas discussed in the previous chapter have focused towards improving the convergence characteristics. As discussed earlier, human beings are known to be intelligent, possess better decision making abilities and efficient problem solving skills. In human cognitive psychology, better planners are regarded as those who continuously regulate their strategies with respect to their current status and perception on their past experiences. This help them to properly adjust their future directions. Inspired from these human self-learning capabilities, this chapter provides a new variant of PSO algorithm incorporating human self-regulation and self-perception strategies. The algorithm is names as the Self-Regulating Particle Swarm Optimization (SRPSO) algorithm.

3.1 Basic Concepts of Human Learning Principles

The learning process adopted by human beings consists of several learning principles. According to the human learning psychology [1], it has been shown that human beings possess multiple hierarchical inter-related layers of information processing. This helps one to perform better self-regulation of the cognitive strategies and hence enhance the decision making abilities. A better planner is one who continuously monitors his performance and adopts the best learning strategies. An effective learner adopts strategic learning process through planning and selection of appropriate learning strategies.

Human learning principles are basically constructed on the idea of learning with understanding. All the learning strategies adopted by human beings during the learning process is strongly advocated by the understanding of environment. Studies in human learning suggest that the maximum outcome is achieved through learning with deep conceptual understanding [123]. This facilitates the learning process by providing such strategies that identify, monitor and regulate the entire process. Further, the learning process is enhanced through socially shared information processing. Learning with understanding means that an individual is continuously interacting with the environment and monitoring his performance through proper regulation of his knowledge and skills. This provides the individual with capabilities of taking intelligent decisions for achieving the desired goals.

The Nelson and Naren's model provides a general framework for integrating the human regulatory processes using the interactive process of monitoring and control [1]. This interactive process effectively controls the acquisition of knowledge as well as its retention and retrieval [124]. This framework consists of a meta-cognitive component and a cognitive component interlinked with each other through flow of information. According to this framework, human beings think about their cognitive processes and improve their cognitive skills through development of new strategies and evaluation of acquired information. The pictorial representation of proposed basic framework is shown in figure 1.1 in chapter 01. The upper layer in the model is called the meta-cognitive layer which consists of the dynamic model of the lower layer called the cognitive component. The two layers communicate with each other through flow of information. The direction of flow of signal between the two components describes the relationship between them. The flow of information from the meta-cognitive component to the cognitive component is the 'control' signal which is used to modify the cognitive component. The control signals can change the state of the cognitive component or even the cognitive component itself. The flow of information from the cognitive component to the meta-cognitive component is the 'monitor' signal which is used to inform the meta-cognitive component about the current state of the cognitive component. Based on this information, the meta-cognitive component decides the future states of the cognitive component.

3.2 Self-Regulating Particle Swarm Optimization Algorithm

3.2.1 Incorporating the human learning concepts in PSO

The Nelson and Naren's model for integrating the human learning processes as discussed in the previous section has been adopted in this study. The model is well-suited to develop a human learning based PSO algorithm as the control and monitoring interface between the two layered framework will provide a decision making mechanism. This will result in incorporating intelligent swarms' concept in the PSO algorithm. The human learning principles based PSO framework, analogous to the Nelson and Naren's model is shown in figure 3.1. The intelligent PSO framework consists of two inter-related components: the basic PSO is the cognitive component and the decision maker is the meta-cognitive component. The decision maker has been incorporated to decide the future step for the particles by providing them the appropriate 'learning strategy' based on their performance. The performance will be continuously monitored through 'performance monitoring' signals.

Self-regulative learning strategies adopted by human beings are considered to be highly efficient in controlling the learning process. These learning strategies enable them to properly plan the future steps by selecting the appropriate set of skills. Hence self-regulating learning strategies provides the self-awareness and expert learning capabilities

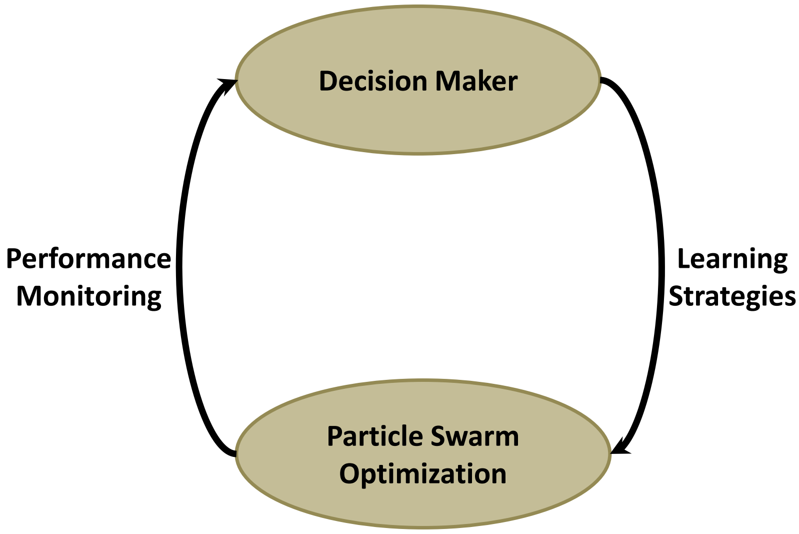

Figure 3.1: Analogous Nelson and Naren's model for Human Learning Principles based PSO

to the learners. This helps the learners to manage their learning throughout the learning process and rectify their deficiencies for achieving the desired goals. In this thesis, inspired from the self-regulating strategies of human learning principles, two main strategies in the basic PSO algorithm for enhancing its search capabilities have been introduced. The two strategies are:

i Self-regulating inertia weight,

ii Self-perception of global search direction.

The first strategy has been introduced for the best particle. During the search process, the best particle will perform the self-regulation of its inertia weight. In this strategy, the particle will be given a higher acceleration of its exploration process through an increased inertia weight. During the accelerated search process, the best particle will only perform exploration of the search space and therefore the search will be performed without any interaction with the other particles. It is to be noted that the strategy is only for the best particle, the particle using this strategy will return to the normal search process (the linearly decreasing inertia weight)as soon as it loses the global best position. The strategy is inspired from the human self-cognition during the learning process. According to self-cognition, if it is confirmed that an individual is the best from everyone in a particular direction guiding him towards the desired goals then he will accelerate quickly following the same direction without much consideration of the social/self-experience. The strategy is purely based on human self-awareness of the surroundings. For example, during a race, if a person at any known position finds himself at the leading position as compared to others, he will start running in the same direction as fast as possible.

The second strategy has been introduced for all the other particles except the best. In a given iteration, the best particle uses his direction as the best direction for the search process. Hence, it is not influenced by any previous self/social experiences. All the other particles, find the appropriate direction in the current iteration using their perception on the global best direction. According to this strategy, particles will follow only those directions from the global best where they have confidence. This strategy is purely based on human perception about himself and his social peers. The best in the group has a very high self-confidence and he is not concerned about his past self-experiences and socially shared information. His perception is always declined towards his current status, neglecting all the previous self/ social experiences. On the other hand, someone who is not the best has a different perception about his self and social experiences. An individual will try to improve his skills by rectifying his deficiencies through monitoring his previous experiences. He will believe in his previous best experience and must have a certain amount of belief on his social peers to learn in a better way.

Incorporating the above-mentioned concepts from the human learning principles, a general equation for the velocity update in the SRPSO algorithm is given by

$$V_{id}^{t+1} = \omega_i V_{id}^t + c_1 r_1 p_{id}^{se}(P_{id}^t - X_{id}^t) + c_2 r_2 p_{id}^{so}(P_{gd}^t - X_{id}^t) \tag{3.1}$$

where ω_i is the inertia weight of i^{th} particle which is self-regulated by the best particle. p_{id}^{se} is the perception for the self-cognition and p_{id}^{so} is the perception for the social cognition and they are defined as

$$p_{id}^{se} = \begin{cases} 0, & \text{for best particle} \\ 1, & \text{otherwise} \end{cases} \tag{3.2}$$

and

$$p_{id}^{so} = \begin{cases} 0, & \text{for best particle} \\ \gamma, & \text{otherwise} \end{cases} \tag{3.3}$$

where γ is either 0 or 1 depending on the threshold value for defining the confidence. In the next section, the proposed two strategies are described in detail.

3.2.2 Self-regulating inertia weight

In the literature of PSO, it has been clearly stated that inertia weight is an essential part of the velocity update equation [73, 125, 33]. For the PSO algorithm, to perform well it is essential that there is a balance between exploration and exploitation processes. Inertia weight plays a vital role in controlling these processes. Previous studies suggest that there is a lot of potential in applying different settings for the inertia weight, thereby yielding favourable results through certain settings for real-world problems [68, 67].

In this thesis, a new strategy for the best particle has been introduced, named as self-regulating inertia weight. According to this strategy, the particle achieving the global best position at any given time during the search process will perform self-regulation of its inertia weight i.e. it will start linearly increasing the inertia weight. This will help the particle to deeply explore the solution it has found by accelerating its search towards the global optimum. As stated earlier, the basic idea behind this strategy is taken from the best learner and accordingly, particle achieving the global best position at any instance believes in his direction and accelerates in the same direction in search of global solution. At the same time, all the other particles follow the standard procedure of linearly decreasing the inertia weight to have balance between their exploration and exploitation of the search space. The self-regulating inertia weight strategy is defined as

$$\omega_i = \begin{cases} \omega_i(t) + \eta\Delta\omega, & \text{for best particle} \\ \omega_i(t) - \Delta\omega, & \text{otherwise} \end{cases} \tag{3.4}$$

where $\omega_i(t)$ is the current inertia weight, $\Delta\omega = \frac{\omega_i - \omega_f}{N_{Iter}}$ (N_{Iter} is the number of iterations, ω_i and ω_f are the initial and final values of inertia weight respectively) and η is the constant to control the rate of acceleration. In our simulation, after testing for different values, the rate is set as 1 which is providing better solutions in most of the test problems.

The pictorial view of self-regulating inertia weights experienced by particles during the search process is presented in figure 3.2. In the figure, three particles 'a', 'b' and 'c' are selected for a minimization problem for illustrating the strategy clearly. For these three particles, the fitness curves and inertia weights are plotted for 1000 iterations.

Particle a: The particle is represented by dashed red lines in the figure. At the start of the search process, this particle has the lowest fitness value and hence it is the best particle. According to the self-regulating inertia weight strategy, if a particle possess a best fitness value at any point during the search then it will linearly increase its inertia weight. From iteration 0 to 200, this particle has the best fitness value. Therefore, as clearly indicated in the figure, the inertia weight for the particle is linearly increasing. Afterwards, from 200 to 400 iterations, the particle is not best and therefore it follows the standard procedure of linearly decreasing inertia weight. After 400th iteration, the particle regains the best position and again it searchs utilizing the self-regulating inertia weight strategy.

Particle b: The particle is represented by black line in the figure. From start till 200th iteration, the particle is not the best and therefore it follows the standard search procedure with linearly decreasing inertia wight. From 200 to 400 iterations, it has taken over the best position from particle 'a' and became the best particle. For this duration, it performs self-regulation of its strategy to search with a linearly increasing inertia weight. Further, when it losses the best position, it again follows the standard procedure of linearly decreasing inertia weight till the time it becomes the best particle.

Particle c: The particle is represented by dashed orange line in the figure. This particle demonstrates the behaviour of such a particle that never achieves the best position

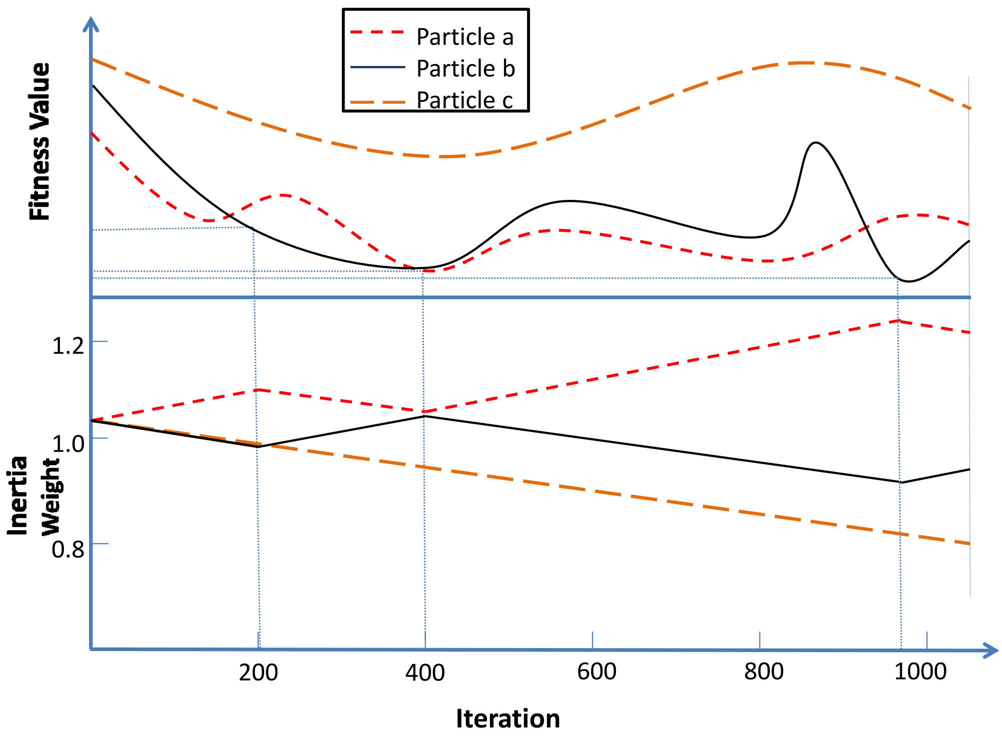

Figure 3.2: The inertia weight update strategy

(best fitness value) in the entire search process. Such a particle will always follow the standard procedure of linearly decreasing inertia weight. In the figure, one can clearly see that the inertia weight of this particle is linearly decreasing throughout the iterations.

3.2.3 Self-perception of global search direction

When humans interact with each other they generate certain perception about others [124]. It is in human cognition that we do not fully trust others and this trust decides the amount of information we want to take and adopt from others. It has been shown in human psychology [126] that the best planners have a better perception over their actions which strongly impact their decisions. They regulate their strategies and plan their actions accordingly. Further, they collaborate with other for information sharing [127]. On the other hand, people interacting with the best planners always have a varying perception, based on which they collaborate. This perception helps them to regulate their action towards achieving the goal [1]. Based on these finding, the velocity update strategy of all the other particles has been kept different from the best particle. Inspired by human socially shared interactive learning strategy, a new learning strategy for all the other particles has been introduced. In this strategy, the best particle will only follow his direction without any self or social exploitation. All the other particles will perform full self-exploitation and partial social exploitation by taking information according to their perception in the global search direction.

In the self-perception strategy, the best particle will have negligible perception on self and social exploitation and his perception will only guide him towards exploration. Therefore, p_{id}^{se} and p_{id}^{so} in equation 3.1 is set to zero. The self-perception strategy for all the other particles is implemented as full self-exploitation and partial social exploitation. In social exploitation, a uniform random number within the range [0, 1], $a \sim U([0,1])$ for all the directions of the global best position is generated and the threshold (λ) value is set at 0.5 for selection of any direction. If a randomly generated number in any direction of the global best position is greater than 0.5, the direction will be selected from the global best position.

It is evident that for achieving better convergence, the particles must have social cognition. The amount of social cognition must be selected intelligently to achieve desired outcome. If one employs full social cognition, all the particles will follow the best particle and the result will be biased towards the global best position reached at a particular iteration. It is possible that the global position at that particular iteration may not be the global optimum solution. While following the best particles, all the particles will converge to that particular point and this may lead to premature convergence. Similarly, if a much higher threshold value is selected, then there will be very low social cognition. All the particles will keep flying in their own directions and the one reaching the best position will not be able to call others to the same position. There will be lack of socially

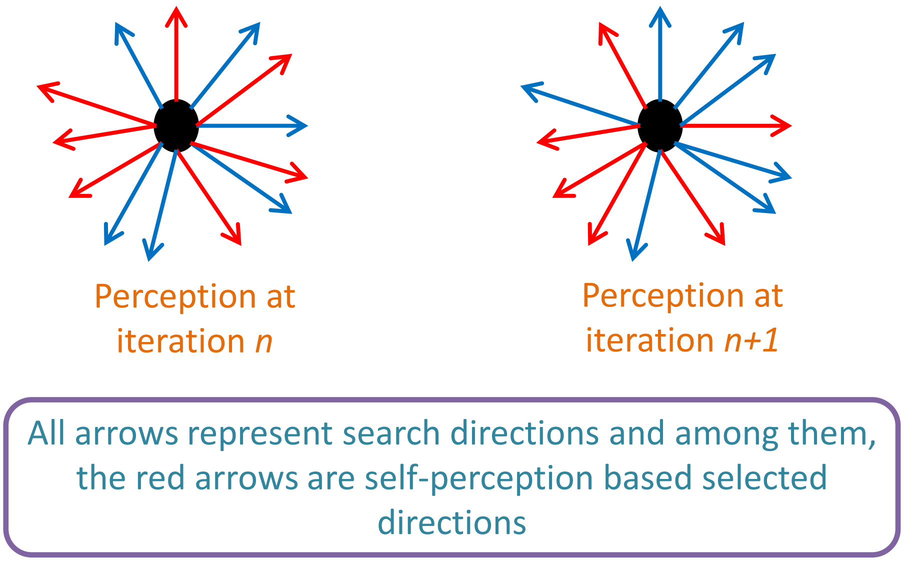

Figure 3.3: Self-Perception of Search Directions strategy

shared information which may leads to diversity loss of the swarm. Hence, in order to perform a balanced social exploitation together with effective exploration 0.5 is selected as the threshold value. The perception of a particle is dynamically changing over time as the selected directions from the global best will keep changing in every iteration.

Figure 3.3 provides a pictorial view of selection of directions by a particle using the self-perception strategy. In the figure, all the arrows are representing the search directions and among them, all the red arrows are those directions which are selected by the particle using the self-perception strategy. It is clearly illustrated in the figure that the selected directions in an iteration is different from those selected in the previous iteration as represented by red arrows for iteration n and iteration n+*1*. The social perception of a particle is given by

$$p_{id}^{so} = \begin{cases} 1, & \text{if } a > \lambda \\ 0, & \text{otherwise where i = 1,2,...,m} \end{cases} \tag{3.5}$$

The self-perception influences the particles' search directions significantly, i.e. the best particle assumes that his direction is the best and discards the social and self-knowledge. Hence, its social (p_{id}^{so}) and self (p_{id}^{se}) perceptions are set to zero. The self-perception for the rest of the particles is set to 1. Hence, other particles use self-cognition and partial social cognition for velocity update. The self-perception in global search direction reduces the risk of particles attracted to local optima.

The velocity update equation for best particle is given by

$$V_{id}^{t+1} = \omega_i V_{id}^{t} \; , \qquad\qquad (i = best) \tag{3.6}$$

and for all the other particles is represented as

$$V_{id}^{t+1} = \omega_i V_{id}^{t} + c_1 r_1 (P_{id}^{t} - X_{id}^{t}) + c_2 r_2 p_{id}^{so} (P_{gd}^{t} - X_{id}^{t}) \tag{3.7}$$

3.2.4 The SRPSO Pseudo-code

Incorporating the two proposed strategies, the pseudo-code for SRPSO is summarized in Algorithm 2.

3.3 Experimental Setup, Results and Discussion

This section provides the empirical assessment of the SRPSO algorithm. Performance has been evaluated on the standard CEC2005 benchmark problems [59] and compared with other well-known PSO variants. The comparison also includes statistical and computational complexity analysis.

This section is organized as follows:

i Brief description of the selected CEC2005 benchmark problems,

ii Impact of self-learning strategies on PSO,

iii Performance evaluation of SRPSO on CEC2005 benchmark problems compared with other PSO variants,

iv Ranks based analysis of the selected algorithms,

v Statistical comparison,

vi Computational complexity analysis.

```
Initialization:
for each particle i do
    Randomly initialize the position X_i in the search range (X_min, X_max)
    And randomly initialize velocity V_i
end
Calculate their fitness values;
Find their personal best position;
The SRPSO Loop:
while (success= 0 and t ≤ max_iterations) do
    Find the Particle with the best fitness value;
    Assign this position to global best;
    for the best particle do
        Calculate the inertia weight ω using equation 3.4;
        Update the velocity using equation 3.6;
    end
    for the rest of the particles do
        for j = 1 : Dimension do
            Generate the uniform random number a;
            if (a > 0.5), then
                Select the directions from global best position
            else
                Neglect the directions
            end
        end
        Update the velocity using equation 3.7;
    end
    Update the position of each particle using equation 2;
end
```

Algorithm 2: The SRPSO Algorithm

3.3.1 Benchmark functions

In May 2005, Suganthan et al. introduced a problem definition and evaluation criteria on real parameter optimization for systematic comparison of optimization algorithms [59]. A set of 25 benchmark functions with common termination criteria, problem dimensions and initialization schemes were introduced in the special session termed as CEC2005. The CEC 2005 benchmark functions [59] broadly fall into four different groups (based on their properties):

i Unimodal (F_1 - F_5),

ii Basic Multimodal (F_6 - F_{12}),

iii Expanded Multimodal (F_{13} and F_{14}) and

iv Hybrid Composition (F_{15} - F_{25}).

To test the robustness and efficacy of any optimization algorithm, the global optimum of all the benchmark functions are shifted as $\boldsymbol{o} = [o_1, o_2, \cdots, o_D]$. The functions are either shifted or shifted and rotated to make the function more complex and hard to solve. The shifted functions are defined as ***z = x-o*** and the shifted rotated functions are defined as ***z = (x-o)*M*** where ***M*** is the transformation matrix for the rotating matrices. The hybrid composition functions are the most complex functions that are designed by selecting different basic unimodal and multimodal functions without shift or rotation properties. Then, all these functions are shifted/ shifted rotated to new optimum positions. A brief overview of the benchmark functions is provided in the Appendix. All the details about the functions can be found in [59].

3.3.2 Impact of self-learning strategies on PSO

This section provides an insight of the impact of the proposed learning strategies, self-regulating inertia weight and self-perception based selection of direction, on the convergence characteristics of the standard PSO algorithm. The impact is illustrated in figure 3.4 using a comparison among performances of PSO, CLPSO and SRPSO. In the figure, the performance on a function from each group is presented where the blue line with triangle marker presents the PSO algorithm, green line with a star marker presents the CLPSO algorithm and the red line with circle marker represents the SRPSO algorithm. All the algorithms has been tested using the same parameter settings as ω = [1.05 - 0.5], $c_1 = c_2 = 1.49445$, swarm size = 40, dimension = 30 and V_{max} = 0.1 * Range. The algorithms have been evaluated 25 times and the plot for median performances are presented in the figure.

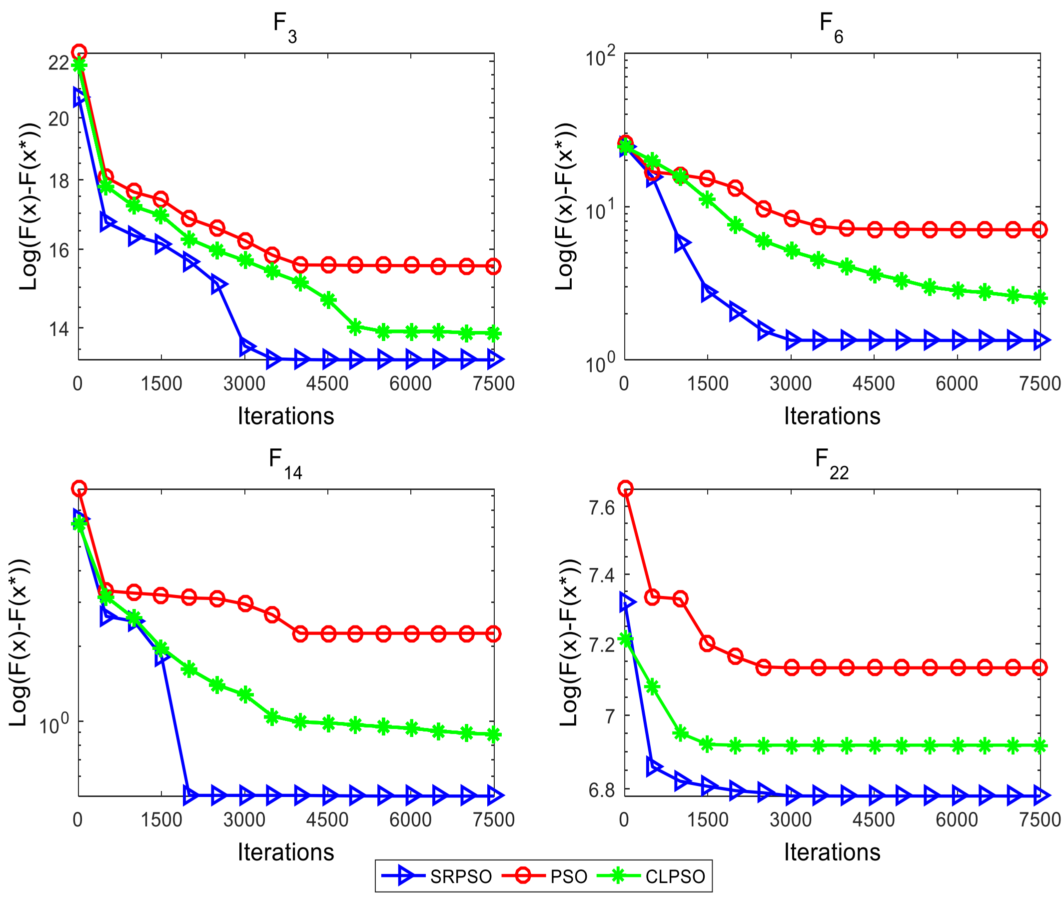

Figure 3.4: Impact of Self-Learning strategies

Figure 3.4 contains four graphs of Log(F(x)-F(x*)) plotted against iterations for each function; one unimodal function F_3, one multimodal function F_6, one expanded multimodal function F_{14} and one hybrid composition function F_{22}. In all the functions, one can easily see that PSO has converged to a point far away from the optimum solution. In comparison to PSO, CLPSO has shown better convergence characteristics towards the optimum solution in all the four functions. On the other hand, with the help of human self-learning principles, the SRPSO algorithm has provided faster convergence much closer to the true optimum solution in all the functions. Similar observations have been made on all the benchmark functions.

3.3.3 Selected PSO algorithms and parameter settings

For a comparative analysis, the following widely accepted PSO variants by the evolutionary computing research community have been selected in this study.

i **The constriction factor PSO (χPSO) algorithm** [31]: The algorithm has provided a balance between the exploration and exploitation capabilities of the particles and has effectively controlled the convergence.

ii **The dynamic multi-swarm particle swarm optimizer (DMSPSO) algorithm** [89]: A novel concept of dynamically changing neighborhood in the search process has been introduced in the algorithm that has provided much better convergence towards the optimum solution compared to the fixed neighborhood search.

iii **The fully informed particle swarm (FIPS)** [32]: The algorithm introduced a novel concept of search strategy for performance update during the search process. Here, the particles perform search using a weighted sum of the neighboring particles that resulted in better solutions on selected problems.

iv **The unified PSO (UPSO) algorithm** [119]: The algorithm combines the local and global search variants of PSO within a single algorithm to enhance the convergence characteristics. Through efficient utilization of both the algorithms, UPSO has proved to be a competitive PSO variant.

v **The comprehensive learning particle swarm optimization (CLPSO) algorithm** [35]: The algorithm is one of the most promising PSO variant widely accepted by the evolutionary community proven to be an efficient algorithm in accelerating the convergence towards the optimum solution. Compared to other variants of PSO, CLPSO searches more potential regions of the search space to find the global optima.

The parameter settings and the experimental setup for the selected PSO variants are:

- Parameters: $\phi = 4.1$, $\chi = 0.72984$, $\omega = [0.9\text{-}0.4]$ and $c_1 = c_2 = 2.05$ [31, 35]
- Ring Topology for neighborhood using a radius of 2
- Function evaluations = $10^4 \times D$ (D is the dimension)

All the experimental guidelines defined in the CEC2005 [59] evaluation criteria has been followed for systematic performance evaluation of SRPSO. All the results for the selected PSO variants are taken from [95]. the median, mean and standard deviation fitness values for both the 30-dimensional (30D) and 50-dimensional (50D) problems are reported. The SRPSO parameters are set as:

- **Inertia weight** ω**:** initial value = 1.05 and final value = 0.5,
- **Constriction factor** c**:** $c_1 = c_2 = 1.49445$
- Swarm size = dimension
- **Maximum velocity:** $V_{max} = 0.1$ * Range.

3.3.4 Experimental results and performance comparison

In this section, a comprehensive performance evaluation of SRPSO compared with the well-known PSO variants on all the CEC2005 benchmark functions is presented. Following the CEC2005 guidelines, experiments were conducted on 30D and 50D benchmark functions.

The results for all the PSO variants including the SRPSO algorithm are provided in the Tables 3.1, 3.2 and 3.3. Table 3.1 presents the median, mean and standard deviation fitness values for all the algorithms on first 9 benchmark functions F_1 to F_9. Similarly, the median, mean and standard deviation fitness values for the next 9 benchmark functions F_{10} to F_{18} are presented in table 3.2. Finally, table 3.3 provides the median, mean and standard deviation fitness values for the remaining 7 benchmark functions F_{19} to F_{25}. As

Table 3.1: Performance on 30D and 50D CEC2005 Benchmark Functions F_1 to F_9

Func.	Algorithm	30-Dimensional			50-Dimensional		
		Median	Mean	StD.	Median	Mean	STD.
F_1	χPSO	5.328E+00	9.657E+00	1.233E+01	5.328E+00	9.657E+00	1.233E+01
	DMSPSO	1.143E+02	3.135E+02	4.149E+02	2.349E+02	3.870E+02	3.855E+02
	FIPS	3.185E+02	5.252E+02	5.571E+02	1.149E+03	1.673E+03	1.524E+03
	UPSO	1.269E+03	1.306E+03	7.328E+02	6.840E+02	7.100E+02	3.290E+02
	CLPSO	**0.000E+00**	**0.000E+00**	0.000E+00	**0.000E+00**	**0.000E+00**	0.000E+00
	SRPSO	**0.000E+00**	**0.000E+00**	0.000E+00	**0.000E+00**	**0.000E+00**	0.000E+00
F_2	χPSO	**0.000E+00**	1.573E+01	8.112E+01	2.334E+02	7.774E+02	1.806E+03
	DMSPSO	1.536E+02	7.801E+02	2.109E+03	3.311E+02	9.666E+02	1.409E+03
	FIPS	1.460E+04	1.470E+04	2.316E+03	2.633E+04	2.574E+04	4.424E+03
	UPSO	6.688E+03	7.602E+03	5.290E+03	3.632E+03	4.220E+03	2.894E+03
	CLPSO	3.828E+02	3.828E+02	1.060E+02	1.013E+04	1.021E+04	1.357E+03
	SRPSO	**0.000E+00**	**0.000E+00**	0.000E+00	**5.870E-03**	**1.931E-02**	1.639E-02
F_3	χPSO	3.491E+06	1.020E+07	1.336E+07	1.852E+07	1.988E+07	1.266E+07
	DMSPSO	3.898E+06	5.623E+06	6.225E+06	8.835E+06	1.317E+07	1.579E+07
	FIPS	1.530E+07	1.945E+07	1.109E+07	5.586E+07	5.867E+07	2.346E+07
	UPSO	4.308E+07	5.303E+07	3.856E+07	4.885E+07	5.340E+07	3.743E+07
	CLPSO	1.204E+07	1.188E+07	3.107E+06	5.084E+07	4.930E+07	1.161E+07
	SRPSO	**1.038E+02**	**1.365E+02**	2.199E+02	**3.638E+02**	**3.927E+02**	2.780E+02
F_4	χPSO	1.555E+03	1.834E+03	1.088E+03	2.813E+04	2.760E+04	1.007E+04
	DMSPSO	2.945E+02	8.561E+02	1.292E+03	1.306E+04	1.343E+04	3.945E+03
	FIPS	2.077E+04	2.068E+04	3.107E+03	3.355E+04	3.424E+04	3.850E+03
	UPSO	1.913E+04	1.876E+04	6.086E+03	1.353E+04	1.449E+04	4.462E+03
	CLPSO	5.481E+03	5.396E+03	1.250E+03	3.477E+04	3.428E+04	5.637E+03
	SRPSO	**2.349E+01**	**3.205E+01**	1.145E+01	**2.373E+03**	**2.559E+03**	8.771E+02
F_5	χPSO	7.886E+03	8.101E+03	1.210E+03	1.127E+04	1.117E+04	1.975E+03
	DMSPSO	4.368E+03	4.261E+03	1.873E+03	5.311E+03	5.533E+03	1.449E+03
	FIPS	1.164E+04	1.174E+04	1.393E+03	1.594E+04	1.589E+04	1.147E+03
	UPSO	1.268E+04	1.282E+04	2.288E+03	1.177E+04	1.207E+04	2.304E+03
	CLPSO	4.011E+03	4.001E+03	4.276E+02	9.753E+03	9.698E+03	7.903E+02
	SRPSO	**1.983E+03**	**2.134E+03**	2.741E+02	**3.646E+03**	**3.725E+03**	5.784E+02
F_6	χPSO	3.602E+02	1.170E+03	1.790E+03	3.979E+01	6.370E+06	2.129E+07
	DMSPSO	2.226E+06	2.721E+07	7.289E+07	2.226E+06	1.768E+07	4.103E+07
	FIPS	9.832E+06	2.457E+07	3.493E+07	6.483E+07	8.021E+07	6.118E+07
	UPSO	6.826E+06	1.187E+07	1.355E+07	1.160E+06	2.731E+06	3.667E+06
	CLPSO	**7.369E+00**	**1.779E+01**	2.285E+01	8.998E+01	8.705E+01	3.757E+01
	SRPSO	2.375E+01	2.446E+01	5.738E+00	**3.547E+01**	**5.754E+01**	3.313E+01
F_7	χPSO	6.788E+03	6.780E+03	1.291E+02	6.158E+03	6.154E+03	7.416E+01
	DMSPSO	4.297E+03	4.335E+03	2.190E+02	6.029E+03	6.050E+03	1.312E+02
	FIPS	7.507E+03	7.477E+03	2.158E+02	1.037E+04	1.036E+04	2.122E+02
	UPSO	7.513E+03	7.524E+03	3.409E+02	7.419E+03	7.420E+03	3.034E+02
	CLPSO	4.696E+03	4.696E+03	1.837E-12	6.195E+03	6.195E+03	4.594E-12
	SRPSO	**4.452E-03**	**6.263E-03**	4.521E-03	**4.396E-01**	**5.197E-01**	1.757E-01
F_8	χPSO	2.090E+01	2.090E+01	5.354E-02	2.114E+01	2.113E+01	4.368E-02
	DMSPSO	2.093E+01	2.093E+01	6.189E-02	2.113E+01	2.113E+01	3.770E-02
	FIPS	2.095E+01	2.094E+01	6.409E-02	2.115E+01	2.114E+01	4.304E-02
	UPSO	2.096E+01	2.095E+01	5.009E-02	**2.094E+01**	**2.093E+01**	5.023E-02
	CLPSO	**2.072E+01**	**2.072E+01**	5.905E-02	2.105E+01	2.104E+01	4.617E-02
	SRPSO	2.090E+01	2.091E+01	3.948E-02	2.114E+01	2.113E+01	2.862E-02
F_9	χPSO	6.517E+01	6.543E+01	1.239E+01	1.782E+02	1.765E+02	2.498E+01
	DMSPSO	4.524E+01	4.848E+01	1.499E+01	1.006E+02	9.893E+01	2.169E+01
	FIPS	5.401E+01	5.395E+01	1.097E+01	1.550E+02	1.530E+02	1.791E+01
	UPSO	7.719E+01	7.839E+01	1.689E+01	6.326E+01	6.520E+01	1.764E+01
	CLPSO	**0.000E+00**	**0.000E+00**	0.000E+00	**0.000E+00**	**0.000E+00**	0.000E+00
	SRPSO	2.179E+01	3.003E+01	5.521E+00	8.457E+01	8.610E+01	1.707E+01

discussed earlier, the CEC2005 benchmark functions fall into four different categories, in this evaluation the performance on each category is discussed next.

Performance on Unimodal Functions ($\mathbf{F_1}$ to $\mathbf{F_5}$): The results for the unimodal functions are presented in table 3.1. From the table, it can be seen that the performance of SRPSO is better than all the selected variant in all the unimodal functions. In both the mean and median performances, the performance of SRPSO is several orders better than the other algorithms. In function F_1, the SRPSO and CLPSO algorithms have converged to the optimum solutions in both the 30D and 50D cases. In function F_2, only SRPSO has converged to the optimum solution in the 30D case. Also it has provided better results several orders less than the other algorithms in the 50D case. The performance of SRPSO in both 30D and 50D functions is several orders better than the other algorithms for F_3 and F_4 where all the algorithms are producing results in the range 10^4 to 10^7 and SRPSO is providing results in the range 10^1 and 10^3. In function F_5, DMSPSO, CLPSO and SRPSO are all producing results in the same order and among them the performance of SRPSO is better than the others. Based on these results, it can be concluded that for both the 30D and 50D functions, SRPSO has provided better mean and median performances and it has successfully outperformed all the other algorithms by a significant margin. Hence, it may be inferred that SRPSO provides significantly better solutions for unimodal functions.

Performance on Basic Multimodal Functions ($\mathbf{F_6}$ to $\mathbf{F_{12}}$): The results for the basic multimodal functions are presented in tables 3.1 and 3.2. In the tables, it is shown that SRPSO has achieved the best performances for both the 30D and 50D cases in 4 out of the 7 functions. In function F_6, the performance of SRPSO and CLPSO are several orders better than the other algorithms. Both CLPSO and SRPSO have produced results in 10^1 range whereas all the other algorithms have a performance in the range 10^3 to 10^7. Among CLPSO and SRPSO, CLPSO has the best performance in 30D case whereas SRPSO has the best performance in 50D case. In function F_7, the

Table 3.2: Performance on 30D and 50D CEC2005 Benchmark Functions F_{10} to F_{18}

Func.	Algorithm	30-Dimensional			50-Dimensional		
		Median	Mean	StD.	Median	Mean	STD.
F_{10}	χPSO	8.756E+01	8.665E+01	1.850E+01	1.837E+02	1.816E+02	3.686E+01
	DMSPSO	7.797E+01	7.999E+01	2.003E+01	1.673E+02	1.662E+02	2.167E+01
	FIPS	1.545E+02	1.525E+02	2.533E+01	3.868E+02	3.928E+02	3.625E+01
	UPSO	1.526E+02	1.589E+02	5.514E+01	1.453E+02	1.442E+02	4.564E+01
	CLPSO	8.008E+01	8.023E+01	1.495E+01	2.183E+02	2.173E+02	2.000E+01
	SRPSO	**4.293E+01**	**4.510E+01**	6.807E+00	**8.173E+01**	**8.690E+01**	2.589E+01
F_{11}	χPSO	2.802E+01	2.797E+01	2.487E+00	5.874E+01	5.817E+01	3.433E+00
	DMSPSO	2.929E+01	2.903E+01	2.259E+00	5.795E+01	5.774E+01	2.256E+00
	FIPS	2.662E+01	2.688E+01	2.641E+00	5.363E+01	5.343E+01	3.789E+00
	UPSO	3.164E+01	3.140E+01	4.692E+00	2.910E+01	2.954E+01	4.439E+00
	CLPSO	2.548E+01	2.526E+01	1.854E+00	5.263E+01	5.268E+01	2.212E+00
	SRPSO	**1.177E+01**	**1.120E+01**	1.002E+00	**2.148E+01**	**2.191E+01**	3.150E+00
F_{12}	χPSO	1.128E+04	1.872E+04	2.137E+04	2.896E+05	3.293E+05	1.922E+05
	DMSPSO	5.375E+04	7.843E+04	6.836E+04	1.212E+05	1.659E+05	1.461E+05
	FIPS	4.679E+04	5.185E+04	3.213E+04	2.771E+05	2.929E+05	1.490E+05
	UPSO	7.752E+04	8.984E+04	5.430E+04	6.052E+04	7.135E+04	4.785E+04
	CLPSO	1.293E+04	1.324E+04	4.162E+03	8.996E+04	8.949E+04	2.001E+04
	SRPSO	**2.145E+03**	**2.448E+03**	1.354E+03	**9.657E+03**	**1.960E+04**	8.888E+03
F_{13}	χPSO	5.389E+00	5.691E+00	1.775E+00	1.471E+01	1.543E+01	4.475E+00
	DMSPSO	9.962E+00	1.127E+01	5.622E+00	7.812E+00	8.219E+00	2.218E+00
	FIPS	9.604E+00	9.641E+00	1.730E+00	2.583E+01	2.638E+01	3.451E+00
	UPSO	8.489E+00	9.231E+00	4.563E+00	6.236E+00	6.478E+00	2.102E+00
	CLPSO	1.979E+00	**1.888E+00**	3.977E-01	7.093E+00	7.074E+00	6.947E-01
	SRPSO	**1.533E+00**	2.270E+00	9.308E-01	**4.778E+00**	**4.732E+00**	7.309E-01
F_{14}	χPSO	1.211E+01	1.205E+01	4.311E-01	2.193E+01	2.190E+01	4.937E-01
	DMSPSO	1.212E+01	1.207E+01	6.594E-01	2.246E+01	2.241E+01	2.289E-01
	FIPS	1.237E+01	1.234E+01	3.198E-01	2.196E+01	2.190E+01	3.078E-01
	UPSO	1.289E+01	1.283E+01	4.178E-01	**1.282E+01**	**1.274E+01**	4.817E-01
	CLPSO	1.248E+01	1.248E+01	3.051E-01	2.214E+01	2.212E+01	2.642E-01
	SRPSO	**1.187E+01**	**1.117E+01**	2.554E+00	2.095E+01	2.084E+01	5.508E-01
F_{15}	χPSO	4.036E+02	3.929E+02	8.708E+01	4.513E+02	4.198E+02	8.718E+01
	DMSPSO	5.105E+02	5.222E+02	8.016E+01	3.694E+02	3.778E+02	6.679E+01
	FIPS	4.536E+02	4.556E+02	8.414E+01	5.175E+02	5.174E+02	5.406E+01
	UPSO	5.373E+02	5.305E+02	9.060E+01	5.087E+02	4.843E+02	1.137E+02
	CLPSO	**4.116E+01**	**5.623E+01**	5.212E+01	**1.301E+02**	**1.422E+02**	5.276E+01
	SRPSO	3.330E+02	3.050E+02	9.521E+01	2.970E+02	2.929E+02	9.531E+01
F_{16}	χPSO	1.512E+02	1.867E+02	1.055E+02	1.679E+02	1.872E+02	5.268E+01
	DMSPSO	2.250E+02	2.916E+02	1.683E+02	1.419E+02	1.670E+02	8.193E+01
	FIPS	3.271E+02	3.414E+02	1.081E+02	3.227E+02	3.296E+02	6.967E+01
	UPSO	3.827E+02	3.865E+02	1.416E+02	2.909E+02	3.287E+02	1.336E+02
	CLPSO	1.413E+02	1.453E+02	3.171E+01	1.967E+02	1.969E+02	3.751E+01
	SRPSO	**1.030E+02**	**1.372E+02**	1.059E+01	**7.647E+01**	**1.263E+02**	1.070E+02
F_{17}	χPSO	1.881E+02	2.136E+02	8.914E+01	2.834E+02	2.890E+02	7.083E+01
	DMSPSO	2.300E+02	2.988E+02	1.675E+02	2.163E+02	2.413E+02	7.257E+01
	FIPS	4.333E+02	4.502E+02	1.420E+02	4.371E+02	4.576E+02	9.022E+01
	UPSO	3.486E+02	3.875E+02	1.322E+02	3.100E+02	3.497E+02	1.259E+02
	CLPSO	2.084E+02	2.134E+02	3.624E+01	2.767E+02	2.822E+02	3.860E+01
	SRPSO	**1.507E+02**	**1.802E+02**	2.945E+01	**1.569E+02**	**1.929E+02**	1.045E+02
F_{18}	χPSO	9.746E+02	9.388E+02	7.088E+01	9.376E+02	9.392E+02	8.816E+00
	DMSPSO	9.249E+02	9.372E+02	3.011E+01	9.300E+02	9.334E+02	1.156E+01
	FIPS	1.055E+03	1.052E+03	2.219E+01	1.070E+03	1.069E+03	1.346E+01
	UPSO	1.051E+03	1.047E+03	3.627E+01	1.035E+03	1.033E+03	3.041E+01
	CLPSO	9.132E+02	8.993E+02	7.031E+01	9.430E+02	9.414E+02	1.894E+01
	SRPSO	**8.543E+02**	**8.451E+02**	3.292E+01	**8.460E+02**	**8.459E+02**	1.347E+00

mean deviation of all the algorithm from the global optimum is in order of 10^3 in both 30D and 50D cases. The performance of SRPSO here is exceptionally well compared to all other selected algorithms as it has converged closer to the optimum several orders lower than others. The performance of all the algorithms is almost identical in both the cases of function F_8, where CLPSO and UPSO are providing slightly better result in 30D and 50D respectively. The performance of SRPSO has not been significantly better in function F_9 where CLPSO has outperformed all the other algorithms by reaching the optimum solution in both 30D and 50D cases. Here, the mean deviation of SRPSO is of the order of 10^1 similar to other algorithms. In functions F_{10} to F_{12}, SRPSO has successfully provided the best results in both the 30D and 50D cases and it has converged to a value one order lower than the other algorithms. Overall, it has been observed that SRPSO performs better in the 50D case, successfully providing better results in 5 out of the 7 functions. Further, in 30D case the performance of SRPSO and CLPSO are almost similar.

Performance on Expanded Multimodal Functions (F_{13} and F_{14}): The performance on these two functions are mixed as shown in table 3.2. All the algorithms have exhibited similar performances by providing results in the same order. Among them, SRPSO is better in 30D case of function F_{14} and 50D case of function F_{13}. Further, CLPSO and UPSO have provided better results 30D case of function F_{13} and 50D case of function F_{14} respectively.

Performance on Hybrid Composition Functions (F_{15} and F_{25}): Tables 3.2 and 3.3 contains the results for the hybrid composition functions. From the table, one can see that the performance of SRPSO is better in both 30D and 50D cases compared to all the selected algorithms in functions F_{16}, F_{17}, F_{18}, F_{19}, F_{20} and F_{25}. Also SRPSO has performed similar to the best algorithm in function F_{21}. It should be noted that for both the 30D and 50D cases, all the selected algorithms are providing solutions in the same range in all the hybrid composition functions with a few exceptions. Among them, the performance of SRPSO and CLPSO are marginally better than others. In the case of function F_{15}, CLPSO has the best performance in both cases followed closely by SRPSO. In function F_{22}, SRPSO has shown better performance on 30D whereas DMSPSO has provided the best solution in 50D case. The performance of CLPSO is better than other on functions F_{23} and F_{24} followed by SRPSO, χPSO and DMSPSO respectively. Based on these results, it can be concluded that the performance of SRPSO is better than other selected PSO variants on the hybrid composition functions followed by the CLPSO algorithm.

To summarize, from the results on the 30D and 50D CEC2005 benchmark functions one can conclude that SRPSO better performance than other algorithms in all the four groups of functions. Furthermore, the performance has been exceptionally well on a few functions (F_2, F_3, F_4, F_7 and F_{12}) where SRPSO has converged to a solution several orders lower than the other algorithms. Furthermore, it has also been observed that the SRPSO algorithms has provided solutions in the same order for both 30D and 50D cases in 22 out of the 25 functions. This shows that the algorithm is robust over dimensionality increase of a problem. Therefore, the performance of SRPSO on the entire 25 benchmark functions can be termed as the best among all the other selected variants. As a whole, the best performances are observed by SRPSO as follows:

- **30D Median Case:** 19 out of 25 benchmark functions.
- **30D Mean Case:** 18 out of 25 benchmark functions.
- **50D Median Case:** 18 out of 25 benchmark functions.
- **50D Mean Case:** 17 out of 25 benchmark functions.

Mixed behaviour has been observed by the other algorithms because of their characteristics. Since CLPSO and DMSPSO are designed only for multimodal functions, their performance is better only in those problems. Similarly, UPSO and χPSO perform well

Table 3.3: Performance on 30D and 50D CEC2005 Benchmark Functions F_{19} to F_{25}

Func.	Algorithm	30-Dimensional			50-Dimensional		
		Median	Mean	StD.	Median	Mean	STD.
F_{19}	χPSO	9.761E+02	9.355E+02	6.813E+01	9.346E+02	9.383E+02	1.234E+01
	DMSPSO	9.191E+02	9.328E+02	2.703E+01	9.297E+02	9.315E+02	9.374E+00
	FIPS	1.047E+03	1.049E+03	1.873E+01	1.070E+03	1.070E+03	1.603E+01
	UPSO	1.040E+03	1.049E+03	4.636E+01	1.027E+03	1.028E+03	3.344E+01
	CLPSO	9.140E+02	9.102E+02	1.853E+01	9.435E+02	9.418E+02	1.317E+01
	SRPSO	**8.406E+02**	**8.376E+02**	1.707E+01	**8.455E+02**	**8.399E+02**	5.761E+01
F_{20}	χPSO	9.756E+02	9.553E+02	5.665E+01	9.371E+02	9.393E+02	1.197E+01
	DMSPSO	9.281E+02	9.394E+02	2.991E+01	9.299E+02	9.303E+02	8.262E+00
	FIPS	1.049E+03	1.050E+03	1.811E+01	1.062E+03	1.065E+03	1.335E+01
	UPSO	1.043E+03	1.044E+03	2.626E+01	1.027E+03	1.026E+03	2.788E+01
	CLPSO	9.132E+02	9.119E+02	8.572E+00	9.430E+02	9.438E+02	4.918E+00
	SRPSO	**8.406E+02**	**8.330E+02**	3.287E+01	**8.761E+02**	**8.996E+02**	1.273E+02
F_{21}	χPSO	5.098E+02	5.893E+02	2.082E+02	1.018E+03	1.019E+03	3.643E+00
	DMSPSO	1.098E+03	1.037E+03	1.558E+02	1.011E+03	1.011E+03	2.700E+00
	FIPS	1.172E+03	1.093E+03	1.633E+02	1.209E+03	1.196E+03	5.787E+01
	UPSO	1.162E+03	1.021E+03	2.099E+02	7.525E+02	8.542E+02	2.499E+02
	CLPSO	**5.000E+02**	**5.000E+02**	0.000E+00	**5.000E+02**	**5.000E+02**	0.000E+00
	SRPSO	**5.000E+02**	**5.000E+02**	0.000E+00	**5.000E+02**	5.233E+02	8.080E+01
F_{22}	χPSO	1.024E+03	1.026E+03	2.429E+01	9.095E+02	9.149E+02	2.018E+01
	DMSPSO	9.261E+02	9.344E+02	5.295E+01	**8.997E+02**	**9.059E+02**	1.972E+01
	FIPS	1.121E+03	1.126E+03	2.782E+01	1.200E+03	1.202E+03	2.191E+01
	UPSO	1.106E+03	1.109E+03	4.488E+01	1.085E+03	1.086E+03	4.747E+01
	CLPSO	9.603E+02	9.609E+02	1.477E+01	9.912E+02	9.917E+02	7.240E+00
	SRPSO	**5.191E+02**	**5.221E+02**	1.098E+01	9.072E+02	9.081E+02	1.411E+01
F_{23}	χPSO	5.068E+02	5.277E+02	1.018E+02	1.019E+03	1.019E+03	3.335E+00
	DMSPSO	1.097E+03	1.063E+03	1.192E+02	1.011E+03	1.011E+03	2.830E+00
	FIPS	1.177E+03	1.072E+03	1.936E+02	1.210E+03	1.191E+03	9.395E+01
	UPSO	1.156E+03	1.030E+03	2.002E+02	1.049E+03	9.335E+02	2.551E+02
	CLPSO	**5.000E+02**	**5.000E+02**	0.000E+00	**5.000E+02**	**5.000E+02**	0.000E+00
	SRPSO	7.342E+02	7.881E+02	9.033E+01	5.391E+02	5.779E+02	1.101E+02
F_{24}	χPSO	**2.000E+02**	**2.000E+02**	0.000E+00	1.029E+03	1.018E+03	5.740E+01
	DMSPSO	9.769E+02	9.851E+02	5.687E+01	1.025E+03	1.025E+03	2.067E+00
	FIPS	1.268E+03	1.254E+03	5.409E+01	1.282E+03	1.283E+03	1.049E+01
	UPSO	1.141E+03	9.806E+02	3.183E+02	5.298E+02	6.952E+02	3.888E+02
	CLPSO	**2.000E+02**	**2.000E+02**	0.000E+00	**2.000E+02**	**2.000E+02**	4.950E-03
	SRPSO	2.080E+02	2.105E+02	4.178E+00	2.200E+02	2.211E+02	1.977E+00
F_{25}	χPSO	1.750E+03	1.750E+03	7.509E+00	1.682E+03	1.682E+03	5.316E+00
	DMSPSO	1.639E+03	1.640E+03	8.368E+00	1.675E+03	1.676E+03	4.880E+00
	FIPS	1.780E+03	1.781E+03	1.046E+01	1.866E+03	1.866E+03	7.076E+00
	UPSO	1.778E+03	1.778E+03	1.256E+01	1.769E+03	1.771E+03	1.389E+01
	CLPSO	1.659E+03	1.659E+03	4.102E+00	1.701E+03	1.702E+03	2.610E+00
	SRPSO	**1.266E+03**	**1.256E+03**	6.399E+00	**4.612E+02**	**6.761E+02**	4.020E+02

only in a few of the problems. To further analyse the performance deeply, a rank based analysis followed by a statistical comparison has been performed and are presented in the next sections.

3.3.5 Rank based analysis

The mean and median performances of the algorithms on the set of benchmark functions are ranked for determine the best performing algorithm. The algorithm producing the lowest value on a particular function is given the best rank (rank 1) followed by ranking in ascending order. The standard competition ranking scheme has been adopted for this analysis. In competition ranking, equally performing algorithms are given the same rank and the algorithm coming after them will get a rank with a gap i.e. if 3 algorithms get rank 1 then the 4th algorithm will be assigned rank 4.

Rank based analysis of median performance for both 30D and 50D cases is presented in table 3.4. The table contains individual ranking on benchmark functions for each algorithm, the average rank for all the functions and the final overall rank. In both the cases, one can see that SRPSO has consistently achieved the best rank most of the time and has remained in the top 3 ranks throughout the experimentation. The only exception is the 50D case for function F_2 where SRPSO has fallen to 4th rank. Based on the average rank, the following order of performance has been observed.

- **30D Case:** SRPSO, CLPSO, χPSO, DMSPSO, FIPS and UPSO respectively.
- **50D Case:** SRPSO, CLPSO, DMSPSO, UPSO, χPSO and FIPS respectively.

As stated earlier, SRPSO has achieved the best rank in 19 and 18 functions out of 25 functions for 30D and 50D respectively whereas CLPSO (the second best ranked algorithm) has only achieved the best rank in 8 and 6 functions out of the 25 25 functions for 30D and 50D respectively. This clearly highlights the efficiency and robustness of SRPSO over other PSO variants.

Table 3.5 provides the rank based analysis of the mean performances for both 30D and 50D cases. Similar to the median ranking, the algorithms here are also ranked individually on the 25 benchmark functions. Using these ranking, the average ranks and the final ranks are reported. The average ranks on the mean performances suggest a

Table 3.4: Rank based analysis of Median performance for 30D and 50D Cases

Dim	Algo.	Individual Ranking of Benchmark Functions																									Avg.
		F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}	F_{16}	F_{17}	F_{18}	F_{19}	F_{20}	F_{21}	F_{22}	F_{23}	F_{24}	F_{25}	RANK(R)
30	χPSO	3	1	2	3	4	3	4	2	5	4	4	2	3	2	3	3	2	4	4	4	3	4	2	1	4	3.04(3)
	DMSPSO	4	3	3	2	3	4	2	4	3	2	5	5	6	3	5	4	4	3	3	3	4	2	4	4	2	3.48(4)
	FIPS	5	6	5	6	5	6	5	5	4	6	3	4	6	5	4	4	5	6	6	6	6	6	5	6	6	5.24(5)
	UPSO	6	5	6	5	6	5	6	6	6	5	6	6	4	6	6	6	5	5	5	5	5	5	6	5	5	5.44(6)
	CLPSO	1	4	4	4	2	1	3	1	1	3	2	3	2	5	1	2	3	2	2	2	1	3	1	1	3	2.28(2)
	SRPSO	1	1	1	1	1	2	1	2	2	1	1	1	1	1	2	1	1	1	1	1	1	1	3	3	1	1.32(1)
50	χPSO	3	2	3	4	4	2	3	4	6	4	6	6	5	3	4	3	4	3	3	3	5	3	4	5	3	3.8(5)
	DMSPSO	4	3	2	2	2	5	2	3	4	3	5	4	4	6	3	2	2	2	2	2	4	1	3	4	2	3.04(3)
	FIPS	6	6	6	5	6	6	6	6	5	6	4	5	6	4	6	6	6	6	6	6	6	6	6	6	6	5.72(6)
	UPSO	5	4	4	3	5	4	5	1	2	2	2	2	2	1	5	5	5	5	5	5	3	5	5	3	5	3.72(4)
	CLPSO	1	5	5	6	3	3	4	2	1	5	3	3	3	5	1	4	3	4	4	4	1	4	1	1	4	3.2(2)
	SRPSO	1	1	1	1	1	1	1	4	3	1	1	1	1	2	2	1	1	1	1	1	1	2	2	2	1	1.4(1)

similar outcome as that of the median ranking. From the table, one can easily infer that SRPSO has performed much better than all the other algorithms. Based on the average rank, the following order of performance has been observed.

- **30D Case:** SRPSO, CLPSO, χPSO, DMSPSO, FIPS and UPSO respectively.
- **50D Case:** SRPSO, CLPSO & DMSPSO, UPSO, χPSO and FIPS respectively.

Based on the above rank (both median and mean) studies, it is evident that SRPSO performs better than the other PSO variants in all the 25 benchmark problems. The rank based analysis has made it evident that SRPSO is a significantly better performing algorithm on the set of benchmark function. Therefore, a statistical comparison has been conducted to understand the significance of the performance of SRPSO over other algorithms.

3.3.6 A statistical comparison

This section provides a statistical comparison to highlight the significance of SRPSO over other selected PSO variants. For statistical comparison, the non-parametric Friedman test on the average rank followed by the pairwise post-hoc Bonferroni-Dunn test [128] has been conducted. Friedman test is a non-parametric statistical test, for testing the

Table 3.5: Rank based analysis of Mean performance for 30D and 50D Cases

Dim	Algo.	Individual Ranking of Benchmark Functions																									Avg.
		F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}	F_{16}	F_{17}	F_{18}	F_{19}	F_{20}	F_{21}	F_{22}	F_{23}	F_{24}	F_{25}	RANK(R)
30	χPSO	3	2	3	3	4	3	4	2	5	4	4	3	3	2	3	3	3	4	4	4	3	4	2	1	3	3.16(3)
	DMSPSO	4	4	2	2	3	6	2	4	3	2	5	5	6	3	5	4	4	3	3	3	5	2	5	5	2	3.68(4)
	FIPS	5	6	5	6	5	5	5	5	4	5	3	4	5	4	4	5	6	6	5	6	6	6	6	6	6	5.16(5)
	UPSO	6	5	6	5	6	4	6	6	6	6	6	6	4	6	6	6	5	5	6	5	4	5	4	4	5	5.32(6)
	CLPSO	1	3	4	4	2	1	3	1	1	3	2	2	1	5	1	2	2	2	2	2	1	3	1	1	4	2.16(2)
	SRPSO	1	1	1	1	1	2	1	3	2	1	1	1	2	1	2	1	1	1	1	1	1	1	3	3	1	1.4(1)
50	χPSO	3	2	3	4	4	4	3	3	6	4	6	6	5	3	4	3	4	3	3	3	5	3	5	4	3	3.84(5)
	DMSPSO	4	3	2	2	2	5	2	3	4	3	5	4	4	6	3	2	2	2	2	2	4	1	4	5	2	3.12(2)
	FIPS	6	6	6	5	6	6	6	6	5	6	4	5	6	3	6	6	6	6	6	6	6	6	6	6	6	5.68(6)
	UPSO	5	4	5	3	5	3	5	1	2	2	2	2	2	1	5	5	5	5	5	5	3	5	3	3	5	3.64(4)
	CLPSO	1	5	4	6	3	2	4	2	1	5	3	3	3	5	1	4	3	4	4	4	1	4	1	1	4	3.12(2)
	SRPSO	1	1	1	1	1	1	1	3	3	1	1	1	1	2	2	1	1	1	1	1	2	2	2	2	1	1.4(1)

performance difference between several related samples. The test will identify the differences between the performance of the algorithms over the set of benchmark functions. The pairwise post-hoc Bonferroni-Dunn test provides valid inferences about differences in the performance of multiple group over multiple comparisons. The test will provide a conclusive inference that the mean performance of SRPSO is better than the other algorithms.

First, the Friedman test using average rank of different algorithms on 30-dimensional (mean and median) and 50-dimensional (mean and median) has been conducted. The average ranks have already been reported in the previous section in tables 3.4 and 3.5. The Friedman null hypothesis states that all the related samples are equal performers. Accordingly, based on the null hypothesis the average ranks of all the algorithms must be equal or else the null hypothesis will be rejected.

The F-statistic value for a confidence level of 95% for the 6 algorithms tested over 25 benchmark functions can be obtained from the F-table which gives the critical F-value as 2.2745. According to the the null hypothesis, if the computed F-score is greater than the critical F-value then the null hypothesis can be rejected. The computed Friedman statistic (F-Score) based on average rank for 30D and 50D median and mean are given in table 3.6. In the table, the computed F-scores are 64.7453, 55.4708, 28.8485 and 28.4442 for 30D median, 30D mean, 50D median and 50D mean respectively. Since the

Table 3.6: Average Rank Difference and Statistics for $\alpha = 0.05$ significance level

Avg Rank difference w.r.t. SRPSO	30D Median	30D Mean	50D Median	50D Mean
χPSO	1.72	1.76	2.4	2.44
DMSPSO	2.16	2.28	1.64	1.72
FIPS	3.92	3.76	4.32	4.28
UPSO	4.12	3.92	2.32	2.24
CLPSO	0.96	0.76	1.8	1.72
F-score (Friedman)	64.7453	55.4708	28.8485	28.4442
Critical Difference (Benforroni-Dunn)	0.5545	0.5863	0.7193	0.7209

computed F-scores are much greater than the critical F-value (2.2745), one can reject the null hypothesis and it can be inferred that the average rank of the algorithms used in this study are statistically different.

Since the Friedman null hypothesis is rejected, a pairwise post-hoc Bonferroni-Dunn test [128] is conducted to highlight the performance of SRPSO over other algorithms. The minimum required difference to signify the performance of one algorithm over another is also present in the table 3.6. If the difference between average ranks of SRPSO and other algorithms is greater than the critical difference (CD) then the performance of SRPSO is statistically significant than the other algorithm. The average rank difference of all the selected algorithms with respect to SRPSO is greater than the critical difference in all the cases considered in the study. This suggests that the performance of SRPSO is statistically better than all the PSO variants considered in this study.

To summarize, the performance evaluation on the CEC2005 benchmark functions and the comparative rank based and statistical analysis clearly highlight hat the human learning strategies incorporated in the SRPSO algorithm have provided better performances. The two proposed human self-learning principles inspired strategies, namely, self-regulating inertia weight and self-perception based selection of directions from the global best position have indeed enhanced the convergence characteristics of the PSO algorithm. It is necessary here to evaluate the computational complexity of the SRPSO algorithm to test its efficiency.

3.3.7 An analysis on computational complexity of the SRPSO algorithm

In this section, the computational complexity and CPU time requirements of the proposed SRPSO algorithm are analysed. The order of complexity of any algorithm describes its efficiency. It is evident that if an algorithm maintains the order of complexity of the standard PSO algorithm and at the same time provide fast, robust and better solutions then it may be concluded that a significant performance improvement has been achieved. Table 3.7 contains the order of complexity of the selected PSO variants and the SRPSO algorithm. Since the selected algorithms have different initialization, evaluation and updation schemes, therefore, the order of complexity in terms of O notation for the process of initialization, evaluation and update has been observed. Finally, the overall complexity of each of the algorithms has been reported in table 3.7. Here, N, n and D represent the population size, number of neighbours and the total dimensions respectively. From the table, one can conclude that SRPSO has maintained the order of complexity of the PSO algorithm at every stage and hence has the overall complexity of O(ND).

If one compares the velocity update equations of PSO given in equation 1 and SRPSO given in equation 9 and 10, it can be seen that SRPSO has the same order of complexity as that of PSO. Next, the CPU clock time requirements of the PSO and SRPSO algorithms in solving the benchmark function are calculated. The following equation as suggested in [95] has been utilized for evaluating the computational time:

$$Burden(F) = \frac{SRPSO_t(F) - PSO_t(F)}{PSO_t(F)} \tag{3.8}$$

A unimodal (F_1) and two multimodal (F_9 and F_{12}) are selected for computational time analysis of SRPSO. The average computational time of SRPSO on the three functions are calculated as 1.16785s, 2.00559s and 2.2293s respectively. Next, the average

Table 3.7: Order of Complexity for selected PSO algorithms

Algorithm	Initialize	Evaluate	Update	Overall
χPSO	O(ND)	O(ND)	O(ND)	O(ND)
DMSPSO	O(ND)	O(ND+NnD)	O(ND+NnD)	O(NnD)
FIPS	O(ND)	O(ND)	O(ND)	O(ND)
UPSO	O(ND)	O(ND)	O(ND+NnD)	O(NnD)
CLPSO	O(ND)	O(ND)	O(ND)	O(ND)
SRPSO	O(ND)	O(ND)	O(ND)	O(ND)

computational time of PSO is calculated which are 1.26856s, 2.12609s and 2.33098s respectively. on the same three functions. Using equation 3.8, it is found that SRPSO reduces the burden by 7.9% for F_1, 5.66% for F_9 and 4.36% for F_{12} compared to the PSO algorithms. This suggests that SRPSO has reduced the CPU clock time requirements of the PSO algorithm together with providing better convergence. This is due to the self-perception strategy introduced in SRPSO as social perception p_{id}^{so} will use the social cognitive part lesser number of times as compared to the standard PSO algorithm.

If one closely studies the strategy update mechanism of the selected PSO variant, it will be noted that none of the algorithms has used the increasing inertia weight strategy for the particles. In SRPSO, there is an increasing inertia weight for the best particle that is assumed as the potential candidate nearer to the desired solution. This increasing inertia weight is accelerating the exploration process and as a result there is a faster convergence compared to other PSO variants.

Optimization is the most essential ingredient in any engineering problem design. In this chapter, a new self-regulating learning strategy has been introduced in the PSO algorithm. The algorithm is named as the Self-Regulating Particle Swarm Optimization (SRPSO) algorithm. The algorithm makes use of two human learning principles

- Self-regulation, where the best particle regulates its inertia weight to perform fast and better exploration and
- Self-perception, where all the other particles apply perception based selection of directions from the global best position for intelligent partial social exploitation.

With the help of the strategies, SRPSO has successfully

- Maintained the complexity of the PSO algorithm and reduced the overall computational time.
- Provided faster and much better convergence on diverse CEC2005 benchmark problems.
- Statistically outperformed the widely accepted state-of-the-art PSO variants.

It has been observed that the performance of SRPSO on a few multimodal and hybrid composition functions has not been the best. This suggests some performance improvement in the algorithm is required. In SRPSO, only the best performing particle is searching with a different learning strategy while the rest of the particles are using the same strategy. This might have prevented the least performing particles from acquiring a better learning strategy as they do not get the required support from the better performing particles to get a directional update towards potentially better search areas and eventually contribute towards convergence. The algorithm is only using human self-learning principles and missing the key concept of socially shared information processing from the human learning psychology.

In the next chapter, human socially shared information processing concepts are integrated in the PSO algorithm to develop a more robust and efficient human learning principles inspired PSO algorithm.

Chapter 4

Dynamic Mentoring and Self-Regulation based Particle Swarm Optimization Algorithm

In the previous chapter, human self-learning principles inspired PSO algorithm was introduced referred to as Self-Regulating Particle Swarm Optimization algorithm. In human learning psychology it is mentioned that human beings possess multiple hierarchical interrelated layers of information processing which enable them to access their performance through socially shared information [129]. Hence, they collaborate with each other efficiently for attaining the maximum gain from the environment. This provides them the essential information for performance improvement by successfully mitigating the deficiencies through collaborative expert learning mechanism. This chapter focus on development of human social intelligence inspired PSO algorithm. First, the socially shared information processing is described in the context of mentoring based learning. Next, the development of algorithm inspired from social learning principles is described in detail.

4.1 Basic Concepts of Human Social Learning Principles

Human beings are known to be adaptive to wide range of environments through the help of improvisational intelligence possessed by them from birth [130]. It has always been argued that human beings are smarter and acquire better cognitive abilities than any other creatures in the world [131]. Using their cognitive abilities, humans interact with each other in an effective manner and exploit the environment in a better way. The interaction can either be collaborative/ cooperative or competitive. In competitive nature, a human is always selfish and tends to hide his knowledge form others whereas the collaborative interaction strengthens the social bond among humans through promoting open-minded and selfless behaviours. This helps them to overcome their deficiencies through social cooperation and they achieve productive mutual benefits. A conscientious human is one who effectively utilize his self and social cognizance, regulate his strategies, monitor his performance, effectively perform information sharing and make better decisions [20]. Hence, socially shared information among humans provides them better understanding of the environment and guides them towards the desired goals.

In social learning, the oldest and most effective way of learning adopted by humans is gaining knowledge from other individuals. It has been stated in [21] that human beings transfer knowledge and gain knowledge from each other through the process of teaching and learning. The individuals who provide the necessary information to others are often considered as the elite learners having efficient problem solving skills. Social learning provides a positive learning environment to the learners. There are several ways in which an individual interact with another for information sharing. Some of these socially shared information processing principles are discussed next.

Exemplar based Learning: This learning technique refers to learning by following a role model. Here, an individual considered as the most effective learner acts as the role model for all other individuals. In most cases, the exemplar is not a single person and there are several exemplars that are being followed by others. This learning process often provides better understanding but it lacks in self-cognition. Here, other individuals consider the exemplars as their guides and totally rely on them for performance improvement. This is often a one way learning where the exemplars are providing the information and learners are following them. Leader follower based learning is also closely related to exemplar based learning where everyone follows an individual. Unlike the exemplars, a leader is always a single person and he prefers to stay at that position to enjoy the luxury of taking all the decisions.

Peer Learning: This is a successful learning technique adopted in human beings especially by the students where learner learn from each other and through their effective interaction achieve some advantageous learning outcomes. In peer learning, all individuals are learner and they interact with each other to learn. Here, any individual is not superior to another and all are considered as having the same level of intelligence. With the help of social sharing all individuals get the opportunity of sharing their ideas, experiences and knowledge with others for achieving mutual benefit.

Peer Teaching: Peer teaching, or peer tutoring, is a far more influential strategy in which the more efficient learners guide the less efficient learners for their performance improvement. Here, the individuals assigned the teaching responsibilities are often those who have prior knowledge of the situation in which the peer learner are currently standing. The teachers are responsible for proving the necessary information to the learners to enhance their understanding of the problem. In peer teaching, the teachers are often not in the learning mode and they only share their prior knowledge with the peers. This often halts the learning process, therefore an alternate reciprocal peer teaching method has been preferred [132, 133, 134, 135].

One of the powerful and effective method of reciprocal peer teaching is the mentoring based learning scheme. Mentoring learning scheme supports the positive learning and provide individual and collective development [136]. A major advantage of this learning scheme is that it combines several socially shared information processing principles adopted by the humans. It is a peer teaching method combined with peer learning and exemplar based learning strategies as well as it also allows individuals to perform self-learning through intelligent social interaction. This method is more close to human intuition of social learning where humans don't completely imitate each and every social influence encountered in the environment [22]. Instead, they perform self-reflection to identify the proper information for adopting in their learning process. Therefore, the same has been adopted in this thesis for development of a human socially shared principles inspired PSO algorithm.

4.2 The Mentoring based Learning Process

Mentoring is the process of positive learning within a group and the learning scheme is regarded as teaching taken to a deeper level. Mentoring is the process of informal transmission of knowledge among peers where an individual with greater relevant knowledge/ experience in a certain area communicates with a person who is perceived to have less knowledge [137]. In social learning theory [22], mentoring is regarded as the key learning scheme for conceptual development of the learners. Mentoring, applied to any learning environment, boosts the skills of the learners and makes them competent individuals for any challenging learning environment. There are several ways in which humans adopt the mentoring scheme, and among them an effective scheme is the one in which the group learn together. This scheme provides the flexibility to the learners possessing a sufficient amount of skills to learn individually. In the mentoring based learning environment, few individuals with better knowledge acquisition through effective learning skills act as the mentors. Further, all those individuals that are not effective learners and cannot excel individually act as mentees. It is necessary to provide certain guidance to the mentees for enhancing their learning skills which is performed by the mentors. There are also some moderate learners capable of self-regulating their strategies to learn independently without any guidance. Hence, there are three types of learners in the mentoring based learning environments; namely, mentors, mentees and independent learners. Next, the roles and responsibilities of each of the three learners is discussed.

Mentor: Any individual with efficient learning skills can take the role of a mentor. Within a group, the elite learners possessing a better understanding of the situation are considered most suitable for the role of a mentor. In [138], mentors are considered as the trusted guides who are capable to translating their knowledge effectively. The attributes associated with the mentors include the capabilities of being a role model, possessing strong communication skills and being knowledgeable about the learning environment.

Mentee: Any individual who is not capable enough of learning individually can take the role of a mentee. Within a group there are always some individuals who always rely on others for performance improvement; these perfectly fit into the mentee group. The mentees generate some amount of trust in the mentor and this trust describe the amount of information that mentee will take confidently from the mentor.

Independent Learner: Any individual who possess a sufficient amount of skills to learn individually and trust his abilities is an independent learner. The role of these learners is flexible in the learning environment. Through continuous performance assessment, they can either become mentee to mitigate performance deficiencies or become mentor to guide others.

There is a dynamic learning environment in the mentoring based learning scheme. The individuals in each group don't remain there forever. Based on their experiences and performances, any individual can either become a mentor or an independent learner or even a mentee. The independent learners perform the acquisition of knowledge independently and they also don't collaborate with others. The mentors also perform learning independently but they are capable enough to convey the information to others. The mentees cannot learn independently and therefore they are guided by the mentors. For an effective learning environment, it is necessary that there exists a significant amount of trust between the mentor and the mentee. This creates a strong relationship between them and provides confidence to the mentee to learn effectively from the mentor. Both the mentors and mentees are learners in this learning environment and hence mentoring

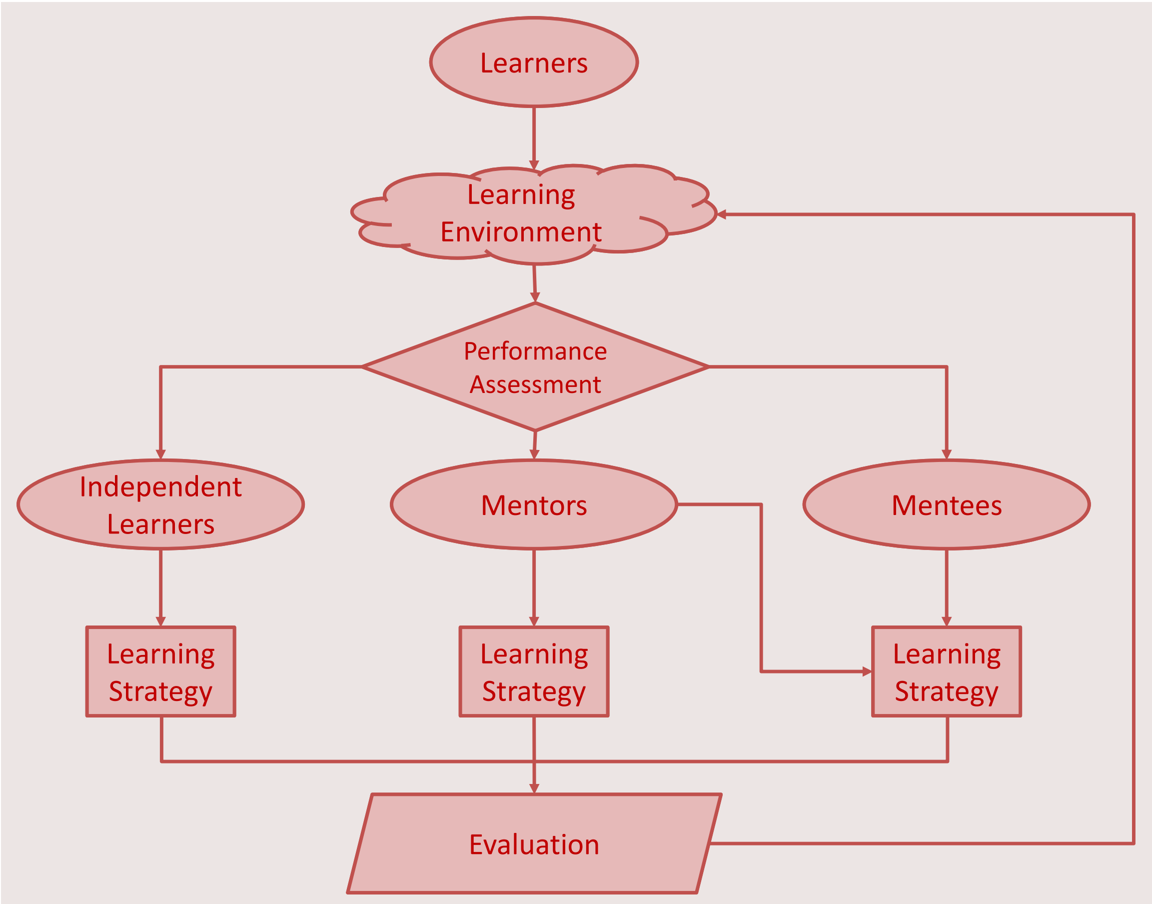

Figure 4.1: The Process of Mentoring based Learning Scheme

based learning is not a one way learning environment. It is in fact an effective learning process that provides equal opportunities for everyone to learn independently and in collaboration to eventually develop better learning abilities.

To clearly illustrate the mentoring based learning scheme in a learning environment, a holistic pictorial view is presented in figure 4.1. The figure provides an insight view of the complete effective learning process implemented through the mentoring based learning scheme. In the figure, it is shown that in any learning environment there are learners that interact with it for knowledge acquisition. The learners apply their skills, take necessary actions and provide the outcome. The performance of these learners is assessed to identify the members of the three groups, viz., mentor, mentee and independent learner. The members in each group have different roles based on their learning abilities and hence they all apply different learning strategies for performance improvement. The improvement in the performance is continuously monitored through the amount of knowledge acquisition and outcome produced by them. This classifies them as a learner in any of the groups. The dynamic nature of the learning environment through continuous performance monitoring regularly change the learners in each group. The mentors are the elite learners; they possess effective learning strategies for efficient outcome. They provide the necessary required guidance to the mentees for enhancing their learning capabilities. The independent learners are often called the moderate learners as they possess sufficient amount of skills and knowledge to satisfactorily provide the desired outcome. They continue to learn independently till the time they achieve satisfactory improvement in their performances. In figure 4.1, the dynamic nature is clearly illustrated through the direction of flow from evaluation back to the learning environment. Therefore, the individuals can take different roles at different occasions based on the following analysis.

- Any individual with degradation in performance will move to the mentee group.
- Any individual with significant performance improvement will move to the mentor group.

The above two points suggest that a degraded performance can move anyone from the group of mentors or independent learners to the group of mentees. Similarly, a significant performance improvement can promote anyone to the mentors group. Such socially shared information processing based learning scheme has been integrated in PSO for better convergence.

4.3 Dynamic Mentoring and Self-Regulation based Particle Swarm Optimization Algorithm

This section presents the detailed description of the proposed human socially shared information processing based PSO algorithm. As described in the previous section, mentoring is an effective learning scheme in social interactive environment that provide individual and mutual performance enhancement. The mentoring process initiates an effective learning environment with the use of confidence and trust among the learners from mentor and mentee groups. This ensures that there is positive learning and mutual benefit for the learners. Within the group there are some individuals who have better cognition on their knowledge and are capable of self-regulating their experiences/ strategies to learn independently. These concepts from the mentoring learning scheme have been taken into account and are incorporated in the PSO algorithm. The new variant is referred to as Dynamic Mentoring and Self-Regulation based Particle Swarm Optimization (DMeSR-PSO) algorithm.

4.3.1 Incorporating the mentoring based learning concepts in PSO

The mentoring based learning scheme has been applied to the particles similar to that as described in figure 4.1. The performance of the particles has been taken into account for their division in the group of mentors, mentees and independent learners. The particle with the best performance i.e. the best fitness value is selected as the best particle. Two different measures with respect to the best particle have been taken into account for division of the particles into the said groups:

- Euclidian distance (S_f),
- Fitness difference (S_{ed}).

Based on the calculated Euclidian distance and fitness difference, the particles are divided into three groups: (i) Mentor (M_r), (ii) Mentee (M_e) and (iii) Independent Learner (I_n). Those particles that are closer to the best particle in terms of both Euclidian distance and fitness difference are considered as the elite particles and they are selected in the Mentor group. Similarly, the particles that are far away from the best particle in terms of both measures are considered as the less efficient learners and are selected in the Mentee group. Finally, the remaining particles are considered as the moderate learners and are selected in the Independent Learner group.

Figure 4.2 provides a pictorial view of the selection of particles in each group. In the figure, there are two circles representing the fitness difference (S_f) and Euclidian distance (S_{ed}) of all the particles and the centre of both circles represent the best particle. The particles are mapped on the scale of 0 to 100 percent where the best particle is at 0% and the outermost circle represents the 100% value. The selection of mentors, mentees and independent learning particles as shown in figure 4.2 is defined as:

$$i = \begin{cases} M_r, & \text{if } S_f \leq \lambda_1 \text{ and } S_{ed} \leq \lambda_2 \\ M_e, & \text{if } S_f > \lambda_3 \text{ and } S_{ed} > \lambda_4 \\ I_n, & \text{if } \lambda_1 < S_f \leq \lambda_3 \text{ and } \lambda_2 < S_{ed} \leq \lambda_4 \end{cases} \tag{4.1}$$

The constants λ_1, λ_2, λ_3 and λ_4 are empirically experimented for the best possible values. After several experimentation, these threshold values are selected as $\lambda_1 = 5\%$, $\lambda_2 = 10\%$, $\lambda_3 = 90\%$ and $\lambda_4 = 50\%$ in the DMeSR-PSO algorithm. The mentoring based learning scheme has been carefully implemented in the PSO algorithm with the proper selection of the threshold values. Using these threshold values, the selection of elite learners and less efficient learners as mentors and mentees has been performed. In any learning environment, all individuals are neither top performers nor they are all least performers. Similar consideration has been taken into account and only those particles that are closer to the best are selected in the mentor group. Similarly, the mentee group consists of all non-performing particles i.e. particles having a much higher fitness value. Further, to achieve better convergence and guide the particles towards the optimum solution, all the particles that are at a distance higher than 50% from the best particles are also included in the mentee group. The justification of selection of these threshold values is provided later in section 4.4.1.

In social learning, the most effective learning outcomes from the mentoring based learning scheme have been achieved by the generation of sufficient amount of confidence

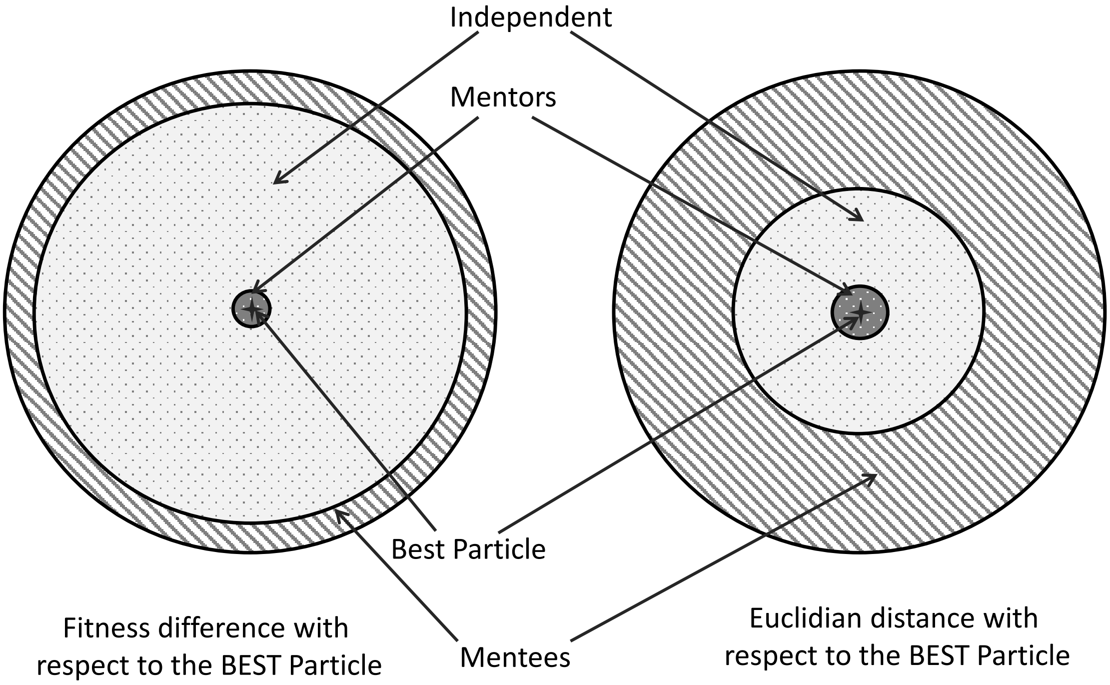

Figure 4.2: Selection of Particles in Mentor, Mentee and Independent Learner Group

and trust among the mentor and mentee [139]. This means that whenever a mentee has confidence on the mentor and he trusts him for knowledge acquisition, the outcome is always favourable and provide mutual benefits. There can be few scenarios arising from the learning scheme as stated below:

- A mentee can trust a mentor more than the other mentors,
- A mentee can trust himself more than other and is confident to learn independently.

To accommodate these concepts in the algorithm, two confidence signals are generated for every dimension of a mentee particle to decide the selection of a mentor particle.

- The self-confidence signal (C_{se}) and
- The mentor-confidence signal (C_{Me})

Both the confidence signals are generated using a uniform random number within the range [0, 1], $\alpha \sim U([0,1])$. The two confidence signals C_{se} and C_{Me} will enable the particle to undergo either self-experience based learning or mentor based learning process. If the confidence signal is high, the particle will learn using his personal best. Furthermore, for a high mentor confidence signal, the particle will learn from a mentor. It is necessary to allow the particles to undergo self-learning and social learning almost equal number of times for intelligent utilization of self and social cognition. To achieve the same, the probability of selection of dimensions to undergo either self-experience based learning or mentor based learning are kept more or less similar. The self-confidence signal is defined as:

$$C_{se} = \begin{cases} 1, & \text{if } \alpha \ > \ 0.5 \\ 0, & \text{otherwise} \end{cases} \tag{4.2}$$

The selection of mentor (S_{Me}) for the guidance of a mentee is defined as:

$$S_{Me} = \begin{cases} 1, & \text{if } C_{se} = 0 \\ 0, & \text{otherwise} \end{cases} \tag{4.3}$$

For a high S_{Me} signal, there will be a selection of mentor for guidance of the particle. For guidance, there is need of development of trust between mentee and mentor. To incorporate the concept of trust, a mentor-confidence signal will be generated for the mentee. A mentee will learn from the selected mentor if there is a high mentor-confidence signal. If the generated signal is low, it will suggest that the trust has not been developed among the mentee and the mentor. Next, a bigger group will be dynamically generated for a selection of more mentors as shown in figure 4.3. The figure illustrated the dynamically expanded mentor group to allow more particles in the mentor group. Here, the mentor group has been extended to $S_f \leq 50\%$ and $S_{ed} \leq 50\%$ for random selection of a new mentor particle.

4.3.2 Learning strategies for the particles in each group

The particles in each group possess varying characteristics and produce different performances. Therefore, it is necessary to apply different learning strategies for the particles belonging to different groups. In the mentor group there are elite particles with efficient

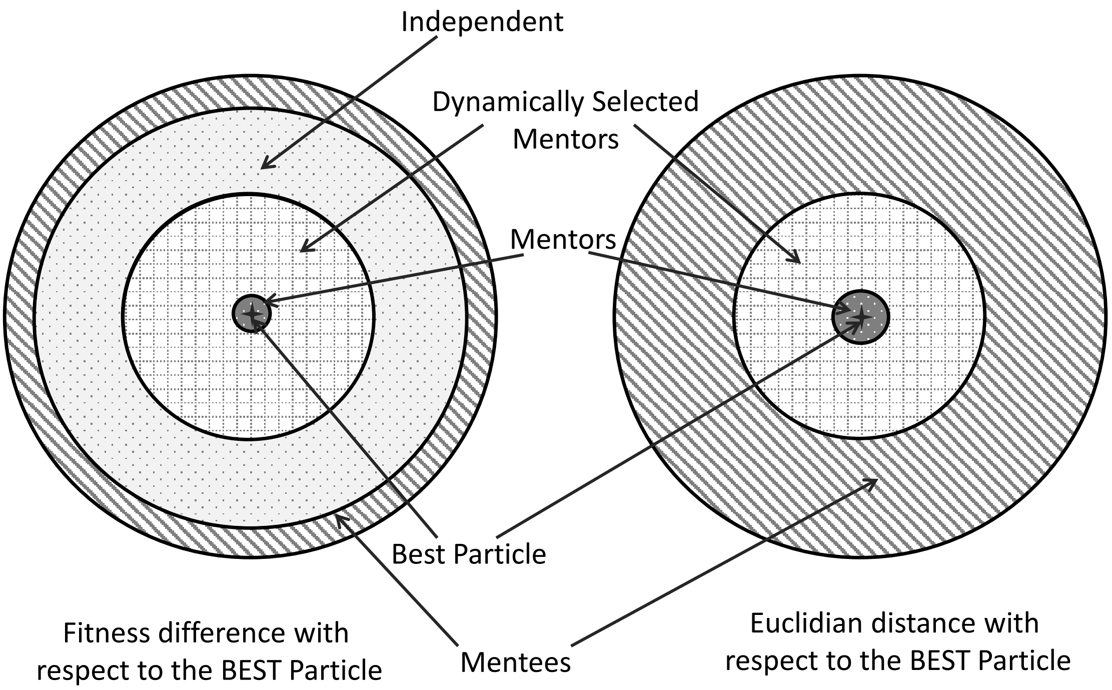

Figure 4.3: Dynamic Expansion of the Mentor Group

performances, hence these particles will be given a more explorative learning strategy to explore the potential areas of the search space. It has already stated earlier that the independent learners are capable of self-regulating their knowledge for achieving the desired goals. Also as stated in [1], independent learners are intelligent enough to perform self-regulation of their experiences for achieving favourable understanding of the environment. The concept of self-regulation has already been introduced in the previous chapter in the proposed SRPSO algorithm [140]. The concepts of self-regulation and self-perception have been adopted as the learning strategy for the independent learners in the DMeSR-PSO algorithm for better convergence. Finally, the mentee group consists of all the less efficient particles with unsatisfactory performance. These particles will perform search using a guidance based learning strategy. Next, the detailed description of the learning strategies in each group is presented.

The Mentor Group: The best particle and the particles closer to it as shown in figure 4.2 are in the mentor group. The best particle being the most efficient one will perform the search using the self-regulating inertia weight strategy and the velocity update equation of the best particle proposed in the SRPSO algorithm in the previous chapter. The other particles, considered as the effective learners will use partial self and social cognition together with a full belief in their search direction. These particles perform the search utilizing more exploration and partial exploitation. The partial self and social cognitions are controlled by β_1 and β_2 respectively. The values for β_1 and β_2 are set as 0.5 to ensure that the particles have less influence of their previous self and social knowledge and have a higher impact of their current search direction. Therefore, the particles will perform search with a strong influence of their current experiences. The velocity update equation of these particles is defined as:

$$V_{id}^{t+1} = \omega_i V_{id}^{t} + c_1 r_1 \beta_1 (P_{id}^{t} - X_{id}^{t}) + c_2 r_2 \beta_2 (P_{gd}^{t} - X_{id}^{t}) \tag{4.4}$$

When the mentor group is dynamically expanded, the particles belonging to independent learners group also come in this group. These particles are not permanent members of the group and are only added to fulfill a particular requirement. Therefore, they perform search using their own learning strategy irrespective of their occurrence in either mentor group or the independent learner group.

The Mentee Group: All the particles far away from the best particle are considered as the lesser efficient particles. These particles require some sort of guidance for performance improvement. As indicated in the previous section, a mentee can trust him more than any other or he may trust a mentor for guidance. These particles perform search either using the self-belief strategy or the social guidance (guided by a mentor) strategy. The velocity update equation for a mentee is given below:

$$V_{id}^{t+1} = \omega_i V_{id}^{t} + c_1 r_1 C_{se} (P_{id}^{t} - X_{id}^{t}) + c_2 r_2 S_{Me} (P_{Med}^{t} - X_{id}^{t}) \tag{4.5}$$

where P_{Med} is the personal best of the selected mentor, $C_{se} = 1$ suggests a self-belief based search and $S_{Me} = 1$ suggest a socially guided search. The particle will only perform search through social guidance or using self-belief because if $C_{se} = 1$ then $S_{Me} = 0$ and vice versa.

The Independent Learner Group: All the particles in this group are capable of learning using their own experiences. These particles have strong belief in their experiences and socially share information based on their perception. These particles utilize the social-awareness based search using self-perception based selection strategy of the SRPSO algorithm.

The positions (X) of all the particles are updated using:

$$X_{id}^{t+1} = X_{id}^{t} + V_{id}^{t+1}, \qquad i = 1, 2, \cdots, N; \qquad d = 1, 2, \cdots, D \tag{4.6}$$

4.3.3 The DMeSR-PSO algorithm

In the DMeSR-PSO algorithm, the particles are divided into three groups consisting of mentors, mentees and independent learners. Further, the mentor group size is dynamically changing to accommodate the trust based relationship between a mentor and a mentee. The particles in each group perform search using different learning strategy. Also, the particles in each group are dynamically changing i.e. in a given iteration a mentor can become an independent learner in the next iteration. The main learning schemes in DMeSR-PSO can be summarized as:

- The dynamic mentoring scheme between the particles of mentor and mentee groups,
- Self-Regulating inertia weight scheme for best particle and
- Self-perception based search for all the particles belonging to the independent learner group.

Incorporating all these learning strategies, the finalised flowchart for DMeSR-PSO is given in figure 4.4.

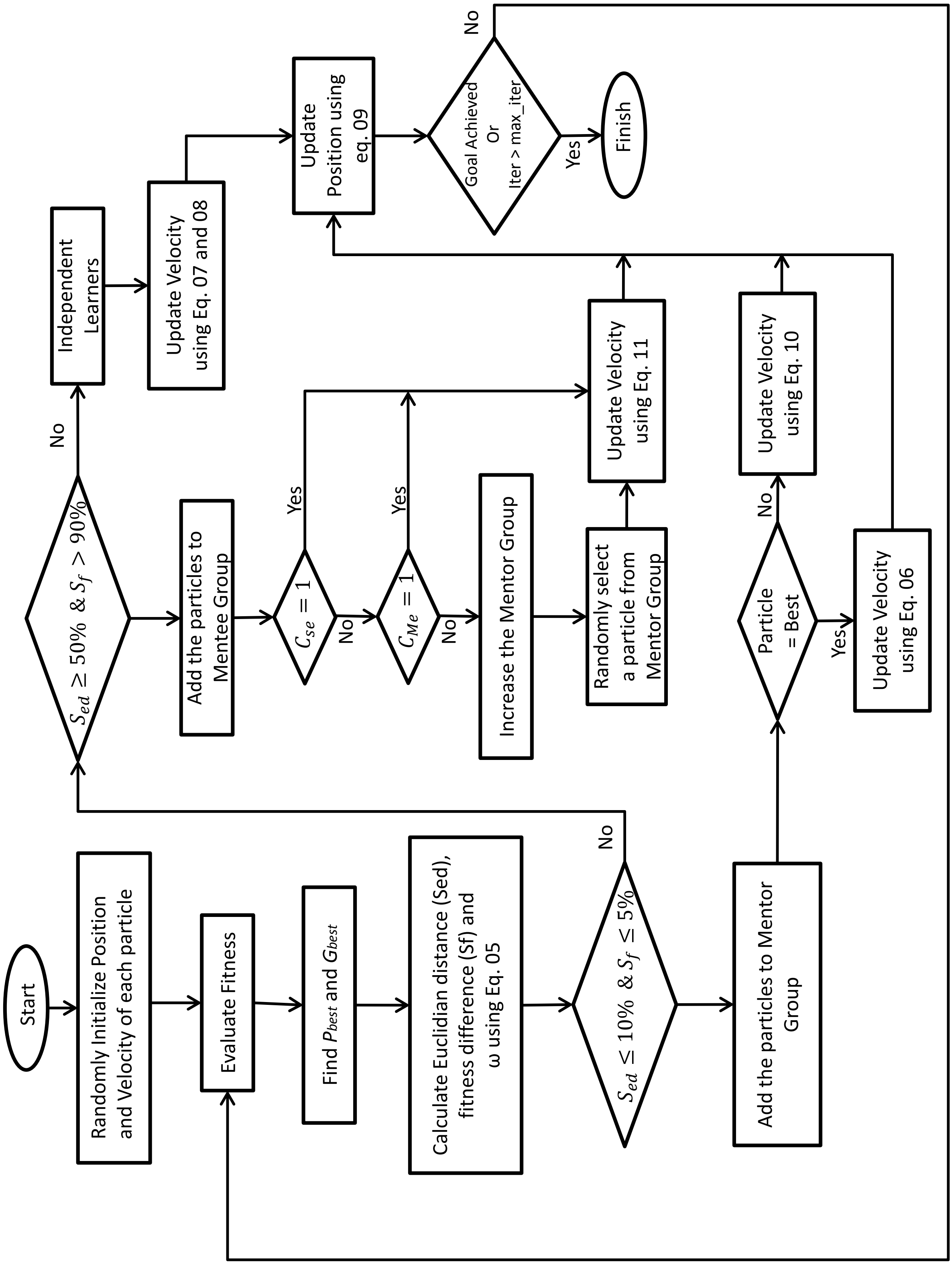

Figure 4.4: DMeSR-PSO Flow Diagram

4.4 Performance Evaluation, Results and Discussion

This section provides the performance analysis of the proposed algorithm. First the analysis of threshold values is provided. Next, the impact of DMeSR-PSO over the convergence of PSO has been studied. Finally, the performance has been compared with widely accepted evolutionary algorithms.

4.4.1 Analysis of Threshold Values λ_1, λ_2, λ_3 and λ_4

The main criteria for the mentoring based learning scheme implementation in the algorithm is the selection of particles in the three groups: Mentor, mentee and independent learner. The convergence of particles towards better solution is dependent on proper selection. The DMeSR-PSO algorithm performance has a greater influence of the selection of the particles in each group. As discussed earlier in section 4.3.1, there are four different threshold values λ_1, λ_2, λ_3 and λ_4 used to define the selection of particles in the three groups. In this study, three different benchmark functions with different characteristics from the CEC2005 [59] are selected for analysing the impact of different threshold values on the convergence of DMeSR-PSO. The three benchmark functions are:

- A unimodal function: F_4, Shifted Schwefel's Problem 1.2 with Noise in Fitness,
- A multimodal function: F_{12}, Schwefel's Problem 2.13,
- A hybrid composition function: F_{23}, Non-Continuous Rotated Hybrid Composition Function.

Table 4.1: Different Percentage Group Values for Mentors and Mentees

Mentor Group	Mentee Group
$M_r(A)$: $S_f \leq 5\%$ and $S_{ed} \leq 5\%$	$M_e(A)$: $S_f > 85\%$ and $S_{ed} > 40\%$
$M_r(B)$: $S_f \leq 5\%$ and $S_{ed} \leq 10\%$	$M_e(B)$: $S_f > 85\%$ and $S_{ed} > 50\%$
$M_r(C)$: $S_f \leq 10\%$ and $S_{ed} \leq 10\%$	$M_e(C)$: $S_f > 90\%$ and $S_{ed} > 50\%$
$M_r(D)$: $S_f \leq 10\%$ and $S_{ed} \leq 15\%$	$M_e(D)$: $S_f > 90\%$ and $S_{ed} > 60\%$
$M_r(C)$: $S_f \leq 15\%$ and $S_{ed} \leq 15\%$	$M_e(E)$: $S_f > 95\%$ and $S_{ed} > 60\%$

Three different values for each threshold have been considered for testing the performance on each problem. The values for

- λ_1 are 5%, 10% and 15%,
- λ_2 are 5%, 10% and 15%,
- λ_3 are 85%, 90% and 95% and
- λ_4 are 40%, 50% and 60%.

Using these different threshold values 5 different groups of Mentor and Mentee are formed as shown in Table 4.1. A set of 25 different experiments have been conducted using the combination of mentor and mentee group from the table. The experimental setup for the analysis includes:

- Swarm size = Dimension = 30,
- Function Evaluations = 300000,
- Total runs = 25

Table 4.2 contains the mean and standard deviation performance of all the combinations of mentor and mentee group. From the table, it is clear that the combination defined by $M_r(B)$ $M_e(C)$ group as $\lambda_1 = 5\%$, $\lambda_2 = 10\%$, $\lambda_3 = 90\%$ and $\lambda_4 = 50\%$ is providing the better convergence compared to all other settings of the threshold values. Next, to properly visualize the performance of all the settings the mean error fitness values are plotted against different combinations as shown in figure 4.5. The figure provides a better analysis for the different selected groups. It is clear from the figure that six different group selection, $M_r(B)$ $M_e(B)$, $M_r(B)$ $M_e(C)$, $M_r(B)$ $M_e(D)$, $M_r(C)$ $M_e(A)$, $M_r(C)$ $M_e(C)$ and $M_r(C)$ $M_e(E)$ are providing better solution compared to the other group selections. Among them, the group defined by $M_r(B)$ $M_e(C)$ is providing the best solutions. Hence, the threshold values defined by the group $M_r(B)$ $M_e(C)$ as $\lambda_1 = 5\%$, $\lambda_2 = 10\%$, $\lambda_3 = 90\%$ and $\lambda_4 = 50\%$ are selected and kept constant for all the experimental studies. Next analysis is on the impact of proposed learning schemes on the convergence towards the optimum solution.

4.4.2 Impact of dynamic mentoring and self-regulation

In DMeSR-PSO there are different learning strategies adopted by particles from different group. In this section, the impact of two main strategies i.e. dynamic mentoring and self-regulation on the PSO algorithm are first tested separately on a benchmark function. Finally, the combined effect on the performance using the DMeSR-PSO algorithm has been observed. Similar to the analysis performed in chapter 3, a function from each group of the CEC2005 [59] benchmark set are selected. The performance has been evaluated 25 time on the 30D problem using a swarm size of 30. The median performance of each algorithm has been plotted against the total number of iterations. The convergence graph of median performances of standard PSO, Dynamic Mentoring based PSO (DMePSO), SRPSO and DMeSR-PSO are presented in figure 4.6. The figure presents the performance of the said algorithms on the four benchmark functions F_3, F_6, F_{14} and F_{22}. From the figure, one can see that when the dynamic mentoring scheme and the self-regulation scheme are applied separately to the standard PSO algorithm there is a significant amount of performance improvement. Using either of the proposed strategies, PSO has converged to a point far better than its original variant. When both the strategies are combined together in the form of DMeSR-PSO, faster convergence closer to the global optimum solution has been achieved in all the categories of benchmark function. Similar analysis can be performed on all the benchmark functions where more or less same performance has been observed.

Next, to have a clear analysis of the impact of DMeSR-PSO on the convergence over SRPSO, a performance comparison on the 30D and 50D CEC2005 benchmark functions

Table 4.2: Comparative analysis of Different Threshold Values λ_1, λ_2, λ_3 and λ_4

Group	F_4		F_{12}		F_{23}	
	Mean	STD.	Mean	STD.	Mean	STD.
$Mr(A)Me(A)$	5.912E+01	2.133E+01	8.722E+03	7.888E+03	7.037E+02	6.73E+01
$Mr(A)Me(B)$	2.437E+01	1.103E+01	8.172E+03	7.880E+02	7.363E+02	6.75E+01
$Mr(A)Me(C)$	2.937E+01	1.733E+01	6.841E+03	7.980E+02	6.722E+02	7.21E+01
$Mr(A)Me(D)$	1.914E+01	1.023E+01	7.725E+03	8.620E+02	6.811E+02	4.17E+01
$Mr(A)Me(E)$	4.124E+01	2.136E+01	5.852E+03	9.950E+02	7.092E+02	7.04E+01
$Mr(B)Me(A)$	2.231E+01	1.024E+01	4.346E+03	8.780E+02	7.478E+02	8.58E+01
$Mr(B)Me(B)$	1.425E+01	4.327E+00	3.231E+03	1.035E+03	6.240E+02	6.84E+01
$Mr(B)Me(C)$	**1.287E+01**	5.067E+00	**2.295E+03**	1.006E+03	**5.337E+02**	9.00E+01
$Mr(B)Me(D)$	1.324E+01	8.365E+00	2.721E+03	9.774E+02	5.849E+02	6.90E+01
$Mr(B)Me(E)$	2.121E+01	1.233E+01	4.222E+03	1.010E+03	6.032E+02	5.16E+01
$Mr(C)Me(A)$	1.510E+01	1.021E+01	3.343E+03	8.780E+02	6.222E+02	6.10E+01
$Mr(C)Me(B)$	3.565E+01	1.255E+01	5.851E+03	9.950E+02	6.375E+02	7.08E+01
$Mr(C)Me(C)$	2.937E+01	1.433E+01	5.460E+03	1.550E+03	6.670E+02	4.57E+01
$Mr(D)Me(B)$	4.025E+01	1.101E+01	6.335E+03	2.340E+03	6.337E+02	4.32E+01
$Mr(D)Me(C)$	3.625E+01	1.521E+01	4.560E+03	1.040E+03	6.164E+02	5.30E+01
$Mr(D)Me(D)$	4.013E+01	2.136E+01	5.342E+03	7.980E+02	6.801E+02	6.33E+01
$Mr(D)Me(E)$	3.266E+01	1.237E+01	7.070E+03	7.580E+02	7.125E+02	5.88E+01
$Mr(E)Me(A)$	5.733E+01	2.155E+01	6.142E+03	8.950E+02	7.221E+02	6.58E+01
$Mr(E)Me(B)$	4.124E+01	2.137E+01	6.690E+03	1.130E+03	7.033E+02	6.55E+01
$Mr(E)Me(C)$	3.857E+01	1.937E+01	6.312E+03	9.650E+02	6.550E+02	5.76E+01
$Mr(E)Me(D)$	5.533E+01	1.935E+01	7.791E+03	6.850E+02	6.934E+02	4.01E+01

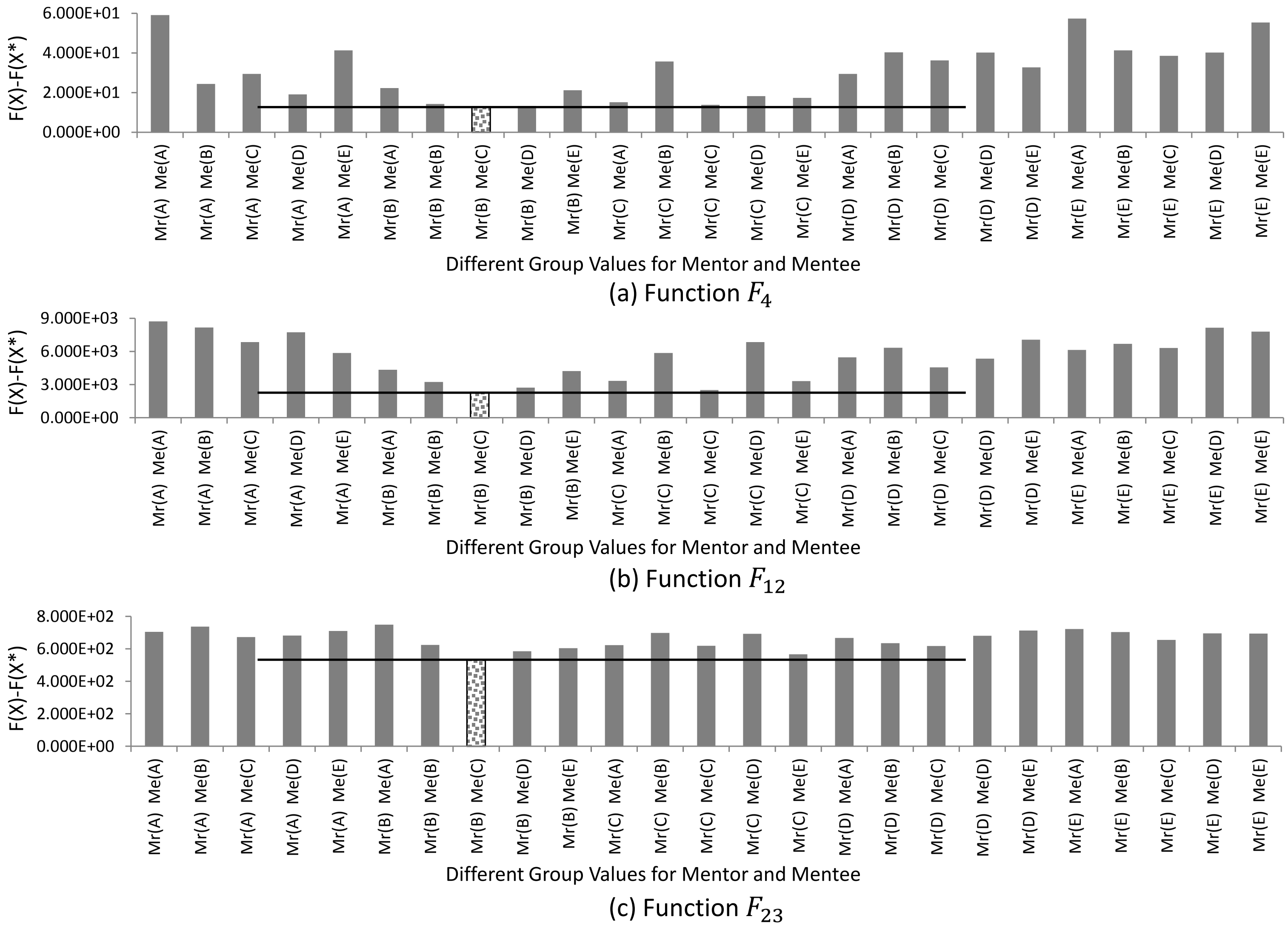

Figure 4.5: Analysis of Different Threshold Values: (a) Impact on Unimodal Function, (b) Impact on Multimodal Function and (c) Impact on Hybrid Composition Function

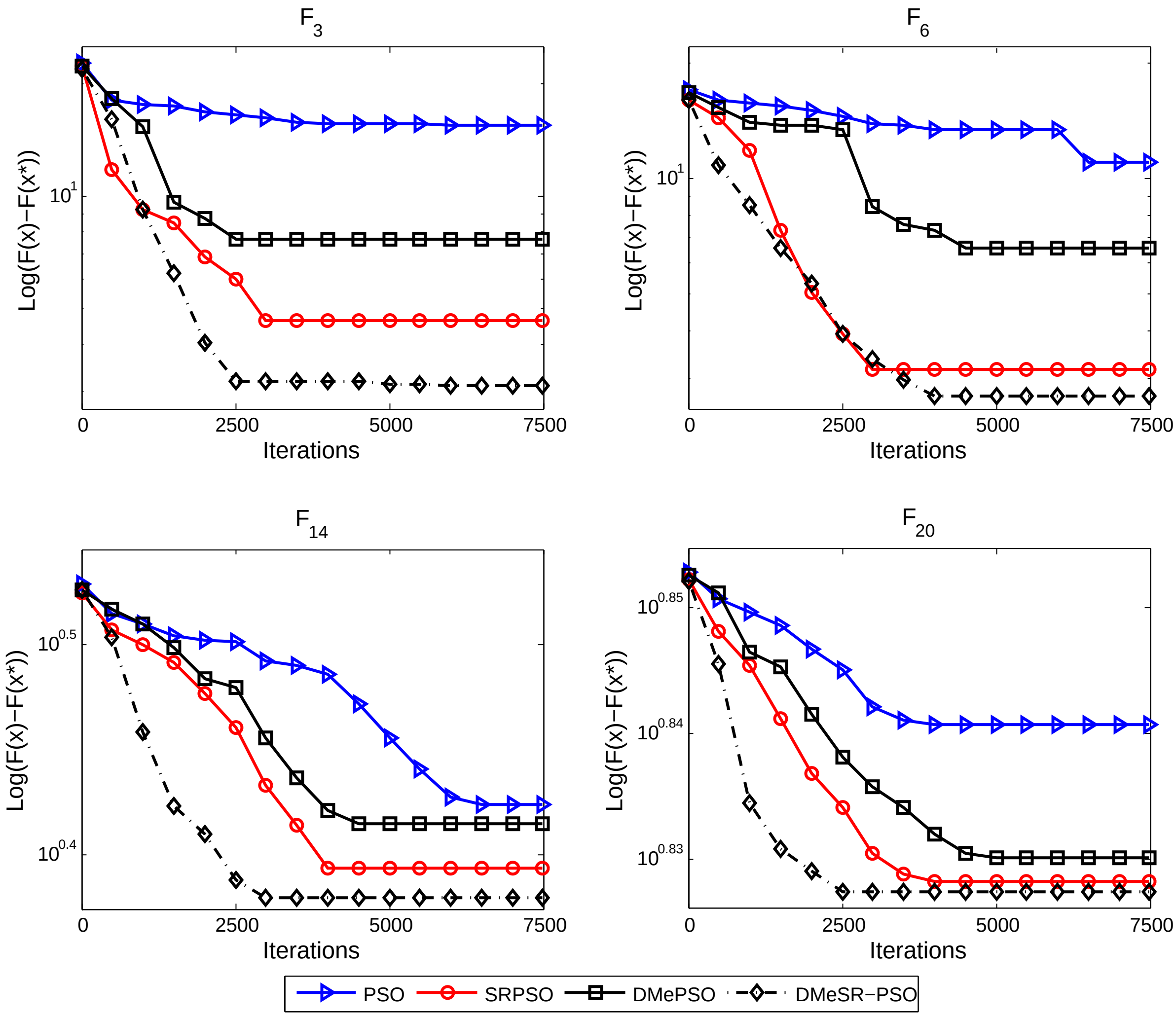

Figure 4.6: Impact of the Dynamic Mentoring and Self-Regulation strategies

have been conducted. All the experimental setup are kept same as that of SRPSO provided in previous chapter, section 3.3.3. The experiments have been conducted 25 times and in every run the function evaluations were set to 10000×D. The median, mean and standard deviation fitness values of the two algorithms are provided in tables 4.3 and 4.4. It is evident from the two tables that DMeSR-PSO has provided better performance in all the functions compared to the SRPSO algorithm. There has been significant performance improvement in the following functions where DMeSR-PSO has provided solutions one order less than SRPSO.

- 50D F_2,
- 30D and 50D F_3,
- 30D and 50D F_7 and
- 30D F_{16}

This suggests an improvement in performance achieved by the DMeSR-PSO algorithm in terms of providing solutions closer to the optimum values. Hence, it can be concluded that the concept of dynamic mentoring and self-regulation applied to the PSO algorithm has indeed enhanced its searching capabilities and convergence characteristics.

4.4.3 Selected algorithms for comparison and parameter settings

In chapter 3, the performance of SRPSO was compared with the well-known PSO variants whereby SRPSO significantly outperformed the compared algorithms by providing solutions closer to the true optimum value. In the previous section, it has been observed that human social intelligence inspired DMeSR-PSO algorithm has shown superior performance over SRPSO algorithm. Therefore, in this evaluation, performance of DMeSR-PSO has been compared with well-known evolutionary algorithms on all the CEC2005 benchmark functions. The following algorithms have been selected for comparison:

i **Differential Evolution (DE) [141]:** The classical DE algorithm has been tested on CEC2005 by [142] as the algorithm has proved to be highly efficient for solving real valued test functions.

ii **Guided Differential Evolution (G-DE) [143]:** An improved variant of DE with an idea of preserving previous known better directions that generated good results in the next generation.

Table 4.3: Performance Comparison of SRPSO and DMeSR-PSO on Functions F_1 to F_{13}

Function	Performance	30D		50D	
		SRPSO	DMeSR-PSO	SRPSO	DMeSR-PSO
F_1	Median	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	Mean	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
	STD.	0.000E+00	0.000E+00	0.000E+00	0.000E+00
F_2	Median	**0.000E+00**	**0.000E+00**	5.870E-03	**2.070E-06**
	Mean	**0.000E+00**	**0.000E+00**	1.931E-02	**8.164E-06**
	STD.	0.000E+00	0.000E+00	1.639E-02	1.030E-05
F_3	Median	1.038E+02	**2.235E+01**	3.638E+02	**6.443E+01**
	Mean	1.365E+02	**4.414E+01**	3.927E+02	**6.706E+01**
	STD.	2.199E+02	1.563E+02	2.780E+02	8.899E+00
F_4	Median	2.349E+01	**1.313E+01**	2.373E+03	**1.509E+03**
	Mean	3.205E+01	**1.287E+01**	2.559E+03	**1.685E+03**
	STD.	1.145E+01	5.067E+00	8.771E+02	5.073E+02
F_5	Median	**1.983E+03**	2.029E+03	**3.646E+03**	4.568E+03
	Mean	**2.134E+03**	2.143E+03	**3.725E+03**	4.557E+03
	STD.	2.741E+02	1.721E+02	5.784E+02	7.466E+02
F_6	Median	2.375E+01	**1.493E+01**	3.547E+01	**1.945E+01**
	Mean	2.446E+01	**1.506E+01**	5.754E+01	**2.368E+01**
	STD.	5.738E+00	2.915E+00	3.313E+01	1.674E+01
F_7	Median	4.452E-03	**1.770E-04**	4.396E-01	**2.093E-02**
	Mean	6.263E-03	**2.506E-03**	5.197E-01	**2.766E-02**
	STD.	4.521E-03	4.918E-03	1.757E-01	1.746E-02
F_8	Median	2.090E+01	**2.089E+01**	2.114E+01	**2.105E+01**
	Mean	2.091E+01	**2.089E+01**	2.113E+01	**2.104E+01**
	STD.	3.948E-02	3.865E-02	2.862E-02	1.835E-02
F_9	Median	2.179E+01	**1.965E+01**	8.457E+01	**4.731E+01**
	Mean	3.003E+01	**2.140E+01**	8.610E+01	**5.226E+01**
	STD.	5.521E+00	1.239E+01	1.707E+01	1.195E+01
F_{10}	Median	4.293E+01	**2.717E+01**	8.173E+01	**5.675E+01**
	Mean	4.510E+01	**3.217E+01**	8.690E+01	**7.574E+01**
	STD.	6.807E+00	1.508E+01	2.589E+01	3.621E+01
F_{11}	Median	1.177E+01	**7.771E+00**	2.148E+01	**1.325E+01**
	Mean	1.120E+01	**7.592E+00**	2.191E+01	**1.301E+01**
	STD.	1.002E+00	7.854E-01	3.150E+00	1.907E+00
F_{12}	Median	2.145E+03	**2.065E+03**	9.657E+03	**8.462E+03**
	Mean	2.448E+03	**2.295E+03**	1.960E+04	**1.111E+04**
	STD.	1.354E+03	1.006E+03	8.888E+03	8.915E+03
F_{13}	Median	1.533E+00	**1.230E+00**	4.778E+00	**3.310E+00**
	Mean	2.270E+00	**1.467E+00**	4.732E+00	**3.285E+00**
	STD.	9.308E-01	4.221E-01	7.309E-01	4.425E-01

Table 4.4: Performance Comparison of SRPSO and DMeSR-PSO on Functions F_{14} to F_{25}

		30D		50D	
Function	Performance	SRPSO	DMeSR-PSO	SRPSO	DMeSR-PSO
F_{14}	Median	1.187E+01	**1.099E+01**	2.095E+01	**1.497E+01**
	Mean	1.117E+01	**1.114E+01**	2.084E+01	**1.492E+01**
	STD.	2.554E+00	7.122E-01	9.508E-01	1.092E-01
F_{15}	Median	3.330E+02	**1.850E+02**	2.970E+02	**1.892E+02**
	Mean	3.050E+02	**1.877E+02**	2.929E+02	**2.003E+02**
	STD.	9.521E+01	1.244E+01	9.531E+01	5.818E+01
F_{16}	Median	1.030E+02	**5.909E+01**	7.647E+01	**7.454E+01**
	Mean	1.372E+02	**7.208E+01**	1.263E+02	**1.007E+02**
	STD.	1.059E+01	1.647E+01	1.070E+02	3.523E+01
F_{17}	Median	1.507E+02	**1.228E+02**	1.569E+02	**1.416E+02**
	Mean	1.802E+02	**1.217E+02**	1.929E+02	**1.768E+02**
	STD.	2.945E+01	2.116E+01	1.045E+02	3.105E+01
F_{18}	Median	8.543E+02	**8.292E+02**	8.460E+02	**8.446E+02**
	Mean	8.451E+02	**8.290E+02**	8.459E+02	**8.447E+02**
	STD.	3.292E+01	3.765E-01	1.347E+00	1.840E-01
F_{19}	Median	8.406E+02	**8.295E+02**	8.455E+02	**8.446E+02**
	Mean	8.376E+02	**8.280E+02**	8.399E+02	**8.447E+02**
	STD.	1.707E+01	1.937E+00	5.761E+01	1.054E-01
F_{20}	Median	8.406E+02	**8.290E+02**	8.761E+02	**8.444E+02**
	Mean	8.330E+02	**8.280E+02**	8.996E+02	**8.445E+02**
	STD.	3.287E+01	2.635E+00	1.273E+02	1.560E-01
F_{21}	Median	**5.000E+02**	**5.000E+02**	**5.000E+02**	**5.000E+02**
	Mean	**5.000E+02**	**5.000E+02**	5.233E+02	**5.000E+02**
	STD.	0.000E+00	0.000E+00	8.080E+01	0.000E+00
F_{22}	Median	5.191E+02	**5.000E+02**	9.072E+02	**5.000E+02**
	Mean	5.221E+02	**5.093E+02**	9.081E+02	**5.000E+02**
	STD.	1.098E+01	7.248E+01	1.411E+01	2.847E-06
F_{23}	Median	7.342E+02	**5.247E+02**	**5.391E+02**	**5.391E+02**
	Mean	7.881E+02	**5.337E+02**	5.779E+02	**5.391E+02**
	STD.	9.033E+01	9.001E+00	1.101E+02	1.735E-04
F_{24}	Median	2.080E+02	**2.000E+02**	2.200E+02	**2.000E+02**
	Mean	2.105E+02	**2.000E+02**	2.211E+02	**2.000E+02**
	STD.	4.178E+00	0.000E+00	1.977E+00	0.000E+00
F_{25}	Median	1.266E+03	**2.000E+02**	4.612E+02	**2.000E+02**
	Mean	1.256E+03	**2.116E+02**	6.761E+02	**2.136E+02**
	STD.	6.399E+00	2.929E+01	4.020E+02	1.863E+01

iii **Flexible Evolutionary Algorithm (FEA) [144]:** An algorithm consisting of several features adopted from different evolutionary algorithms. Different features have provided FEA higher flexibility in solving optimization problems.

iv **Real-Coded Evolutionary Algorithm (RC-EA) [145]:** RC-EA is a steady-state, population-based search algorithm derived from population-based algorithm generator. The developed algorithm has efficiently solved selected problems and it is less sensitive to noise and rotation of the landspace.

v **Estimation of Distribution Algorithm (EDA) [146]:** EDA is an algorithm based on probabilistic modeling where the new generation is sampled from a probability distribution.

vi **Hybrid Real-Coded Genetic Algorithm (Hy-RCGA) [147]:** The algorithm is a hybrid of global and local versions of RCGA to effectively utilize global and local search at the same time. The algorithm has been proven to be highly robust as compared to the other RCGAs.

vii **Real-Coded Memetic Algorithm (RCMA) [148]:** RCMA is a hybridized version of RCGA with local search techniques. The algorithm has successfully maintained the balance between exploration and exploitation of the search space and provided promising solutions.

viii **Covariance Matrix Adaptation Evolution Strategy (CMAES) [8]:** One of the most efficient and promising optimization algorithm developed using the covariance matrix of the normal mutation search distribution. A newer, more efficient version of CMAES referred to as restart CMAES with increasing population (IPOP-CMA-ES) has been tested on CEC2005 in [149] which has provided highly competitive solutions to non-linear, non-separable problems.

Furthermore, the SRPSO algorithm has also been included in the performance evaluation. All the guidelines of CEC2005 has been followed in the experimentation and the results of the selected algorithms have been taken from their respective articles. Further, the settings are kept same as discussed earlier in section 4.4.2.

4.4.4 Experimental results and performance comparison

Since most of the selected algorithms have been systematically evaluated on the 30D benchmark functions from the CEC2005 problem suite, therefore, the comparative analysis has been performed on the 30D functions. The performance (median, mean and standard deviation values) for the 10 algorithms is presented in tables 4.5 - 4.9.

From the tables, one can see that the compared algorithms have varying performances on the benchmark functions. The DE, RCMA and Hy-RCGA algorithms have shown better performances in some unimodal and multimodal functions. Further, the IPOP-CMAES algorithm has outperformed all the algorithms in the category unimodal and multimodal functions. Similarly, DMeSR-PSO algorithm has achieved the best performance in the expanded multimodal and hybrid composition functions. In 3 functions (F_1 , F_2 and F_{21}), identical performances have been observed where most of the selected algorithms have provided similar results.

Overall, DMeSR-PSO has achieved the best median performance in 13 out of the 25 functions. Furthermore, it has provided the best mean performances in 16 benchmark functions. Also, DMeSR-PSO has achieved the best performance in terms of both mean and median fitness values in 13 out of the 25 benchmark functions whereas IPOP-CMAES has provided better mean and median performances in 11 functions. This indicates that the performance of IPOP-CMAES and DMeSR-PSO are comparable on the set of benchmark functions whereby IPOP-CMAES is well-suited for the unimodal and basic multimodal functions and DMeSR-PSO is well suited for the expanded multimodal and hybrid composition functions. To have a clear illustration of performance significance of

Table 4.5: Performance Comparison among selected Evolutionary Algorithms on F_1-F_6 CEC2005 Benchmark Functions

Algorithm	$\mathbf{F_1}$			$\mathbf{F_2}$		
	Median	Mean	STD.	Median	Mean	STD.
DE	**0.00E+00**	**0.00E+00**	0.00E+00	2.22E-02	3.33E-02	4.90E-03
RCMA	**0.00E+00**	**0.00E+00**	0.00E+00	8.75E-06	8.72E-06	7.47E-07
IPOP-CMAES	**0.00E+00**	**0.00E+00**	0.00E+00	**0.00E+00**	**0.00E+00**	0.00E+00
RC-EA	1.23E-02	7.97E-01	2.49E+00	6.95E-04	4.40E-01	1.52E+00
FEA	7.52E+00	7.06E+01	1.20E+02	9.25E+02	9.82E+02	4.01E+02
Hy-RCGA	**0.00E+00**	**0.00E+00**	0.00E+00	**0.00E+00**	**0.00E+00**	0.00E+00
G-DE	**0.00E+00**	4.54E-01	1.08E+00	2.46E+01	5.11E+01	6.77E+01
EDA	4.89E+01	4.88E+01	9.57E+00	1.62E+02	1.61E+02	2.78E+01
SRPSO	**0.00E+00**	**0.00E+00**	0.00E+00	**0.00E+00**	**0.00E+00**	0.00E+00
DMeSR-PSO	**0.00E+00**	**0.00E+00**	0.00E+00	**0.00E+00**	**0.00E+00**	0.00E+00
	$\mathbf{F_3}$			$\mathbf{F_4}$		
	Median	Mean	STD.	Median	Mean	STD.
DE	7.29E+05	6.92E+05	2.04E+05	**6.68E+00**	1.52E+01	1.81E+01
RCMA	7.64E+05	8.77E+05	5.81E+04	2.56E+01	3.97E+01	8.55E+00
IPOP-CMAES	**0.00E+00**	**0.00E+00**	0.00E+00	1.93E+01	1.11E+04	3.02E+04
RC-EA	3.05E+00	3.67E+02	1.35E+03	4.39E+03	4.80E+03	3.44E+03
FEA	7.06E+06	7.45E+06	2.93E+06	9.82E+03	9.82E+03	3.42E+03
Hy-RCGA	2.21E+03	3.11E+03	2.33E+03	1.42E+01	1.68E+01	9.96E+00
G-DE	3.38E+06	3.81E+06	1.94E+06	2.29E+02	6.59E+02	1.35E+03
EDA	3.66E+06	3.75E+06	9.09E+05	1.38E+03	1.28E+03	2.91E+02
SRPSO	1.04E+02	1.36E+02	2.20E+02	2.35E+01	3.20E+01	1.15E+01
DMeSR-PSO	2.24E+01	4.41E+01	1.56E+02	1.31E+01	**1.29E+01**	5.07E+00
	$\mathbf{F_5}$			$\mathbf{F_6}$		
	Median	Mean	STD.	Median	Mean	STD.
DE	9.35E+01	1.70E+02	1.84E+02	1.29E+01	2.51E+01	2.90E+01
RCMA	2.13E+03	2.18E+03	7.83E+01	1.90E+01	4.95E+01	1.93E+01
IPOP-CMAES	**0.00E+00**	**0.00E+00**	0.00E+00	**0.00E+00**	**0.00E+00**	0.00E+00
RC-EA	8.13E+03	8.34E+03	1.04E+03	3.14E+02	1.21E+03	1.83E+03
FEA	9.82E+03	9.91E+03	2.09E+03	1.23E+02	2.58E+02	3.06E+02
Hy-RCGA	2.37E+02	3.33E+02	2.47E+02	4.51E-08	2.60E-07	4.32E-07
G-DE	2.23E+03	2.35E+03	8.61E+02	3.19E+01	5.62E+01	4.50E+01
EDA	6.26E+03	6.19E+03	1.16E+03	1.69E+05	1.83E+05	6.24E+04
SRPSO	1.98E+03	2.13E+03	2.74E+02	2.37E+01	2.45E+01	5.74E+00
DMeSR-PSO	2.03E+03	2.14E+03	1.72E+02	1.49E+01	1.51E+01	2.91E+00

Table 4.6: Performance Comparison among selected Evolutionary Algorithms on F_7-F_{12} CEC2005 Benchmark Functions

Algorithm	$\mathbf{F_7}$			$\mathbf{F_8}$		
	Median	Mean	STD.	Median	Mean	STD.
DE	1.20E-08	2.96E-03	5.55E-03	2.10E+01	2.10E+01	5.11E-02
RCMA	9.86E-03	1.33E-02	1.27E-03	2.08E+01	2.07E+01	3.77E-01
IPOP-CMAES	**0.00E+00**	**0.00E+00**	0.00E+00	**2.00E+01**	**2.01E+01**	2.79E-01
RC-EA	1.06E-02	1.41E-01	2.83E-01	2.09E+01	2.09E+01	6.82E-02
FEA	1.70E+00	3.59E+00	4.43E+00	2.04E+01	2.03E+01	1.13E-01
Hy-RCGA	**0.00E+00**	**0.00E+00**	0.00E+00	2.10E+01	2.09E+01	5.75E-02
G-DE	1.63E+03	1.63E+03	1.47E+02	2.10E+01	2.10E+01	6.30E-02
EDA	2.63E+01	2.66E+01	3.92E+00	2.10E+01	2.09E+01	5.06E-02
SRPSO	4.45E-03	6.26E-03	4.52E-03	2.09E+01	2.09E+01	3.95E-02
DMeSR-PSO	1.77E-04	2.51E-03	4.92E-03	2.09E+01	2.09E+01	3.86E-02
	$\mathbf{F_9}$			$\mathbf{F_{10}}$		
	Median	Mean	STD.	Median	Mean	STD.
DE	1.81E+01	1.85E+01	5.20E+00	4.88E+01	9.69E+01	8.23E+01
RCMA	**9.95E-01**	**6.81E-01**	1.21E-01	9.51E+01	9.06E+01	4.89E+00
IPOP-CMAES	**9.95E-01**	9.38E-01	1.18E+00	**1.00E+00**	**1.65E+00**	1.35E+00
RC-EA	1.34E+02	1.31E+02	2.39E+01	2.35E+02	2.32E+02	3.51E+01
FEA	1.04E+01	2.41E+01	3.31E+01	1.75E+02	1.86E+02	5.35E+01
Hy-RCGA	1.52E+01	1.51E+01	5.04E+00	3.28E+01	3.52E+01	1.03E+01
G-DE	5.88E+01	5.79E+01	1.70E+01	7.46E+01	7.36E+01	2.14E+01
EDA	2.30E+02	2.30E+02	9.44E+00	2.41E+02	2.39E+02	1.07E+01
SRPSO	2.18E+01	3.00E+01	5.52E+00	4.29E+01	4.51E+01	6.81E+00
DMeSR-PSO	1.96E+01	2.14E+01	1.24E+01	2.72E+01	3.22E+01	1.51E+01
	$\mathbf{F_{11}}$			$\mathbf{F_{12}}$		
	Median	Mean	STD.	Median	Mean	STD.
DE	3.95E+01	3.42E+01	1.03E+01	1.78E+03	2.75E+03	3.22E+03
RCMA	3.39E+01	3.11E+01	1.52E+00	2.16E+03	4.39E+03	1.40E+03
IPOP-CMAES	**6.18E+00**	**5.48E+00**	3.13E+00	**4.54E+01**	4.43E+04	2.19E+05
RC-EA	3.77E+01	3.77E+01	1.53E+00	8.53E+04	1.01E+05	7.22E+04
FEA	3.23E+01	3.14E+01	3.15E+00	1.26E+04	1.48E+04	1.30E+04
Hy-RCGA	2.41E+01	2.47E+01	3.53E+00	8.07E+03	9.52E+03	8.07E+03
G-DE	2.10E+01	2.11E+01	5.05E+00	1.58E+03	2.44E+03	2.61E+03
EDA	3.99E+01	3.98E+01	9.74E-01	4.19E+05	4.48E+05	2.07E+05
SRPSO	1.18E+01	1.12E+01	1.00E+00	2.15E+03	2.45E+03	1.35E+03
DMeSR-PSO	7.77E+00	7.59E+00	7.85E-01	2.06E+03	**2.29E+03**	1.01E+03

Table 4.7: Performance Comparison among selected Evolutionary Algorithms on F_{13}-F_{18} CEC2005 Benchmark Functions

Algorithm	$\mathbf{F_{13}}$			$\mathbf{F_{14}}$		
	Median	Mean	STD.	Median	Mean	STD.
DE	3.18E+00	3.23E+00	8.23E-01	1.34E+01	1.34E+01	1.41E-01
RCMA	2.54E+00	3.96E+00	5.38E-01	1.27E+01	1.26E+01	7.81E-02
IPOP-CMAES	2.61E+00	2.49E+00	5.13E-01	1.29E+01	1.29E+01	4.19E-01
RC-EA	8.89E+00	9.02E+00	2.28E+00	1.33E+01	1.32E+01	2.92E-01
FEA	4.65E+00	5.57E+00	2.80E+00	1.29E+01	1.29E+01	3.92E-01
Hy-RCGA	3.18E+00	5.15E+00	4.02E+00	1.23E+01	1.21E+01	6.69E-01
G-DE	3.97E+00	4.18E+00	1.34E+00	1.31E+01	1.31E+01	3.93E-01
EDA	6.86E+01	7.36E+01	2.36E+01	1.36E+01	1.36E+01	1.26E-01
SRPSO	1.53E+00	2.27E+00	9.31E-01	1.19E+01	1.12E+01	2.55E+00
DMeSR-PSO	**1.23E+00**	**1.47E+00**	4.22E-01	**1.10E+01**	**1.11E+01**	2.12E-01
	$\mathbf{F_{15}}$			$\mathbf{F_{16}}$		
	Median	Mean	STD.	Median	Mean	STD.
DE	4.00E+02	3.60E+02	1.08E+02	2.28E+02	2.12E+02	1.10E+02
RCMA	3.00E+02	3.56E+02	1.51E+01	4.00E+02	3.26E+02	3.00E+01
IPOP-CMAES	2.00E+02	2.08E+02	2.75E+01	**3.04E+01**	**3.50E+01**	2.04E+01
RC-EA	4.14E+02	4.11E+02	4.37E+01	3.88E+02	3.81E+02	4.54E+01
FEA	4.00E+02	3.41E+02	9.49E+01	4.00E+02	2.70E+02	1.26E+02
Hy-RCGA	3.00E+02	3.04E+02	7.35E+01	5.87E+01	8.87E+01	9.69E+01
G-DE	2.31E+02	2.58E+02	6.71E+01	1.31E+02	1.91E+02	1.22E+02
EDA	4.59E+02	4.81E+02	4.67E+01	2.58E+02	2.56E+02	1.27E+01
SRPSO	3.33E+02	3.05E+02	9.52E+01	1.03E+02	1.37E+02	1.06E+01
DMeSR-PSO	**1.85E+02**	**1.88E+02**	1.24E+01	5.91E+01	7.21E+01	1.65E+01
	$\mathbf{F_{17}}$			$\mathbf{F_{18}}$		
	Median	Mean	STD.	Median	Mean	STD.
DE	2.53E+02	2.37E+02	1.22E+02	9.04E+02	9.04E+02	3.13E-01
RCMA	1.96E+02	2.79E+02	2.76E+01	8.81E+02	8.78E+02	3.23E+00
IPOP-CMAES	2.13E+02	2.91E+02	1.93E+02	9.04E+02	9.04E+02	2.88E-01
RC-EA	4.65E+02	4.54E+02	5.65E+01	1.06E+03	1.06E+03	3.02E+01
FEA	2.85E+02	3.15E+02	1.25E+02	9.58E+02	9.64E+02	1.89E+01
Hy-RCGA	**8.34E+01**	1.35E+02	1.12E+02	9.04E+02	9.04E+02	5.95E-01
G-DE	1.27E+02	1.45E+02	7.02E+01	8.63E+02	8.62E+02	6.06E+00
EDA	2.88E+02	2.88E+02	1.05E+01	9.27E+02	9.28E+02	4.23E+00
SRPSO	1.51E+02	1.80E+02	2.94E+01	8.54E+02	8.45E+02	3.29E+01
DMeSR-PSO	1.23E+02	**1.22E+02**	2.12E+01	**8.29E+02**	**8.29E+02**	3.77E-01

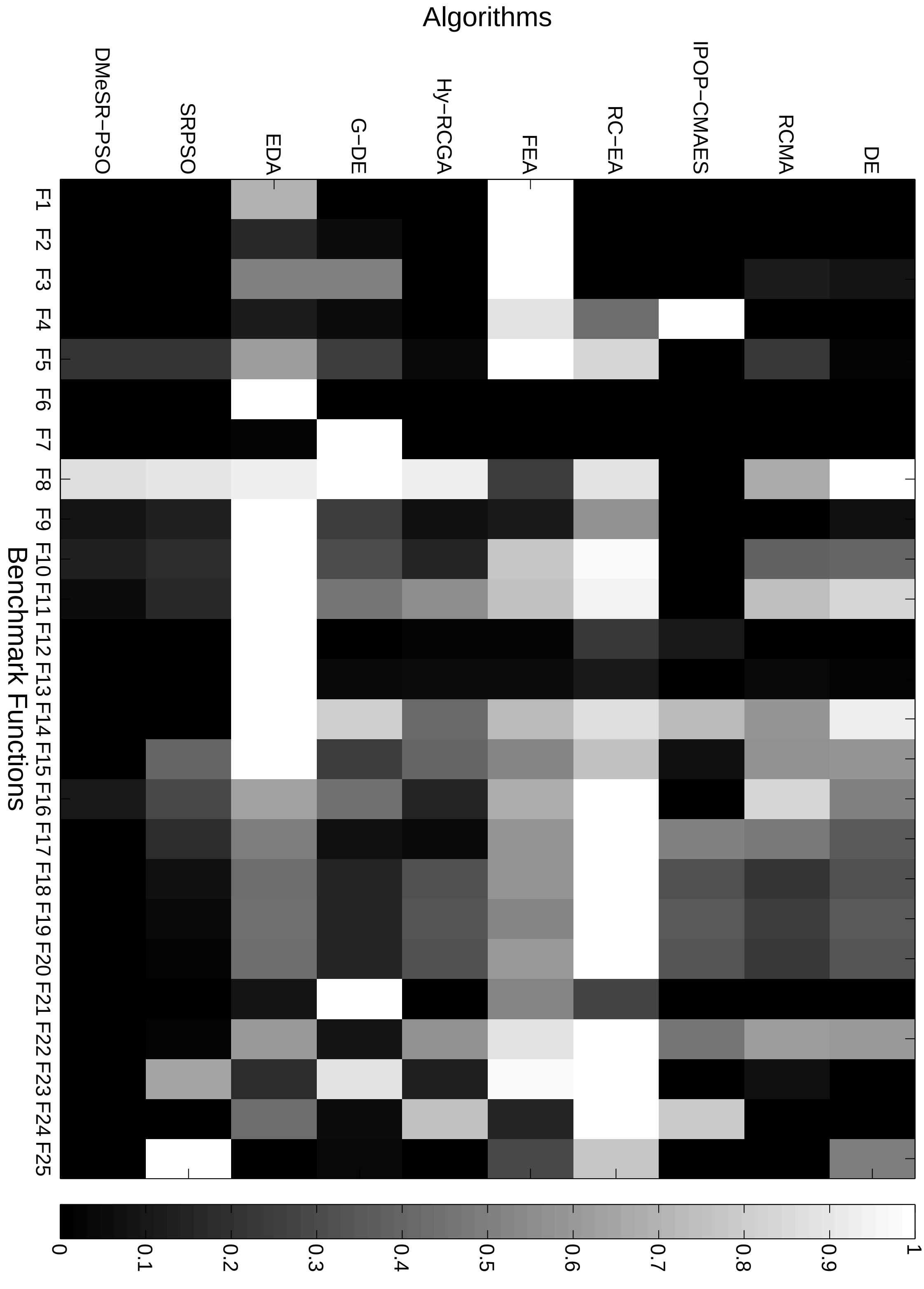

Figure 4.7: Heat Map for the Mean Performance of the Selected Algorithms on all the Benchmark Functions

Table 4.8: Performance Comparison among selected Evolutionary Algorithms on F_{19}-F_{22} CEC2005 Benchmark Functions

Algorithm	**F_{19}**			**F_{20}**		
	Median	Mean	STD.	Median	Mean	STD.
DE	9.04E+02	9.04E+02	6.25E-01	9.04E+02	9.04E+02	6.15E-01
RCMA	8.80E+02	8.80E+02	7.40E-01	8.80E+02	8.79E+02	8.48E-01
IPOP-CMAES	9.04E+02	9.04E+02	2.71E-01	9.04E+02	9.04E+02	2.48E-01
RC-EA	1.04E+03	1.05E+03	2.98E+01	1.05E+03	1.06E+03	4.33E+01
FEA	9.59E+02	9.43E+02	6.55E+01	9.72E+02	9.67E+02	4.21E+01
Hy-RCGA	9.03E+02	9.04E+02	5.13E-01	9.03E+02	9.03E+02	2.44E-01
G-DE	8.63E+02	8.62E+02	4.01E+00	8.64E+02	8.63E+02	3.60E+00
EDA	9.26E+02	9.26E+02	3.52E+00	9.27E+02	9.27E+02	4.44E+00
SRPSO	8.41E+02	8.38E+02	1.71E+01	8.41E+02	8.33E+02	3.29E+01
DMeSR-PSO	**8.29E+02**	**8.28E+02**	1.94E+00	**8.29E+02**	**8.28E+02**	2.64E+00
	F_{21}			**F_{22}**		
	Median	Mean	STD.	Median	Mean	STD.
DE	**5.00E+02**	**5.00E+02**	0.000E+00	8.97E+02	8.97E+02	1.33E+01
RCMA	**5.00E+02**	**5.00E+02**	0.00E+00	9.11E+02	9.08E+02	1.69E+00
IPOP-CMAES	**5.00E+02**	**5.00E+02**	0.00E+00	8.00E+02	8.03E+02	1.86E+01
RC-EA	5.01E+02	6.04E+02	2.18E+02	1.16E+03	1.16E+03	2.94E+01
FEA	**5.00E+02**	6.95E+02	3.20E+02	1.08E+03	1.08E+03	4.31E+01
Hy-RCGA	**5.00E+02**	**5.00E+02**	0.00E+00	8.77E+02	8.74E+02	1.82E+01
G-DE	8.67E+02	8.69E+02	4.48E+00	5.61E+02	5.61E+02	4.62E+00
EDA	5.06E+02	5.31E+02	1.19E+02	8.97E+02	8.97E+02	7.15E+00
SRPSO	**5.00E+02**	**5.00E+02**	0.00E+00	5.19E+02	5.22E+02	1.10E+01
DMeSR-PSO	**5.00E+02**	**5.00E+02**	0.00E+00	**5.00E+02**	**5.09E+02**	7.25E+01

the algorithms, the mean performances are normalized on a 0 to 1 scale and plotted using a heat map. The heat map of normalized mean performances of all the selected algorithms is presented in figure 4.7. In the figure, the benchmark functions are represented by the x-axis and the algorithms are represented on the y-axis. In the figure, the performances are plotted on a gray scale map where black colour represents the best performance whereas the white colour represents the worst performance. From the heatmap, one can easily identify the poorly performing algorithms from the white patches and the better performing algorithms from the black patches. Therefore, the peformance of RC-EA, FEA and EDA on the set of benchmark functions can be termed as poor compared to

Table 4.9: Performance Comparison among selected Evolutionary Algorithms on F_{23}-F_{25} CEC2005 Benchmark Functions

Algorithm	$\mathbf{F_{23}}$			$\mathbf{F_{24}}$			$\mathbf{F_{25}}$		
	Median	Mean	STD.	Median	Mean	STD.	Median	Mean	STD.
DE	5.34E+02	**5.34E+02**	4.26E-04	**2.00E+02**	**2.00E+02**	0.00E+00	7.30E+02	7.30E+02	3.74E-01
RCMA	5.34E+02	5.59E+02	2.44E+01	**2.00E+02**	**2.00E+02**	0.00E+00	2.11E+02	**2.11E+02**	4.48E-02
IPOP-CMAES	5.34E+02	**5.34E+02**	2.22E-04	9.39E+02	9.10E+02	1.48E+02	2.10E+02	**2.11E+02**	9.21E-01
RC-EA	8.58E+02	9.22E+02	1.81E+02	1.12E+03	1.10E+03	1.21E+02	1.07E+03	1.03E+03	1.92E+02
FEA	1.18E+03	9.10E+02	3.32E+02	**2.00E+02**	3.28E+02	3.54E+02	**2.00E+02**	5.17E+02	5.04E+02
Hy-RCGA	5.70E+02	5.87E+02	7.70E+01	9.68E+02	8.77E+02	2.55E+02	2.11E+02	**2.11E+02**	2.15E-01
G-DE	8.75E+02	8.75E+02	3.55E+00	2.50E+02	2.50E+02	3.22E+00	2.53E+02	2.53E+02	1.36E+00
EDA	5.59E+02	6.02E+02	1.51E+02	2.17E+02	5.79E+02	3.89E+02	2.13E+02	2.13E+02	9.33E-01
SRPSO	7.34E+02	7.88E+02	9.03E+01	2.08E+02	2.10E+02	4.18E+00	1.27E+03	1.26E+03	6.40E+00
DMeSR-PSO	**5.25E+02**	**5.34E+02**	9.00E+00	**2.00E+02**	**2.00E+02**	0.00E+00	**2.00E+02**	**2.11E+02**	2.93E+01

other algorithms. Similarly, the performance of IPOP-CMAES, SRPSO and DMeSR-PSO can be termed as significant over others. One can also identify the most robust algorithm from the heat map using the steady dark colour on the entire tested functions. It can be seen that for DMeSR-PSO there is steady dark colour with an exception of function F_8 and this suggests that the performance is consistent over the entire benchmark functions. Hence, one can conclude that DMeSR-PSO is a better performing algorithm on the set of benchmark functions.

4.4.5 A statistical comparison

In order to further study the significance of the compared algorithms over each other, a statistical test has been conducted using the non-parametric Friedman test on the average rank followed by the pairwise post-hoc Bonferroni-Dunn test [128]. The average ranks of the median and mean performances of all the algorithms are presented in table 4.10. The table also contain the statistical test values of Friedman test and the Benforroni-Dunn test. From the table, one can easily see that the average rank of DMeSR-PSO is lowest in both the median and mean performances indicating it as the best performing algorithm. According to average ranks, the performance of the selected algorithms can be ranked as follows:

- **Median Perforamnce:** DMeSR-PSO followed by IPOP-CMAES, SRPSO, HyRCGA, RCMA, DE, G-DE, FEA, RC-EA and EDA respectively.
- **Mean Perforamnce:** DMeSR-PSO followed by IPOP-CMAES, SRPSO, HyRCGA, RCMA, DE, G-DE, FEA, EDA and RC-EA respectively.

Further, the statistical test results are also included in the same table 4.10. The computed F-scores of the Friedman test are 22.4735 and 26.8261 for median and mean performances respectively. In this study, there are 10 algorithms and 25 functions for which the Friedman F-critical value obtained from the F-table is 1.9234. Since, the F-critical value is

Table 4.10: Average Ranks and Statistics for $\alpha = 0.05$ significance level

Algorithms	Average Rank on Median Values	Average Rank on Mean Values
DE	5.38	5.34
RCMA	4.94	4.90
IPOP-CMAES	3.20	3.70
RC-EA	8.36	8.68
FEA	7.38	7.94
Hy-RCGA	4.36	4.14
G-DE	5.98	5.90
EDA	8.56	8.22
SRPSO	4.26	3.94
DMeSR-PSO	**2.58**	**2.24**
F-score (Friedman)	22.4735	26.8261
Critical Difference (Benforroni-Dunn)	1.2128	1.1648

comparatively much smaller than the computed F-scores in both the cases therefore; the null hypothesis can be easily rejected. Hence, it may be concluded that the performance of algorithms selected for comparison on the CEC2005 benchmark functions are significantly different from each other.

Next, the significance of the performance of DMeSR-PSO over other algorithms can be accessed using the Benforroni-Dunn test. The minimum required difference in the average ranks to signify an algorithm over others is 1.2128 for median values and 1.1648 for mean values. The average rank difference between DMeSR-PSO and IPOP-CMAES in median performance is 0.62 which suggest that the median performances of DMeSR-PSO and IPOP-CMAES are similar to each other. Further, the average rank difference between DMeSR-PSO and the third ranked algorithm SRPSO is 1.68 which is greater than 1.2128. Hence, for the median performance it can be concluded that DMeSR-PSO has significantly provided better results compared to all the other selected evolutionary algorithms with 95% confidence level and its performance is similar to the IPOP-CMAES algorithm. The average rank difference between DMeSR-PSO and IPOP-CMAES in mean performance is 1.46 which is greater than the critical value of 1.1648. Hence, it can be concluded that the mean performance of DMeSR-PSO is significantly better than all the other selected algorithms with a confidence level of 95%.

In this chapter, a human social intelligence inspired PSO algorithm incorporating the mentoring and self-regulating schemes has been presented. The algorithm is referred to as Dynamic Mentoring and Self-Regulation based Particle Swarm Optimization (DMeSR-PSO) algorithm. In DMeSR-PSO, the particles are divided into three different groups of mentors, mentees and independent learners where each group performed search using different learning strategy. The particles utilized the following strategies:

- The dynamic mentoring scheme,
- Self-Regulating inertia weight scheme and
- Self-perception based selection of direction from the global best position.

Using the schemes, the algorithm utilizes the human self-learning and socially shared information processing strategies simultaneously and it has successfully:

- Improved the convergence characteristics of the SRPSO algorithm.
- Provided faster convergence closer to the optimum solution with robustness on diverse CEC2005 benchmark problems.
- Statistically outperformed the widely accepted state-of-the-art search based optimization algorithms.

Hence, it may be concluded that the idea of human socially shared information processing strategy applied to the PSO algorithm has successfully enhanced its convergence characteristics. Further, from the experimental results one can see that the performance of DMeSR-PSO is better than others but still it has converged to a point which is not the optimum solution. This suggests that the mentoring scheme where sometimes the mentees are allowed to learn themselves could have prevented the algorithm to move towards the optimum solution. A more stiff scheme, where the lesser performing particles are fully guided by elite particles can help in achieving improved performance.

In the next chapter, a socially guided interaction among the particles has been implemented in the SRPSO algorithm to enhance its convergence.

Chapter 5

Directionally Driven Self-Regulating Particle Swarm Optimization Algorithm

In chapters 3 and 4, two new PSO variants have been proposed inspired from human self-learning principles and human socially shared information processing strategy respectively. The SRPSO algorithm introduced in chapter 3 employed self-regulating inertia weight for the best particle and self-perception based strategy for others. Furthermore, the dynamic mentoring scheme together with the self-regulation scheme was introduced in chapter 4 as the DMeSR-PSO algorithm. Both the algorithms have exhibited significant performance improvement with accuracy and robustness. It has been observed that the performance has suffered due to lack of guidance in the proposed strategies. Further, recent studies have indicated that the nature of real-world optimization problems have become extremely complex with the association of increasing dimensionality, differentiability, multi-modality and rotation characteristics [150]. In this chapter, a new algorithm has been proposed with a socially guided interactive learning scheme together with a rotational invariant scheme to address the complexities associated with optimization problems.

5.1 Overview

The increasing complexity in the nature of real-world problem has raised several new difficulties. One of these major difficulties is the rotation of search space which requires a rotationally invariant optimization algorithm for providing accurate solutions. If any problem contains the rotation characteristics then its dimensions become non-separable which makes it extremely difficult for any rotationally variant optimization algorithm to locate the true optimum solution. Therefore, any algorithm performing the dimension wise optimization of such a problem will not be able to locate the true optimum solution. It has been theoretically proven in [151] that the PSO algorithm performs the optimization separately along each dimension as the velocity update is carried out along dimension by dimension. This clarify that PSO is a rotationally variant optimization algorithm and its performance is affected by the rotation of the search space. Similarly, SRPSO and DMeSR-PSO proposed in this thesis are also rotationally variant algorithms. In this chapter, a rotationally invariant scheme has been proposed to tackle with rotated optimization problems.

It has also been observed in the previous chapter that in most of the cases the obtained results are far away from the true optimum value. The mentoring learning scheme allows the mentees to perform self-learning and therefore the mentee particles in DMeSR-PSO were also allowed to perform self-learning during the search. This might have prevented the least performing particles from acquiring a better learning strategy as they do not get the required support from the better performing particles to get a directional update towards potentially better search areas and eventually contribute towards convergence. Therefore an immutable guidance scheme has been proposed for the lesser performing particles to strict them towards learning from the elite particles. In real-life, it has been observed that when people learn in group, the least efficient among them are always connected with efficient learners from whom they take guidance for performance improvement. For example, during the learning phase of driving a car one always need to take driving guidance from an expert. The expert is always closely connected with the learner and guide him the techniques of driving. Once the person learns the driving techniques, then the expert allow him to drive. Similarly, such a guidance is required by the lesser performing particles for their performance improvement.

Incorporating these ideas, a new rotationally invariant and socially guided PSO has been proposed in this chapter referred to as Directionally Driven SRPSO (DD-SRPSO) algorithm. In the next section, the detailed description of DD-SRPSO algorithm is presented.

5.2 Directionally Driven Self-Regulating Particle Swarm Optimization Algorithm

The DD-SRPSO algorithm addresses the issue regarding the rotation variance of the search space (the optimum solution cannot be located by optimizing the objective function along each dimension separately). The importance of socially shared information in a learning environment is used here to develop a directional update strategy for the poorly performing particles. In DD-SRPSO algorithm, two new learning strategies have been added to the basic SRPSO algorithm:

- A Directional Update strategy and
- A Rotational Invariant strategy.

With the inclusion of two new strategies, the DD-SRPSO algorithm combines 4 different learning strategies for the particles. Next, the new proposed strategies are described in detail.

5.2.1 Directional update strategy

The directional update strategy is inspired by the human social interactions that occur in the real world. In human learning psychology, a less efficient learner always seeks guidance from efficient learners for acquiring a proper awareness of the learning environment and to eventually enhance his learning abilities [1]. These poor learners are not only dependent on the best learner but they also seek guidance from other top learners. This motivated us to select a social interactive mechanism for poorly performing particles of the SRPSO algorithm. In the directional update strategy, a group of elite particles from the top performing particles are selected to update the search directions of the group of poorly performing particles. Both the groups have been selected as a fraction of the entire swarm size represented by ε. The value of ε has been selected after several experimental evaluations as 0.05 (5% of the total population). In other words, if the swarm size is 80, top 4 and bottom 4 particles will be selected for the group of elite particles and poorly performing particles respectively. The guidelines for the selection of ε is provided later in the results section. The selection process is described in Figure 5.1. In the figure, it is shown that the total particles 'n' are first sorted according to their performances. Then the top 5% particles are selected in the group of elite particles and the bottom 5% particles are selected in the group of poorly performing particles.

All the poorly performing particles will have a full social-cognition (they will always follow the global search directions). In addition, any three particles from the elite group will be randomly selected and the median of the personal best of these three particles will be used as the personal best for the poorly performing particle. Note that if the elite

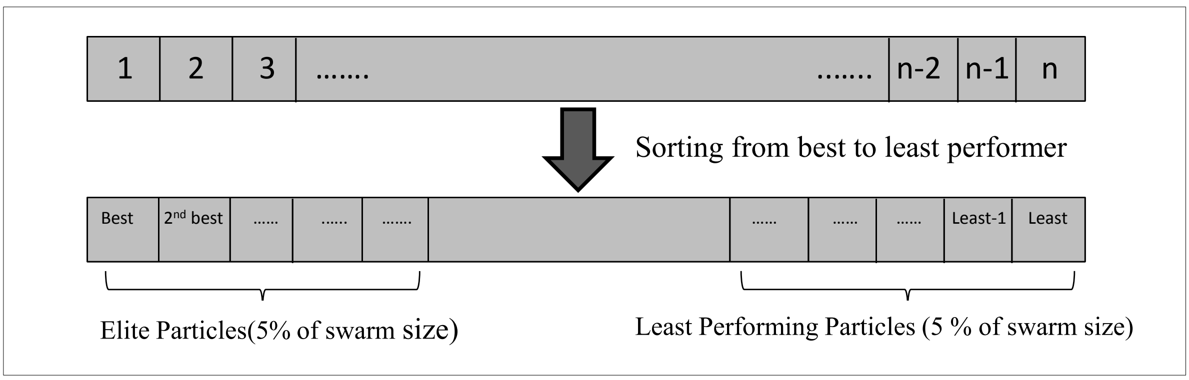

Figure 5.1: Division of particles in Elite and Poorly Performing particles Groups

group contains smaller than three particles then the top three performing particles will be selected. This has been done to prevent the particle from biasing towards a single particle that might lead to a position further away from the global optimum. The selection of the personal best for poorly performing particles using the median of three particles in shown in Figure 5.2. In the figure, Particle A,B and C are the randomly selected three particles from the elite group and the median is the centroid of the personal best positions of the three particles.

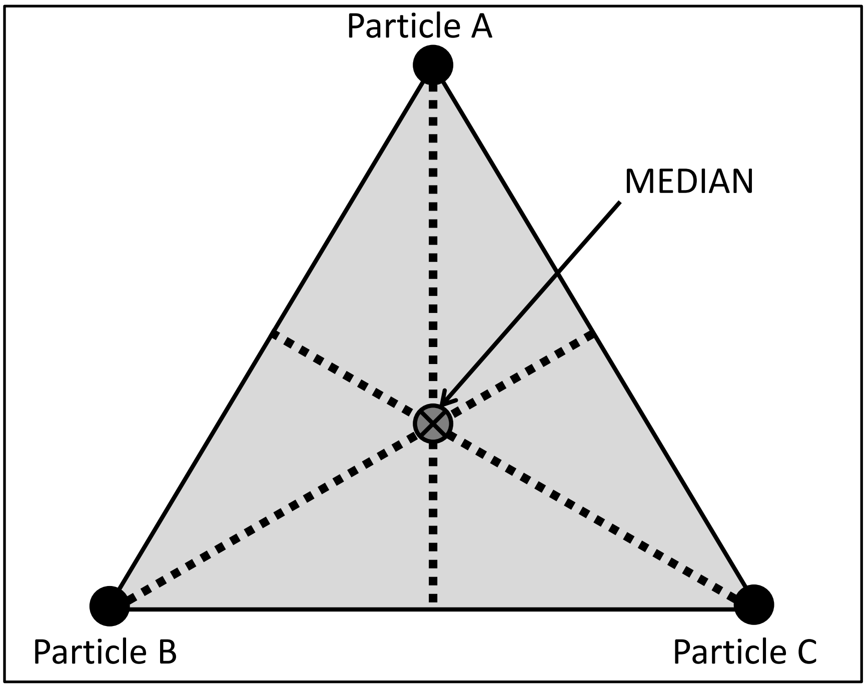

Figure 5.2: Selection of Personal Best for Poorly performing particles

The velocity update equation for the poorly performing particles is given by:

$$V_{kd}^{t+1} = \omega_k V_{kd}^t + c_1 r_1 (lP_{kd}^t - X_{kd}^t) + c_2 r_2 (P_{gd}^t - X_{kd}^t) \tag{5.1}$$

where lP_{kd}^t is the median of personal best of three particles from the elite group and 'k' is the particle from the poorly performing particles group. The rationale of this strategy is to improve the exploitation ability of the poorly performing particles by following the guidance of the top performing particles. Hence, the poorly performing particles are able to move closer to the optimum value that is defined by the group of top performing particles. The group fraction $\varepsilon = 0.05$ has been selected because the size of the group will affect the stability of exploitation process. The bigger size of the group leads to instability in exploitation as more particles will get involved as guides for the poorly performing particles.

5.2.2 Rotationally invariant strategy

The rotationally invariant strategy to enable the particles to tackle problems with the rotational characteristics of the search space has been recently proposed in SPSO-2011 [152]. In SPSO-2011, the velocity update process has been carried out by exploiting the idea of rotational invariance which has significantly enhanced the algorithm's performance. The rotational invariance strategy is defined by a center of gravity denoted by $\overrightarrow{G_i}$ for the i_{th} particle calculated around three points; $\overrightarrow{X_i}$ (current position of the particle), a point slightly beyond $\overrightarrow{P_i}$ (personal best position of the particle) and a point slightly beyond $\overrightarrow{L_i}$ (Local best position). This $\overrightarrow{G_i}$ is then used to generate a hyper-sphere and a random point is selected in the hyper-sphere as the new current position of the particle to incorporate the rotationally invariant characteristics in the particles. Improved performance has been observed by SPSO-2011 on rotated multimodal functions, but this algorithm has shown weakest performance for all the composition problems. The calculation of $\overrightarrow{G_i}$ has been performed using the local neighbourhood information which has kept the particles unaware of the global best location.

In DD-SRPSO, the calculation of $\overrightarrow{G_i}$ is performed using the global best location. Therefore, for a given particle i, the center of gravity $\overrightarrow{G_i}$ is defined by three points: $\overrightarrow{X_i}$ (current position of the particle), a point slightly beyond $\overrightarrow{P_i}$ (personal best position of the particle) represented as $\overrightarrow{p_i}$ and a point slightly beyond $\overrightarrow{P_g}$ (Global best position best position) represented as $\overrightarrow{p_g}$. Further, the gravity calculation is dependent on the self-perception of particles for global search directions $\overrightarrow{p_i^{sb}}$. If $\overrightarrow{p_i^{sb}} = 1$, the gravity will be

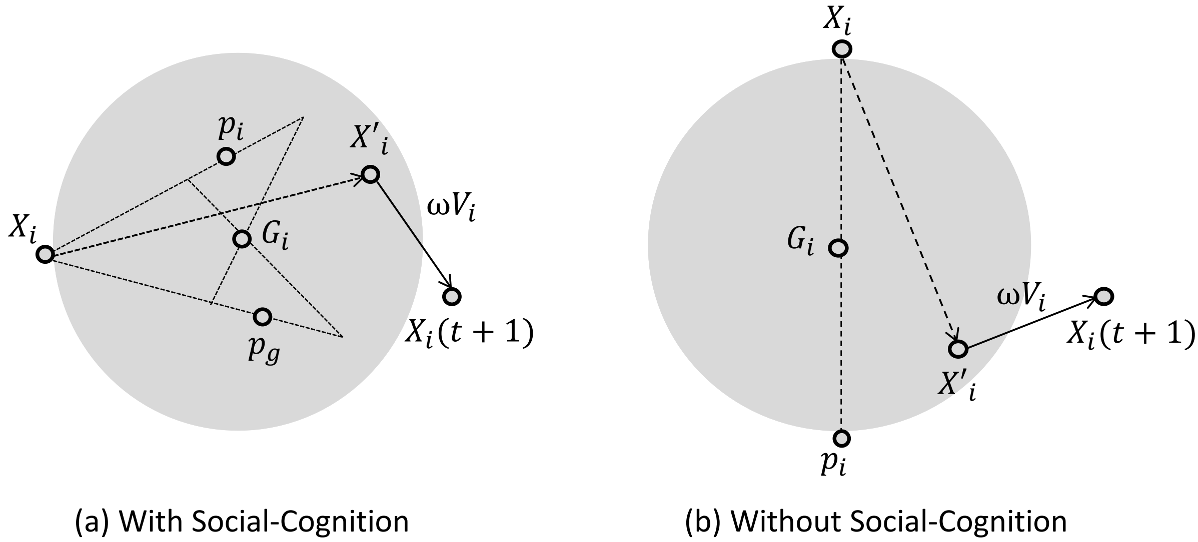

Figure 5.3: Center of Gravity calculation for the Hypersphere

calculated using the global best value otherwise the gravity will only be calculated using only $\overrightarrow{X_i}$ and $\overrightarrow{p_i}$. Both these scenarios are clearly shown in Figure 5.3. Figure 5.3(a) describes the scenario where $\overrightarrow{p_i^{so}} = 1$ and the particles perform search following social cognition. Figure 5.3(b) describes the scenario where $\overrightarrow{p_i^{so}} = 0$ and the particle performs a search without social cognition.

Mathematically, the calculation of G_{id} is represented by the following set of equations:

$$\overrightarrow{p_i^t} = \overrightarrow{X_i^t} + c_1 r_1 \otimes (\overrightarrow{P_i^t} - \overrightarrow{X_i^t}) \tag{5.2}$$

$$\overrightarrow{p_g^t} = \overrightarrow{X_i^t} + c_2 r_2 \overrightarrow{p_i^{so}} (\overrightarrow{P_g^t} - \overrightarrow{X_i^t}) \tag{5.3}$$

$$\overrightarrow{G_i^t} = \begin{cases} \frac{\overrightarrow{p_i^t} + \overrightarrow{p_g^t} + \overrightarrow{X_i^t}}{3}, & \text{if } \overrightarrow{p_i^{so}} = 1 \\ \frac{\overrightarrow{p_i^t} + \overrightarrow{X_i^t}}{2}, & \text{if } \overrightarrow{p_i^{so}} = 0 \end{cases} \tag{5.4}$$

This $\overrightarrow{G_{i^t}}$ is then used to randomly select a point $\overrightarrow{X_i'^t}$ in the hypersphere $H_i(\overrightarrow{G_{i^t}}, \|\overrightarrow{G_i^t} - \overrightarrow{X_i^t}\|)$. The velocity update equation for providing the particles rotationally invariant characteristics is given by:

$$\overrightarrow{V_i^{t+1}} = \omega \overrightarrow{V_i^t} + H_i(\overrightarrow{G_i^t}, \|\overrightarrow{G_i^t} - \overrightarrow{X_i^t}\|) - \overrightarrow{X_i^t} \tag{5.5}$$

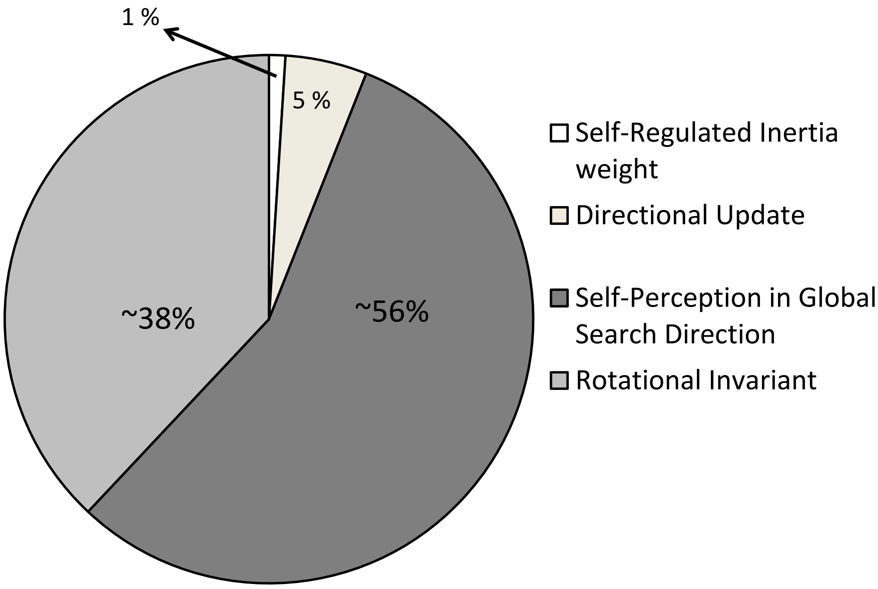

Figure 5.4: Percentage of Particles in each strategy

The position update equation of all the particles is the same as that of SRPSO defined by equation 4.6.

To summarize, the DD-SRPSO algorithm is an improved variant of the SRPSO algorithm with two newly proposed learning strategies, viz., the directional update strategy and the rotational invariant strategy in addition to the self-regulating and self-perception strategies of the basic SRPSO algorithm. Next, the selection of particles to perform the search using either of the above strategies is presented.

5.2.3 Selection of particles for each strategy

In the DD-SRPSO algorithm, the best particle employs a self-regulated inertia weight strategy, whereas the poorly performing particles employ a directional update strategy. The assigning of the remaining two strategies to all the other particles is performed using a uniformly distributed random number within the range of [0, 1],$\delta \sim U([0,1])$. All the remaining particles are assigned a random number and a threshold value β has been defined to divide the particles into two groups for adopting either the self-perception in global search direction strategy or the rotational invariant strategy. Some experiments have been carried out to determine a suitable value for β and is presented in the section 5.3.2. The best possible value selected for β is 0.6 and according to this value all the particles will perform a search using rotational invariant strategy if $\delta > \beta$ otherwise the search will be performed using self-perception in the global search direction strategy. The approximate percentage of particles in each strategy, according to the above defined criteria for a swarm size of 100 particles is summarized in Figure 5.4.

5.2.4 Analysis and impact of the proposed learning strategies

The DD-SRPSO algorithm is equipped with different learning strategies for different particles and collectively the learning strategies are contributing towards its improved performance. To analyse the working of DD-SRPSO and show the benefits of the proposed strategies in detail, a unimodal rotated high conditional elliptic problem, as defined in [153], has been selected for illustration here. The problem is a smooth, non-separable, ill-conditioned and rotated function defined as:

$$f_2(x) = \sum_{i=1}^{D} (10^6)^{\frac{i-1}{D-1}} z_i^2 + f_2^* \tag{5.6}$$

where D is the dimension, f_2^* represents the shifted global optimum. Further details about the problem are available in [153].

The experiments have been conducted on a 50 dimensional problem with a swarm size of 50 particles and the function evaluations have been set to 5000×D. A set of seven different experiments was conducted 30 times to evaluate the working of each of the individual strategies and the mean performance on each of the experiments is shown in Figure 5.5. From the figure, it can be seen that the performance of SRPSO is much better than that of the basic PSO algorithm. Further, with the addition of the rotational invariant strategy, both the algorithms observe marginal improvement in their performances. In addition, when the directional update strategy is applied to PSO

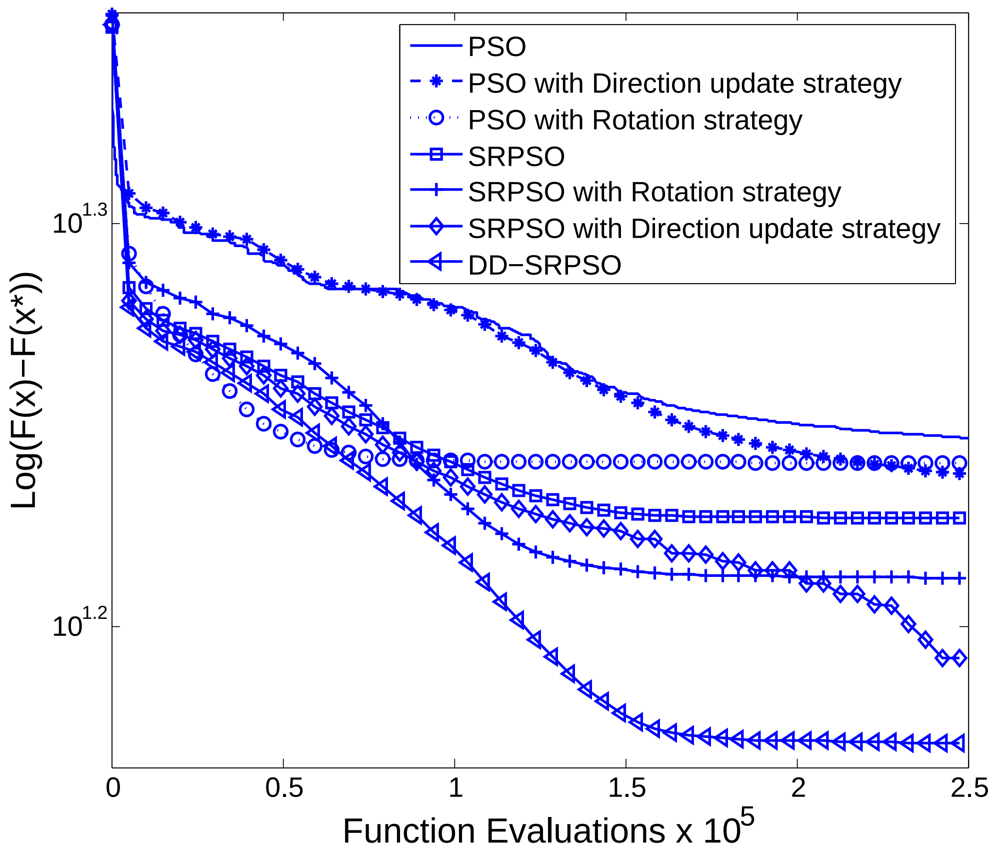

Figure 5.5: Impact of Learning Strategies

and SRPSO, both experience better improvement in their convergence. Finally, when both the rotational invariant and directional update strategies are applied to the SRPSO algorithm, the performance is significantly enhanced as represented by the convergence plot of the DD-SRPSO algorithm.

Incorporating all the above strategies, the pseudo-code for DD-SRPSO is summarized below in Algorithm 3.

5.3 Experimental Setup, Performance Evaluation and Discussion

This section presents the performance analysis of DD-SRPSO compared to well-known PSO variants and other evolutionary algorithms. First, in this section, the selected

```
Initialization:
for the particles (i) do
    Initialize the position X_i randomly in the search range (X_min, X_max)
    Randomly initialize velocity V_i
end
Using the fitness function, calculate the fitness values;
From X_i, determine the personal best position;
The DD-SRPSO Loop:
while (success= 0 and t ≤ max_iterations) do
    Find the best fitness value from the entire swarm;
    Let the position be the global best;
    Sort all particles w.r.t the fitness values;
    Group the top 5% particles as the elite particles;
    Group the bottom 5% particles as the poorly performing particles;
    for the best particle do
        Calculate the inertia weight ω using equation (3.4);
        Update the velocity using equation (3.6);
    end
    for the poorly performing particles do
        Randomly select 3 particles from the group of elite particles;
        for j = 1 : Dimension do
            Calculate the median of the personal best of the 3 particles;
            Update the velocity using equation (5.1);
        end
    end
    for the remaining particles do
        Generate the random number δ and set β = 0.6;
        for j = 1 : Dimension do
            Generate a using uniform random distribution;
            if (a > 0.5), then
                The dimension from global best is selected
            else
                the dimension is rejected
            end
            if (δ > β), then
                Calculate G_id using equation (5.4)
                Update the velocity using equation (5.5)
            else
                Update the velocity using equation (3.7)
            end
        end
    end
    Update the position of each particle using equation (4.6);
end
```

Algorithm 3: The DD-SRPSO Algorithm

benchmark functions are discussed followed by analytical study of the proposed strategies. Next, the experimental evaluation is provided for comparative analysis. Finally, the significance of performance has been proved.

5.3.1 Benchmark functions

As stated in the introduction, the nature of complex real world problems is associated with rotated characteristics together with increased dimensionality and multi-modality. Therefore, the choice of benchmark problems for DD-SRPSO performance evaluation should address these problems. One of the recently proposed benchmark function, the CEC2013 [153] problem sets are specifically designed to address the recent issues of the optimization problems. The same CEC2013 [153] benchmark functions have been used in this study for experimental evaluation.

The selected CEC2013 benchmark functions [153] are the collection of twenty-eight complex, rotated and shifted problems. In this test suite, all the composition functions have been improved with the inclusion of control parameters and weights for each selected function. With the help of these parameters, the properties of each function are merged together in the composition function. The CEC 2013 benchmark functions [153] broadly fall into three major groups (based on their characteristics):

- Unimodal functions (F_1 - F_5),
- Basic Multimodal functions (F_6 - F_{20}) and
- Composition functions (F_{21} - F_{28}).

The global optimum solutions for all the test functions are shifted to $\boldsymbol{o} = [o_1, o_2, \cdots, o_D]$ and all of them are defined in the same search range of $[-100, 100]^D$. The characteristics of the functions are made more complex by introducing $\mathbf{M_1}, \mathbf{M_2}, \cdots, \mathbf{M_3}$ which are the orthogonal (rotation) matrices used to rotate the functions. Further details about the benchmark functions are given in [153].

5.3.2 Guidelines for the selection of particles for each strategy

The DD-SRPSO algorithm is performing optimization by utilizing different strategies for different group of particles. It is necessary to choose an appropriate number of particles to undergo a defined strategy and eventually provide the best solutions. In DD-SRPSO, there are two main selections involved:

- Selection of particles in the group of elite particles and poorly performing particles to undergo directional update learning scheme.
- Selection of particles to perform search either by using rotational invariant strategy or the self-perception in global search direction strategy for all the remaining particles.

In this study, the first selection is represented as a fraction over the swarm size (ε). The value of ε is equal for the elite and poorly performing particles from the entire swarm i.e. a value of 0.1 means that top 10% better performing particles are selected in the elite group and the bottom 10% performing particles are selected in the poorly performing particles' group. The second selection is represented by a constant β. Here the selection is done in such a way that for $\beta = \rho$, where ρ represents the the percentage of particles, will perform search using self-perception in the global search direction strategy. All the remaining particles will perform the search using the rotational invariant strategy.

A set of experiments were conducted using the following different values:

- ε = 0.03, 0.05 and 0.1
- β = 0.4, 0.5,0.6 and 0.7

Every possible combinations of the selected values for ε and β have been tested to find the best possible combination. The experiments have been conducted on the following benchmark functions:

- Two unimodal functions: F_2 and F_4,
- Three multimodal functions: F_6, F_{12} and F_{18} and
- One composition function: F_{25}.

The experiments were conduction on the 50D problems using a swarm size of 100 particles. . The experiments were conducted 30 times and every time the total number of function evaluations was set to 250,000. Finally, the mean and standard deviation performances from the different settings of algorithm are reported.

Table 5.2 presents the mean and standard deviation of the performances for all the different settings for the DD-SRPSO algorithm. From the table, it can be seen that two of the settings from the twelve different combinations are providing better solutions. These two settings are $\varepsilon = 0.05$ and $\beta = 0.6$ and $\varepsilon = 0.05$ and $\beta = 0.5$. Among them, the first setting is providing best solutions in 4 out of the 6 selected functions whereas the second setting is only providing the best solutions in two functions. From these observations, it may be inferred that the value of $\varepsilon = 0.05$ is the best possible group size for the directional update strategy. Furthermore, the selection of $\beta = 0.6$ is the best value for selection of particles to undergo the self-perception in global search direction strategy.

To further visualize the performance of each setting, a heat map of the normalized performances is presented in figure 5.6. The twelve different settings, starting from DD-SRPSO (ε=0.03, β=0.4) till DD-SRPSO (ε=0.1, β=0.7) are represented along the x-axis of the graph as set1 till set12 respectively. The selected benchmark functions are presented on the y-axis of the graph. From the figure, one can easily identify that set7 representing the setting ε=0.05, β=0.6 is the best performer among all other settings. It has provided consistent solutions on the set of selected benchmark functions. Therefore, the same values of ε and β are selected for all the further experimentation.

Table 5.1: Strategy analysis on selected CEC 2013 Benchmark Functions

Func.	Algorithm	Mean	STD.
F_2	DD-SRPSO (ε=0.03, β=0.4)	4.52E+06	7.54E+05
	DD-SRPSO (ε=0.03, β=0.5)	5.64E+06	9.12E+05
	DD-SRPSO(ε=0.03, β=0.6)	2.62E+06	2.07E+06
	DD-SRPSO (ε=0.03, β=0.7)	6.47E+06	3.72E+06
	DD-SRPSO (ε=0.05, β=0.4)	3.45E+06	1.49E+06
	DD-SRPSO (ε=0.05, β=0.5)	3.10E+06	1.10E+06
	DD-SRPSO (ε=0.05, β=0.6)	**2.09E+06**	6.88E+05
	DD-SRPSO (ε=0.05, β=0.7)	8.11E+06	6.10E+06
	DD-SRPSO (ε=0.1, β=0.4)	6.37E+06	8.24E+06
	DD-SRPSO (ε=0.1, β=0.5)	4.59E+06	2.95E+06
	DD-SRPSO (ε=0.1, β=0.6)	5.12E+06	5.56E+06
	DD-SRPSO (ε=0.1, β=0.7)	6.18E+06	6.44E+06
F_4	DD-SRPSO (ε=0.03, β=0.4)	1.01E+04	1.98E+03
	DD-SRPSO (ε=0.03, β=0.5)	9.03E+03	1.26E+03
	DD-SRPSO(ε=0.03, β=0.6)	8.50E+03	1.07E+03
	DD-SRPSO (ε=0.03, β=0.7)	7.78E+03	1.11E+03
	DD-SRPSO (ε=0.05, β=0.4)	9.40E+03	1.47E+03
	DD-SRPSO (ε=0.05, β=0.5)	8.86E+03	1.49E+03
	DD-SRPSO (ε=0.05, β=0.6)	**7.47E+03**	8.50E+02
	DD-SRPSO (ε=0.05, β=0.7)	8.89E+03	2.59E+03
	DD-SRPSO (ε=0.1, β=0.4)	9.57E+03	2.17E+03
	DD-SRPSO (ε=0.1, β=0.5)	8.71E+03	1.61E+03
	DD-SRPSO (ε=0.1, β=0.6)	9.51E+03	1.72E+03
	DD-SRPSO (ε=0.1, β=0.7)	9.05E+03	1.71E+03
F_6	DD-SRPSO (ε=0.03, β=0.4)	4.64E+01	8.11E-01
	DD-SRPSO (ε=0.03, β=0.5)	4.59E+01	1.10E+00
	DD-SRPSO(ε=0.03, β=0.6)	4.58E+01	6.60E-01
	DD-SRPSO (ε=0.03, β=0.7)	4.58E+01	3.81E-01
	DD-SRPSO (ε=0.05, β=0.4)	4.62E+01	1.19E+00
	DD-SRPSO (ε=0.05, β=0.5)	4.60E+01	1.06E+00
	DD-SRPSO (ε=0.05, β=0.6)	**4.41E+01**	1.22E+00
	DD-SRPSO (ε=0.05, β=0.7)	4.68E+01	1.33E+00
	DD-SRPSO (ε=0.1, β=0.4)	4.63E+01	1.62E+00
	DD-SRPSO (ε=0.1, β=0.5)	4.58E+01	9.13E-01
	DD-SRPSO (ε=0.1, β=0.6)	4.60E+01	1.53E+00
	DD-SRPSO (ε=0.1, β=0.7)	4.59E+01	1.16E+00

Table 5.2: Strategy analysis on selected CEC 2013 Benchmark Functions

Func.	Algorithm	Mean	STD.
F_{12}	DD-SRPSO (ε=0.03, β=0.4)	9.97E+01	1.89E+01
	DD-SRPSO (ε=0.03, β=0.5)	1.01E+02	3.15E+01
	DD-SRPSO(ε=0.03, β=0.6)	1.04E+02	2.59E+01
	DD-SRPSO (ε=0.03, β=0.7)	9.57E+01	1.36E+01
	DD-SRPSO (ε=0.05, β=0.4)	1.08E+02	2.00E+01
	DD-SRPSO (ε=0.05, β=0.5)	**8.34E+01**	1.60E+01
	DD-SRPSO (ε=0.05, β=0.6)	8.97E+01	1.72E+01
	DD-SRPSO (ε=0.05, β=0.7)	1.12E+02	2.47E+01
	DD-SRPSO (ε=0.1, β=0.4)	1.04E+02	1.90E+01
	DD-SRPSO (ε=0.1, β=0.5)	9.23E+01	1.77E+01
	DD-SRPSO (ε=0.1, β=0.6)	9.03E+01	1.71E+01
	DD-SRPSO (ε=0.1, β=0.7)	8.77E+01	1.22E+01
F_{18}	DD-SRPSO (ε=0.03, β=0.4)	3.14E+02	2.65E+01
	DD-SRPSO (ε=0.03, β=0.5)	3.38E+02	2.70E+01
	DD-SRPSO(ε=0.03, β=0.6)	3.25E+02	4.16E+01
	DD-SRPSO (ε=0.03, β=0.7)	3.50E+02	1.62E+01
	DD-SRPSO (ε=0.05, β=0.4)	2.18E+02	8.21E+01
	DD-SRPSO (ε=0.05, β=0.5)	**1.66E+02**	3.87E+01
	DD-SRPSO (ε=0.05, β=0.6)	1.99E+02	9.59E+01
	DD-SRPSO (ε=0.05, β=0.7)	2.92E+02	1.11E+02
	DD-SRPSO (ε=0.1, β=0.4)	2.04E+02	6.98E+01
	DD-SRPSO (ε=0.1, β=0.5)	2.24E+02	6.24E+01
	DD-SRPSO (ε=0.1, β=0.6)	2.15E+02	9.28E+01
	DD-SRPSO (ε=0.1, β=0.7)	2.32E+02	8.46E+01
F_{25}	DD-SRPSO (ε=0.03, β=0.4)	3.16E+02	1.65E+01
	DD-SRPSO (ε=0.03, β=0.5)	3.11E+02	1.00E+01
	DD-SRPSO(ε=0.03, β=0.6)	2.99E+02	1.30E+01
	DD-SRPSO (ε=0.03, β=0.7)	3.07E+02	1.63E+01
	DD-SRPSO (ε=0.05, β=0.4)	2.93E+02	1.13E+01
	DD-SRPSO (ε=0.05, β=0.5)	3.07E+02	1.60E+01
	DD-SRPSO (ε=0.05, β=0.6)	**2.78E+02**	1.49E+01
	DD-SRPSO (ε=0.05, β=0.7)	2.89E+02	1.03E+01
	DD-SRPSO (ε=0.1, β=0.4)	2.98E+02	1.43E+01
	DD-SRPSO (ε=0.1, β=0.5)	3.01E+02	8.56E+00
	DD-SRPSO (ε=0.1, β=0.6)	3.12E+02	1.57E+01
	DD-SRPSO (ε=0.1, β=0.7)	3.21E+02	1.46E+01

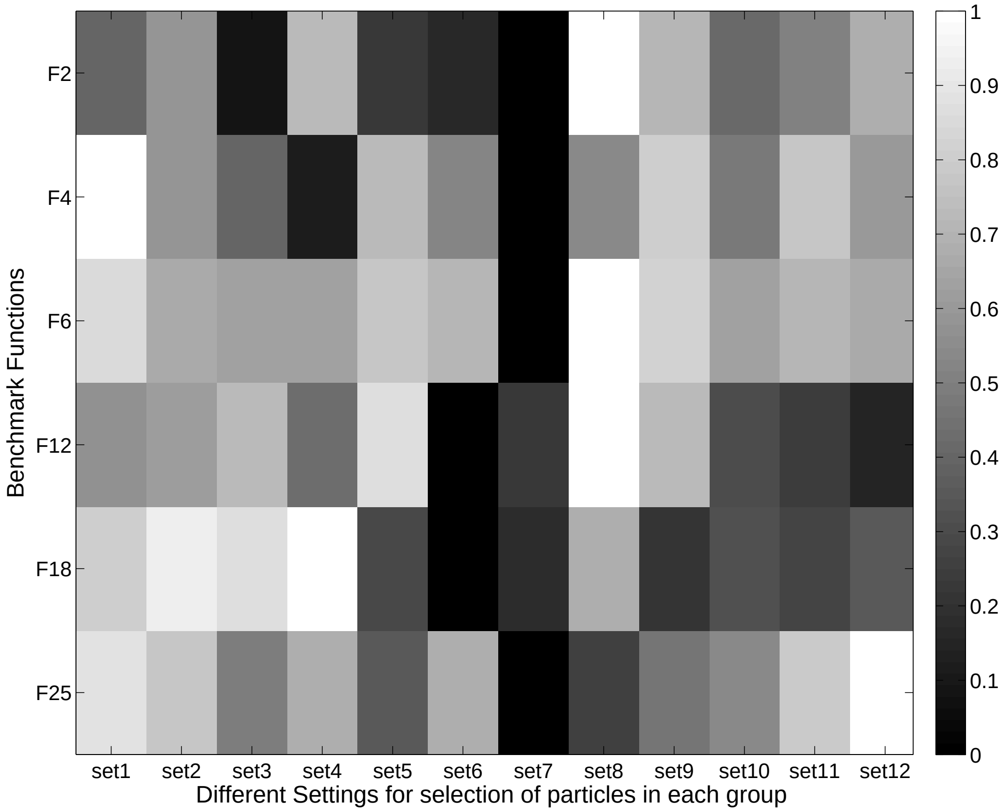

Figure 5.6: Heat map of the performance of different settings in the DD-SRPSO algorithm

5.3.3 Experimental results, comparison and discussion

In this section, performance of DD-SRPSO has been compared with well-known PSO variants and other evolutionary algorithms on all the CEC2013 benchmark functions.

5.3.3.1 Selected algorithms for comparison and parameter settings

It has been observed in the performance evaluation of SRPSO in chapter 3 that the performance of CLPSO, DMSPSO, FIPS were comparatively better than other selected algorithms. Therefore, only these three algorithms are selected in this evaluation. Further, a recently proposed human learning principles inspired algorithm has also been selected in this study. Further, from the performance evaluation of DMeSR-PSO in chapter 4, it was observed that CMAES was significantly better than other algorithms and equally good as DMeSR-PSO. CMAES together with some recently proposed evolutionary algorithms are included in the study. Finally, the performance of the two proposed algorithms in this thesis (SRPSO and DMeSR-PSO) is also compared with DD-SRPSO. To summarize, the following algorithms have been selected for comparison:

i **The Fully Informed Particle Swarm (FIPS) [32]:** The algorithm introduced a novel concept of search strategy for performance update during the search process. Here, the particles perform search using a weighted sum of the neighboring particles that resulted in better solutions on selected problems.

ii **Dynamic Multi-Swarm Particle Swarm Optimizer (DMSPSO) [89]:** A novel concept of dynamically changing neighborhood in the search process has been introduced in the algorithm that has provided much better convergence towards the optimum solution compared to the fixed neighborhood search.

iii **Comprehensive Learning Particle Swarm Optimization (CLPSO) algorithm [35]:** The algorithm is one of the most promising PSO variant widely accepted by the evolutionary community proven to be an efficient algorithm in accelerating the convergence towards the optimum solution. Compared to other variants of PSO, CLPSO searches more potential regions of the search space to find the global optima.

iv **Standard Particle Swarm Optimization (SPSO2011) algorithm [154]:** A newly proposed rotationally invariant version of the PSO algorithm. The algorithm has shown faster convergence characteristics with robustness on the rotated problems. The algorithm has recently been tested on the CEC2013 benchmark functions in [152].

v **Social Learning Particle Swarm Optimization (SL-PSO) algorithm [40]:** Recently proposed SL-PSO inspired by human social learning concepts has incorporated the idea of learning from any better performer from the group. The algorithm has consistently provided competitive results on complex functions.

vi **Covariance Matrix Adaptation Evolution Strategy (CMAES) [8]:** One of the most efficient and promising optimization algorithm developed using the covariance matrix of the normal mutation search distribution. Recently, Caraffini et al. proposed a much efficient CMA-ES variant utilizing the super-fit scheme referred to as CMAES with Re-sampled Inheritance Search (CMAES-RIS) and tested on the CEC2013 benchmark functions [155] where highly efficient solutions were achieved.

vii **Self Adaptive Differential Evolution (SaDE) [156]:** The algorithm is one of the most promising PSO variant widely accepted by the evolutionary community proven to be an efficient algorithm. Recently, SaDE has been proposed with more strategies for better performance referred to as Self-adaptive jDEsoo algorithm for solving the complex CEC2013 benchmark functions in [157].

viii **Genetic Algorithm with Three Parent Crossover (GA-TPC) [158]:** GA-TPC is a new variant of GA equipped with a diversity operative and generates three new offspring in every generation. The algorithm has successfully provided optimal/ near-optimal solutions to diverse problems.

ix **Self-Regulating Particle Swarm Optimization (SRPSO) algorithm [140]:** The algorithm proposed in chapter 3 in this thesis.

x **Dynamic Mentoring and Self-Regulation based Particle Swarm Optimization (DMeSR-PSO) algorithm [159]:** The algorithm proposed in chapter 4 in this thesis.

To have a fair comparison of performance, all the guidelines defined in the CEC2013 [153] evaluation criteria has been followed for systematic performance evaluation of DD-SRPSO. All the parameter setting and reported results for all the selected algorithms are adopted from their respective articles. The experiments have been conduction on the 50D CEC2013 benchmark functions using a swarm size of 100 particles. For all three human learning principles inspired PSO variants, SRPSO, DMeSR-PSO and DD-SRPSO the following settings have been used in the experimentation:

- $\omega = 1.05 - 0.5$,
- $c_1 = c_2 = 1.49445$,
- Vmax = 0.1 * Range
- Function Evaluations = 500000
- Total Run = 51 times

5.3.3.2 Performance evaluation

The mean and standard deviation of the performance for all the selected algorithms are provided in the tables 5.3 to 5.5. Table 5.3 presents the performance of the algorithms on the first 12 benchmark functions (F_1-F_{12}). Table 5.4 contains the results for the next 12 benchmark functions (F_{13}-F_{24}) and table 5.5 contains the results for the final 4 benchmark functions (F_{25}-F_{28}).

From the tables, one can easily identify that the CMAES-RIS has outperformed all the other algorithms on the unimodal functions by a significant margin. In fact, it has achieved the global optimum solution on the benchmark functions F_2 and F_4 that none of the compared algorithms have achieved. This suggests that the CMAES-RIS algorithm is the most suited for unimodal functions. There is a completely different scenario in the case of basic multimodal functions (F_6-F_{20}) and composition functions (F_{21}-F_{28}). A mixed performance among the algorithms has been observed on the basic multimodal functions. Here, GA-TPC and CLPSO have provided best solutions in one function each where CLPSO is the best performer in function F_{11} and GA-TPC is best performer in function F_{18}. Similarly, SL-PSO has produced the best performance in two functions, F_7 and F_9 whereas CMAES-RIS has achieved the best performance in three functions, F_6, F_{10} and F_{16}. The best performance among the selected algorithms in this category of benchmark functions has been achieved by DD-SRPSO and jDEsoo as they both have produced best performance in 4 functions. DD-SRPSO has achieved the best performance in functions F_8, F_{12}, F_{15} and F_{20} and jDEsoo has achieved the best performance in functions F_{13}, F_{14}, F_{17} and F_{19}.

Among the composite functions, the performance of DD-SRPSO is significantly better than all the other algorithms as it has provided best solutions in 4 out of 8 composite functions whereas none of the other algorithms has provided best solutions in more than 2 functions. The performance of five algorithms FIPS, CLPSO, SL-PSO, DMeSR-PSO and DD-SRPSO is identical in function F_{28} where all have achieved the same solution. It may be noted that the algorithms producing best results in function F_{28} are all PSO variants which suggest that PSO variants are most suitable for such a problem. In function F_{22}, jDEsoo has produced the best result and SL-PSO has achieved the best solution in function F_{27}. The performance of CMAES-RIS is best in functions F_{21} and F_{26} whereas DD-SRPSO has provided the best performance in functions F_{23}, F_{24} and F_{25}.

It should be noted that, the performance of DD-SRPSO is not good in functions F_{11}, F_{14} and F_{17} as SRPSO and DMeSR-PSO has produced better mean performances on these functions. If one study the characteristics of these functions, it will be seen that these problems are simple multimodal functions without rotated properties. Since, DD-SRPSO algorithm has been designed for the rotated problems, therefore its performance is marginally inferior to SRPSO and DMeSR-PSO on these functions.

Overall, DD-SRPSO has achieved the best mean performance in 11 out of the 28 benchmark functions whereas CMAES-RIS has provided better mean performances in 10 benchmark functions. This indicates that the performance of DD-SRPSO and CMAES-RIS are comparable on the set of benchmark functions whereby CMAES-RIS is the best performer on the unimodal functions and DD-SRPSO is the best performer on the composition functions. Furthermore, both have exhibited more or less similar performances

Table 5.3: Performances Evaluation on F_1-F_{12} CEC2013 Benchmark Functions

Algorithm	$\mathbf{F_1}$		$\mathbf{F_2}$		$\mathbf{F_3}$	
	Mean	STD.	Mean	STD.	Mean	STD.
FIPS	**0.00E+00**	0.00E+00	2.71E+07	5.01E+06	6.61E+08	2.82E+08
DMSPSO	8.27E-06	1.12E-05	1.11E+07	5.91E+06	3.58E+08	1.74E+08
CLPSO	**0.00E+00**	0.00E+00	6.55E+07	9.86E+06	5.92E+09	1.69E+09
SPSO2011	**0.00E+00**	0.00E+00	3.07E+06	1.07E+06	4.30E+08	5.80E+07
SL-PSO	**0.00E+00**	0.00E+00	2.22E+06	5.90E+05	1.31E+07	2.13E+07
jDEsoo	2.76E-08	3.25E-08	6.05E+05	2.46E+05	4.78E+07	6.86E+07
GA-TPC	**0.00E+00**	0.00E+00	4.76E+05	2.14E+05	1.06E+08	1.50E+08
CMAES-RIS	**0.00E+00**	0.00E+00	**0.00E+00**	0.00E+00	**2.83E+05**	7.80E+05
SRPSO	**0.00E+00**	0.00E+00	8.31E+06	3.39E+06	2.71E+08	4.24E+08
DMeSR-PSO	**0.00E+00**	0.00E+00	1.18E+06	1.67E+06	6.15E+08	3.62E+08
DD-SRPSO	**0.00E+00**	0.00E+00	2.07E+06	7.40E+05	1.01E+07	2.25E+08
	$\mathbf{F_4}$		$\mathbf{F_5}$		$\mathbf{F_6}$	
	Mean	STD.	Mean	STD.	Mean	STD.
FIPS	4.26E+04	7.63E+03	**0.00E+00**	0.00E+00	4.55E+01	1.13E+00
DMSPSO	3.21E+04	2.26E+03	5.32E-04	5.69E-04	4.74E+01	7.84E-01
CLPSO	6.47E+04	1.06E+04	**0.00E+00**	0.00E+00	4.79E+01	4.30E-01
SPSO2011	3.99E+04	5.68E+01	6.01E-08	3.85E-08	5.37E+01	3.40E+00
SL-PSO	3.82E+04	3.11E+03	**0.00E+00**	0.00E+00	4.52E+01	2.33E+00
jDEsoo	8.34E+04	1.56E+04	2.43E-06	1.14E-06	4.30E+01	3.71E+00
GA-TPC	3.33E+00	4.88E+00	4.77E+04	1.70E+05	4.72E+01	1.40E+01
CMAES-RIS	**0.00E+00**	0.00E+00	3.54E-08	1.44E-08	**9.51E+00**	1.41E+01
SRPSO	2.30E+02	1.07E+02	3.15E-08	6.06E-08	4.88E+01	1.13E+01
DMeSR-PSO	7.55E+01	3.02E+01	**0.00E+00**	0.00E+00	4.51E+01	1.29E+00
DD-SRPSO	2.16E+02	4.92E+01	**0.00E+00**	0.00E+00	3.21E+01	9.14E+00
	$\mathbf{F_7}$		$\mathbf{F_8}$		$\mathbf{F_9}$	
	Mean	STD.	Mean	STD.	Mean	STD.
FIPS	9.38E+01	6.94E+00	2.12E+01	2.51E-02	5.93E+01	2.20E+00
DMSPSO	5.63E+01	7.67E+00	2.12E+01	2.86E-02	4.72E+01	3.48E+00
CLPSO	1.14E+02	1.06E+01	2.12E+01	5.49E-02	5.71E+01	2.62E+00
SPSO2011	9.26E+01	2.32E+01	2.12E+01	4.93E-02	5.50E+01	1.79E+00
SL-PSO	**7.49E+00**	1.51E+00	2.12E+01	3.44E-02	**1.82E+01**	1.75E+00
jDEsoo	2.94E+01	1.29E+01	2.11E+01	3.84E-02	5.33E+01	9.78E+00
GA-TPC	4.17E+01	1.83E+01	2.12E+01	3.98E+02	7.43E+01	3.97E+00
CMAES-RIS	4.81E+01	2.16E+01	2.11E+01	6.20E-02	4.72E+01	2.76E+00
SRPSO	3.50E+01	1.16E+01	2.11E+01	4.27E-02	2.82E+01	3.98E+00
DMeSR-PSO	6.80E+01	1.00E+01	2.11E+01	2.66E-02	2.64E+01	2.18E+00
DD-SRPSO	3.17E+01	7.67E+00	**2.10E+01**	4.99E-02	2.77E+01	3.32E+00
	$\mathbf{F_{10}}$		$\mathbf{F_{11}}$		$\mathbf{F_{12}}$	
	Mean	STD.	Mean	STD.	Mean	STD.
FIPS	2.97E-01	7.31E-02	1.32E+02	1.91E+01	4.09E+02	1.02E+01
DMSPSO	6.99E+00	3.59E+00	5.92E+00	3.87E+00	1.26E+02	6.61E-01
CLPSO	2.95E+01	8.24E+00	**7.97E-05**	6.60E-05	3.31E+02	3.14E+01
SPSO2011	2.65E-01	1.66E-01	2.36E+02	8.25E+00	4.53E+02	1.28E+01
SL-PSO	2.41E-01	1.06E-01	2.65E+01	5.87E+00	3.39E+02	2.11E+01
jDEsoo	1.47E-01	7.66E-02	1.95E-02	1.39E-01	9.72E+01	2.56E+01
GA-TPC	1.05E-01	7.09E+02	5.57E+01	2.23E+01	9.83E+01	2.45E+01
CMAES-RIS	**8.30E-03**	5.86E-03	5.36E+01	1.31E+01	2.66E+02	1.02E+02
SRPSO	1.94E+00	3.33E-01	6.82E+01	1.45E+01	1.09E+02	2.00E+01
DMeSR-PSO	1.87E+00	2.39E-01	5.98E+01	7.88E+00	1.09E+02	3.33E+01
DD-SRPSO	2.06E-01	9.30E-02	6.94E+01	8.51E+00	**8.76E+01**	1.62E+01

Table 5.4: Performances Evaluation on F_{13}-F_{24} CEC2013 Benchmark Functions

Algorithm	**F_{13}**		**F_{14}**		**F_{15}**	
	Mean	STD.	Mean	STD.	Mean	STD.
FIPS	4.18E+02	1.89E+01	1.07E+04	7.91E+02	1.41E+04	3.95E+02
DMSPSO	2.36E+02	3.66E+01	1.87E+01	1.06E+01	9.22E+03	2.75E+03
CLPSO	4.07E+02	3.49E+01	1.44E+02	2.65E+01	1.04E+04	8.86E+02
SPSO2011	4.26E+02	2.25E+01	7.58E+03	1.96E+02	8.49E+03	4.60E+02
SL-PSO	3.43E+02	2.45E+01	1.08E+03	3.95E+02	1.23E+04	3.16E+03
jDEsoo	**1.76E+02**	2.37E+01	**8.01E+00**	6.76E+00	9.48E+03	1.06E+03
GA-TPC	1.93E+02	5.30E+01	2.55E+03	1.14E+03	9.84E+03	3.19E+03
CMAES-RIS	4.56E+02	8.54E+01	1.49E+03	3.07E+02	6.30E+03	6.74E+02
SRPSO	2.70E+02	6.03E+01	3.00E+03	5.95E+02	1.13E+04	2.22E+03
DMeSR-PSO	2.23E+02	4.20E+01	1.72E+03	2.03E+02	7.19E+03	1.24E+03
DD-SRPSO	2.03E+02	4.27E+01	3.44E+03	4.73E+02	**6.24E+03**	8.57E+02
	F_{16}		**F_{17}**		**F_{18}**	
	Mean	STD.	Mean	STD.	Mean	STD.
FIPS	3.50E+00	3.54E-01	3.56E+02	3.25E+01	4.44E+02	1.59E+01
DMSPSO	2.01E+00	8.40E-01	6.47E+01	3.58E+00	2.08E+02	9.37E+01
CLPSO	2.82E+00	6.00E-01	6.28E+01	1.42E+00	4.52E+02	1.86E+01
SPSO2011	2.95E+00	2.46E-01	3.12E+02	7.68E+00	3.26E+02	1.58E+01
SL-PSO	3.33E+00	3.27E-01	3.70E+02	1.41E+01	3.97E+02	9.63E+00
jDEsoo	3.13E+00	3.95E-01	**5.08E+01**	3.25E-01	2.18E+02	3.11E+01
GA-TPC	3.68E+00	3.88E-01	1.15E+02	2.00E+01	**1.68E+02**	1.02E+02
CMAES-RIS	**8.66E-02**	4.20E-02	1.02E+02	1.04E+01	4.18E+02	5.24E+01
SRPSO	2.71E+00	3.52E-01	1.12E+02	1.41E+01	3.92E+02	3.18E+01
DMeSR-PSO	2.39E+00	1.69E-01	1.14E+02	7.11E+00	4.02E+02	1.12E+01
DD-SRPSO	2.28E+00	2.05E-01	1.12E+02	7.07E+00	2.00E+02	1.21E+02
	F_{19}		**F_{20}**		**F_{21}**	
	Mean	STD.	Mean	STD.	Mean	STD.
FIPS	2.94E+01	1.72E+00	2.17E+01	4.98E-01	3.62E+02	2.73E+02
DMSPSO	4.15E+00	1.12E+00	2.35E+01	1.01E+00	7.88E+02	4.22E+02
CLPSO	3.92E+00	6.39E-01	2.38E+01	5.50E-01	4.91E+02	2.19E+02
SPSO2011	3.91E+01	6.09E-01	2.28E+01	0.00E+00	6.54E+02	8.77E+01
SL-PSO	9.18E+00	5.24E+00	2.24E+01	3.47E-01	7.60E+02	4.06E+02
jDEsoo	**2.24E+00**	5.46E-01	2.15E+01	4.32E-01	8.24E+02	4.01E+02
GA-TPC	8.92E+00	3.17E+00	2.35E+01	8.02E-01	7.93E+02	3.63E+02
CMAES-RIS	5.04E+00	9.33E-01	2.43E+01	5.49E-01	**2.85E+02**	2.20E+02
SRPSO	5.54E+00	1.00E+00	2.15E+01	8.43E-01	8.18E+02	3.34E+02
DMeSR-PSO	6.71E+00	1.03E+00	2.11E+01	1.46E+00	8.84E+02	1.08E+02
DD-SRPSO	3.89E+00	1.34E+00	**1.95E+01**	1.16E+00	7.53E+02	1.27E+02
	F_{22}		**F_{23}**		**F_{24}**	
	Mean	STD.	Mean	STD.	Mean	STD.
FIPS	1.05E+04	1.04E+03	1.46E+04	2.37E+02	3.37E+02	9.04E+00
DMSPSO	5.54E+01	4.76E+01	7.95E+03	1.06E+03	2.71E+02	1.50E+01
CLPSO	6.07E+02	1.45E+02	1.15E+04	8.47E+02	3.55E+02	3.28E+00
SPSO2011	9.77E+03	2.75E+02	1.09E+04	8.11E+02	3.34E+02	3.99E+01
SL-PSO	1.14E+03	3.05E+02	1.10E+04	4.27E+03	2.46E+02	8.35E+00
jDEsoo	**3.10E+01**	3.89E+01	9.48E+03	1.02E+03	2.89E+02	1.20E+01
GA-TPC	3.51E+03	1.90E+03	1.08E+04	3.71E+03	3.79E+02	2.18E+01
CMAES-RIS	2.39E+03	3.67E+02	8.37E+03	1.01E+03	3.22E+02	1.95E+01
SRPSO	3.59E+03	9.25E+02	9.69E+03	2.48E+03	2.88E+02	1.91E+01
DMeSR-PSO	2.12E+03	3.63E+02	7.62E+03	1.85E+03	2.87E+02	1.19E+01
DD-SRPSO	9.64E+02	4.45E+02	**6.71E+03**	1.90E+03	**2.14E+02**	7.29E+00

Table 5.5: Performances Evaluation on F_{25}-F_{28} CEC2013 Benchmark Functions

Algorithm	F_{25}		F_{26}	
	Mean	STD.	Mean	STD.
FIPS	3.73E+02	1.57E+01	2.50E+02	1.07E+02
DMSPSO	3.29E+02	1.18E+01	3.88E+02	2.62E+01
CLPSO	3.94E+02	8.18E+00	2.07E+02	1.30E+00
SPSO2011	4.10E+02	1.97E+01	4.52E+02	7.15E+01
SL-iPSO	2.94E+02	8.23E+00	3.35E+02	7.91E+00
jDEsoo	3.17E+02	1.09E+01	3.97E+02	2.36E+01
GA-TPC	3.89E+02	4.42E+00	4.28E+02	4.11E+01
CMAES-RIS	3.66E+02	9.94E+00	**2.00E+02**	3.25E-01
SRPSO	3.45E+02	1.48E+01	3.57E+02	5.92E+01
DMeSR-PSO	2.84E+02	9.54E+00	2.30E+02	7.61E+01
DD-SRPSO	**2.76E+02**	8.54E+00	2.64E+02	8.22E+01
Algorithm	F_{27}		F_{28}	
	Mean	STD.	Mean	STD.
FIPS	1.74E+03	1.30E+02	**4.00E+02**	6.00E-03
DMSPSO	1.19E+03	8.34E+01	1.68E+03	1.75E+03
CLPSO	1.87E+03	6.00E+01	**4.00E+02**	1.22E-02
SPSO2011	1.63E+03	7.54E+01	1.66E+03	9.94E+01
SL-PSO	**7.29E+02**	8.12E+01	**4.00E+02**	0.00E+00
jDEsoo	1.16E+03	1.24E+02	9.43E+02	1.19E+03
GA-TPC	2.08E+03	2.36E+02	4.59E+02	4.24E+02
CMAES-RIS	1.25E+03	2.08E+02	1.24E+03	1.53E+03
SRPSO	1.13E+03	1.06E+02	7.02E+02	9.27E+02
DMeSR-PSO	1.12E+03	9.58E+01	**4.00E+02**	3.47E-04
DD-SRPSO	9.68E+02	9.78E+01	**4.00E+02**	9.35E+02

on the basic multimodal functions. To further investigate the performance significance of the algorithms, the mean performances are normalized on a 0 to 1 scale and plotted using a heat map. The normalized mean performances of all the selected algorithms are presented in figure 5.7. In the figure, the benchmark functions are represented by the y-axis and the algorithms are represented on the x-axis. In the map, the best performance is represented by the black colour whereas the white colour represents the worst performance. From the heat map, it can be seen that with the exception of 3 functions, the DD-SRPSO algorithm has consistently produced the results in the dark colour ranges i.e. the better performing range and can be termed as the best performer. Similarly, the performance of DMeSR-PSO, CMAES-RIS and jDEsoo are better compared to other algorithms. This indicates that the performance of DD-SRPSO, DMeSR-PSO, CMAES-RIS and jDEsoo are comparable to each other and must be further investigated for their significance over each other. Therefore, a rank based analysis on the mean performances of the algorithms has been performed to further investigate the significance of

DD-SRPSO compared to algorithms on the set of CEC2013 benchmark functions.

5.3.3.3 A rank based analysis

In this section, the mean performances of the algorithms on the benchmark functions will be ranked to determine the best performing algorithm according to the average rank achieved on the 28 benchmark functions.

The individual ranks, average ranks, variance in rank and the final ranks of all the selected algorithms are given in table 5.6. From the table, one can see that the with an exception of two functions (F_{11} and F_{14}) DD-SRPSO algorithm has consistently achieved better ranks compared to the other algorithms and this is also evident in the average ranks. The average rank of the DD-SRPSO algorithm is 2.964 whereas the average ranks of second and third best algorithms (DMeSR-PSO and CMAES-RIS) are 4.500 and 4.571 respectively. This suggests that there is a significant difference in the average ranks of DD-SRPSO and other better performing algorithms.

Further, the variance in rank is also provided in the table to determine the consistency and robustness of the algorithms over the entire range of benchmark functions. The variance in rank of the DD-SRPSO algorithm is 4.925 which is the lowest among the selected algorithms. Also, the variance in rank of SRPSO and DMeSR-PSO is lower compared to other algorithms. This clearly proves that the DD-SRPSO algorithm is providing better solutions on the CEC2013 benchmark functions with consistency and robustness. Based on the average ranks, the final ranks are given to the algorithms provided in the last row of the table 5.6. Here, the top rank is of the DD-SRPSO algorithm followed by DMeSR-PSO, CMAES-RIS, jDEsoo, SL-PSO, DMSPSO, SRPSO, GA-TPC, CLPSO, FIPS and SPSO2011 respectively. Furthermore, the robustness and consistency of the algorithms can be ranked by considering the variance in ranks. Here, the top rank is of the DD-SRPSO algorithm followed by SRPSO, SPSO2011, DMeSR-PSO, DMSPSO, SL-SPO, jDEsoo, GA-TPC, CMAES-RIS, FIPS and CLPSO respectively.

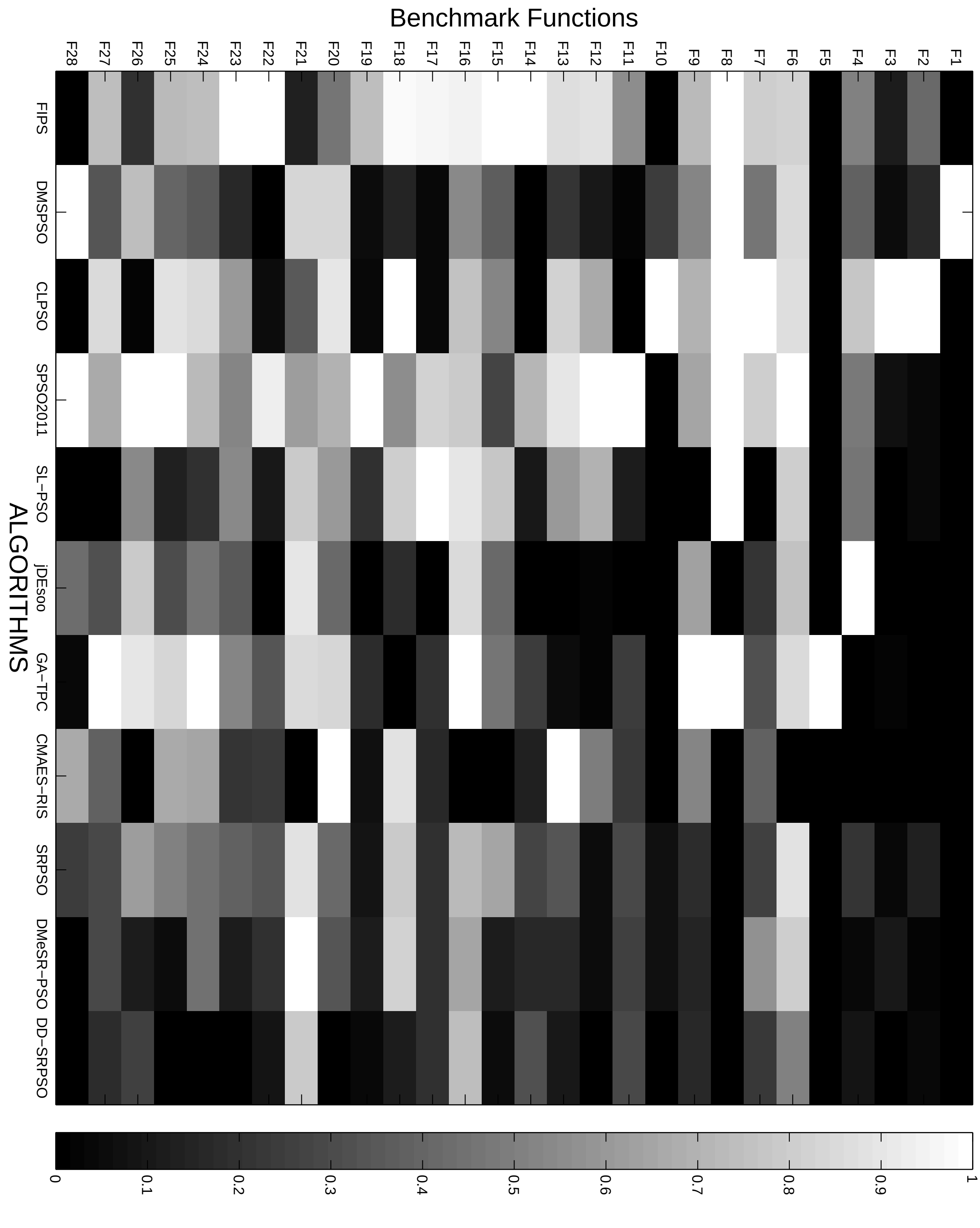

Figure 5.7: Heat Map for the Mean Performance of the Selected Algorithms on all the Benchmark Functions

Table 5.6: Rank based analysis of Mean Performances

Func.	Individual Ranking on CEC2013 Benchmark Functions										
	FIPS	DMSPSO	CLPSO	SPSO2011	SL-PSO	jDEsoo	GA-TPC	CMAES-RIS	SRPSO	DMeSR-PSO	DD-SRPSO
F_1	1	11	1	1	1	10	1	1	1	1	1
F_2	10	9	11	7	6	3	2	1	8	4	5
F_3	10	7	11	8	3	4	5	1	6	9	2
F_4	9	6	10	8	7	11	2	1	5	3	4
F_5	1	10	1	8	1	9	11	7	6	1	1
F_6	6	8	9	11	5	3	6	1	10	4	2
F_7	10	7	11	9	1	2	5	6	4	8	3
F_8	6	6	6	6	6	2	6	2	2	2	1
F_9	10	5	9	8	1	7	11	5	4	2	3
F_{10}	7	10	11	6	5	3	2	1	9	8	4
F_{11}	10	3	1	11	4	2	6	5	8	7	9
F_{12}	10	6	8	11	9	2	3	7	4	4	1
F_{13}	9	5	8	10	7	1	2	11	6	4	3
F_{14}	11	2	3	10	4	1	7	5	8	6	9
F_{15}	11	5	8	4	10	6	7	1	9	3	2
F_{16}	10	2	6	7	9	8	11	1	5	4	3
F_{17}	10	3	2	9	11	1	8	4	5	7	5
F_{18}	10	3	11	5	7	4	1	9	6	8	2
F_{19}	10	4	3	11	9	1	8	5	6	7	2
F_{20}	5	8	10	7	6	3	8	11	3	2	1
F_{21}	2	7	3	4	6	10	8	1	9	11	5
F_{22}	11	2	3	10	5	1	8	7	9	6	4
F_{23}	11	3	10	8	9	5	7	4	6	2	1
F_{24}	9	3	10	8	2	6	11	7	5	4	1
F_{25}	8	5	10	11	3	4	9	7	6	2	1
F_{26}	4	8	2	11	6	9	10	1	7	3	5
F_{27}	9	6	10	8	1	5	11	7	4	3	2
F_{28}	1	11	1	10	1	8	6	9	7	1	1
Average Rank	7.893	5.893	6.750	8.107	5.179	4.679	6.500	4.571	6.000	4.500	**2.964**
Variance in Rank	10.988	7.655	14.935	6.396	9.485	10.152	10.778	10.921	5.037	7.444	**4.925**
Final Rank	10	6	9	11	5	4	8	3	7	2	**1**

The experimental results, heat map and the rank based analysis on the mean performances suggest that DD-SRPSO is a better performing algorithm on the set of benchmark functions compared to all the other selected algorithms. To further investigate the significance of DD-SRPSO over other algorithms a statistical comparative analysis has been performed.

5.3.3.4 A statistical comparison

This study provides the significance of the compared algorithm over each other. Here, a statistical test has been conducted using the non-parametric Friedman test on the average rank followed by the pairwise post-hoc Bonferroni-Dunn test [128]. The average rank difference in mean performances of all the algorithms compared to DD-SRPSO is presented in table 5.7. The table also provides the statistical test values F-score for Friedman test and the critical difference for the Benforroni-Dunn test.

The computed F-score of the Friedman test is 8.0976. In this study, there are 11 algorithms and 28 functions for which the Friedman F-critical value obtained from the F-table is 1.8659. Since, the F-critical value is comparatively much smaller than the computed F-scores therefore; the null hypothesis can be easily rejected. Hence, it may be concluded that the performance of algorithms selected for comparison on the CEC2013 benchmark functions are significantly different from each other.

Next, the significance of the performance of DD-SRPSO over other algorithms can be determined using the Benforroni-Dunn test. The minimum required difference in the average ranks to signify an algorithm over others as provided in the table is 1.5349 for a confidence level of 95%. The average rank difference of DMeSR-PSO and CMAES-RIS with respect to DD-SRPSO is 1.536 and 1.607 respectively that are smaller than 1.5349. Hence, it can be concluded that the mean performance of DD-SRPSO is significantly better than all the other selected algorithms with a confidence level of 95%. Furthermore, it was observed in chapter 4 that the performance of DMeSR-PSO and CMAES are almost

Table 5.7: Average Ranks and Statistics for $\alpha = 0.05$ significance level

Algorithms	Average Rank Difference w.r.t. DD-SRPSO
FIPS	4.929
DMSPSO	2.929
CLPSO	3.786
SPSO2011	5.143
SL-PSO	2.214
jDEsoo	1.714
GA-TPC	3.536
CMAES-RIS	1.607
SRPSO	3.304
DMeSR-PSO	1.536
F-score (Friedman)	8.0976
Critical Difference (Benforroni-Dunn)	1.5349

similar to each other on the set of CEC2005 benchmark functions. The same has been observed here in this study whereby the average rank difference of both DMeSR-PSO and CMAES-RIS with respect to DD-SRPSO are almost similar. Hence, from these findings it can be inferred that the performance of DMeSR-PSO is comparable to that of CMAES and DD-SRPSO is a better performer compared to both the algorithms.

In the next section, the box plots and convergence graphs are presented for comparative performance analysis of the three proposed algorithm SRPSO, DMeSR-PSO and DD-SRPSO.

5.3.4 Comparative performance analysis of SRPSO, DMeSR-PSO and DD-SRPSO

From the experimental evaluation on CEC2005 and CEC2013 it has been proved that the proposed algorithms are efficient, consistent and robust. In this section, an analysis has been performed to study the convergence characteristics of SRPSO, DMeSR-PSO and DD-SRPSO on the CEC2013 benchmark functions using convergence graphs and box plots.

First, the convergence graphs of the mean performances of SRPSO, DMeSR-PSO and DD-SRPSO are plotted to compare their convergence characteristics. The convergence graphs are presented in figures 5.8 to 5.12. Figure 5.8, presents the convergence of SRPSO, DMeSR-PSO and DD-SRPSO on benchmark functions F_1 to F_6. From the figure, it can be seen that all the three algorithms have converged to the optimum solution in functions F_1 and F_5 around 25% of the total iterations and among them DD-SRPSO has converged in minimum time. In the remaining functions, mixed performances have been observed whereby DD-SRPSO has converged closer to the optimum in functions F_3 and F_6 and DMeSR-PSO has best convergence in functions F_2 and F_4. In most of the cases, DD-SRPSO has converged to a solution faster than SRPSO and DMeSR-PSO that indicates the faster convergence characteristics of the algorithm. Further, in functions F_2 and F_4 a downward slope has been observed that provides an inference that if the algorithms are allowed to run for more iterations they might converge to the optimum solution.

The convergence graphs for functions F_7 to F_{12} are presented in figure 5.9. In the figure, it is clearly shown that in function F_{10} DD-SRPSO has converged towards the optimum solution in almost 35% of the total iterations whereas other algorithms have converged to a point slightly away from the optimum solution. In functions F_{11}, DMeSR-PSO has converged to a point closer to the optimum solution. In the remaining three functions F_8, F_9 and F_{12}, the performance of DD-SRPSO has been better than SRPSO and DMeSR-PSO as it has converged faster to a point closer to the optimum solution.

Figure 5.10 contains the convergence analysis on functions F_{13} to F_{18}. From the figure, it can be seen that DD-SRPSO has shown much better convergence compared to SRPSO and DMeSR-PSO in functions F_{13}, F_{15}, F_{16} and F_{18} whereby it has converged to a point closer to the optimum solution at around 50% of the total iterations. In function F_{17}, all the algorithms have shown identical convergence characteristics as they have converged

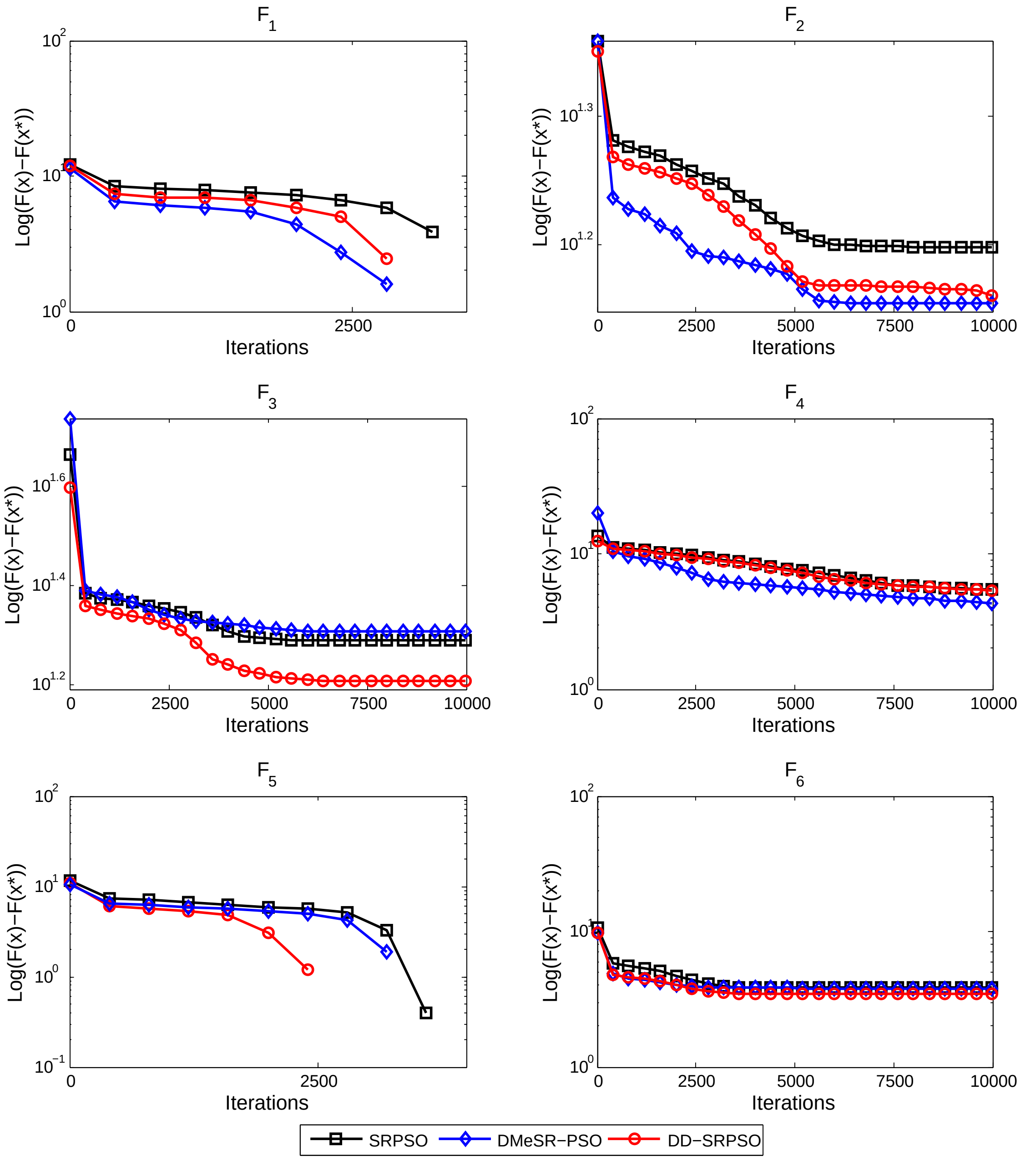

Figure 5.8: Mean Convergence plots for functions F_1 to F_6

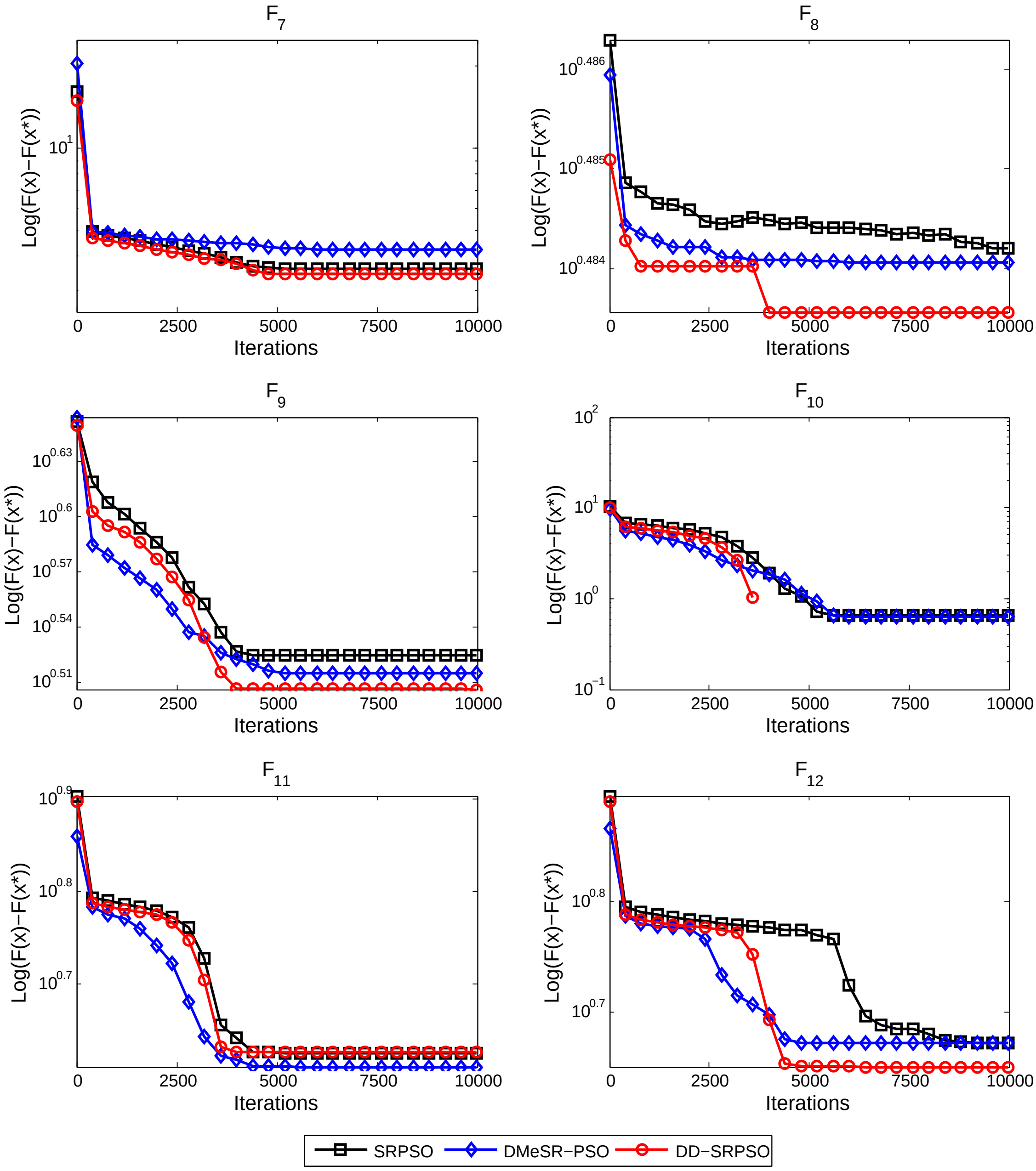

Figure 5.9: Mean Convergence plots for functions F_7 to F_{12}

to the same point in 50% of the total iterations. In function F_{14}, the performance of SRPSO and DMeSR-PSO is better than DD-SRPSO.

The convergence graphs for functions F_{19} to F_{24} are presented in figure 5.11. From the figure, it can be easily identified that DD-SRPSO has converged to a point much closer to the optimum solution in all the functions. In function F_{20}, a descending graph has been observed suggesting that the algorithms are capable of achieving the optimum solution for this function with an allocation of sufficient amount of time. Similar observation has been made in function F_{24} for the DD-SRPSO algorithm where the algorithm has descended to a lower point towards the end of the iterations.

The convergence graphs of the last four functions F_{25} to F_{28} are provided in figure 5.12. As shown in the figure, in all of the functions the algorithms have shown faster convergence characteristics as they have converged to a point in almost 35% of iterations. Among them, DD-SRPSO has converged to a point closer to the optimum solution in functions F_{25} and F_{27} whereas DMeSR-PSO has achieved the same in function F_{26}. In function F_{28}, both DD-SRPSO and DMeSR-PSO have shown identical convergence through faster convergence towards a point closer to the optimum solution.

Overall, from the convergence analysis on the set of 28 benchmark functions it may be concluded that the algorithms have shown faster convergence characteristics closer to the optimum solution. Among them, DD-SRPSO has exhibited better convergence characteristics by converging towards optimum/ near-optimum solution in most of the functions at aroung 35% of total iterations. A continuous descending convergence graph has also been observed in a few functions suggesting that with an allowance of sufficient amount to time the algorithms will be able to locate the optimum solution.

Next, the box plots of Error fitness values (F(x)-F(x*)) for the entire 51 runs are given to demonstrate the convergence analysis towards the achieved solution. The box plots for the entire set of 28 benchmark functions are provided in Appendix D and to showcase the performance, the box plots of selected 6 benchmark functions are provided in figure

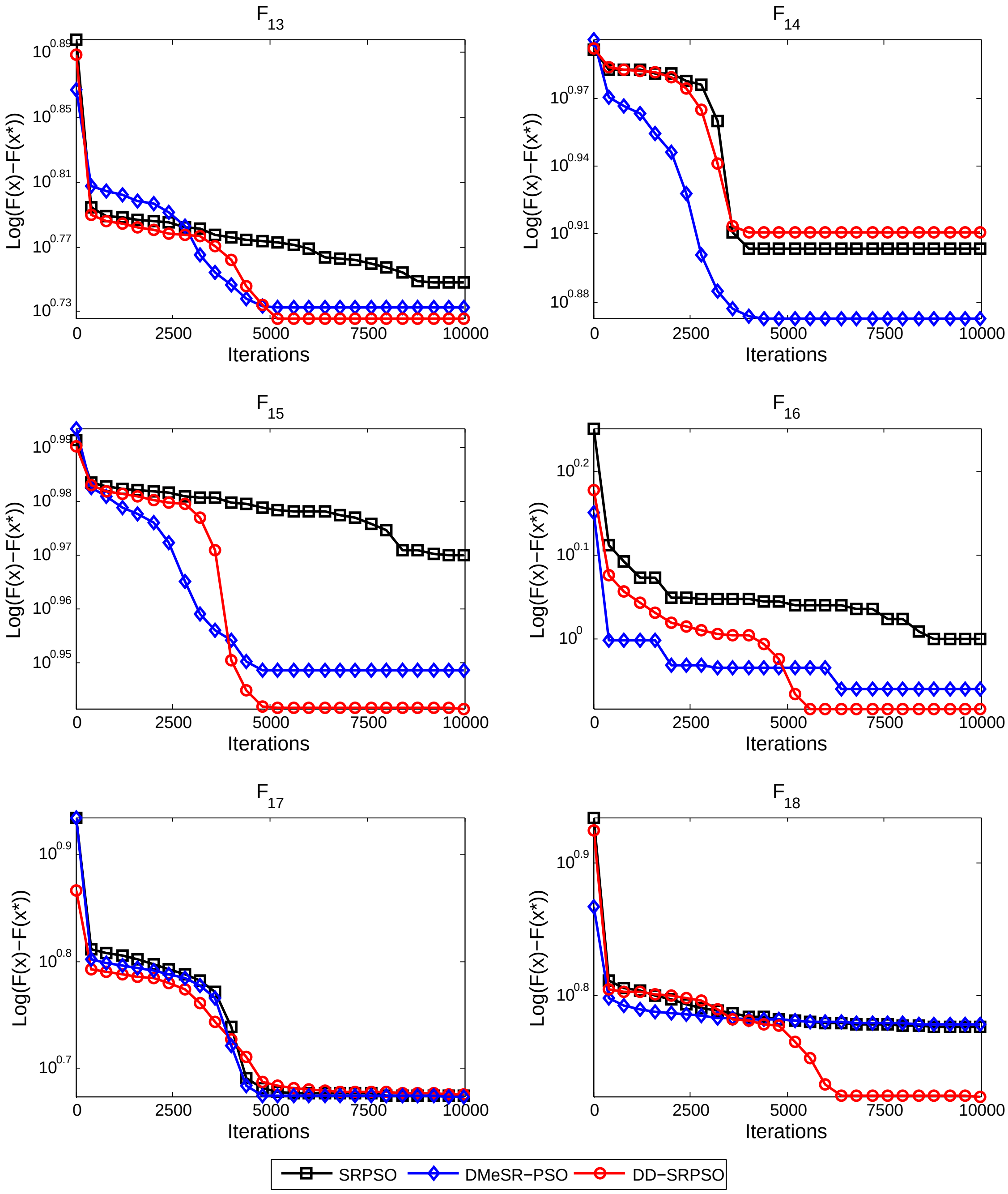

Figure 5.10: Mean Convergence plots for functions F_{13} to F_{18}

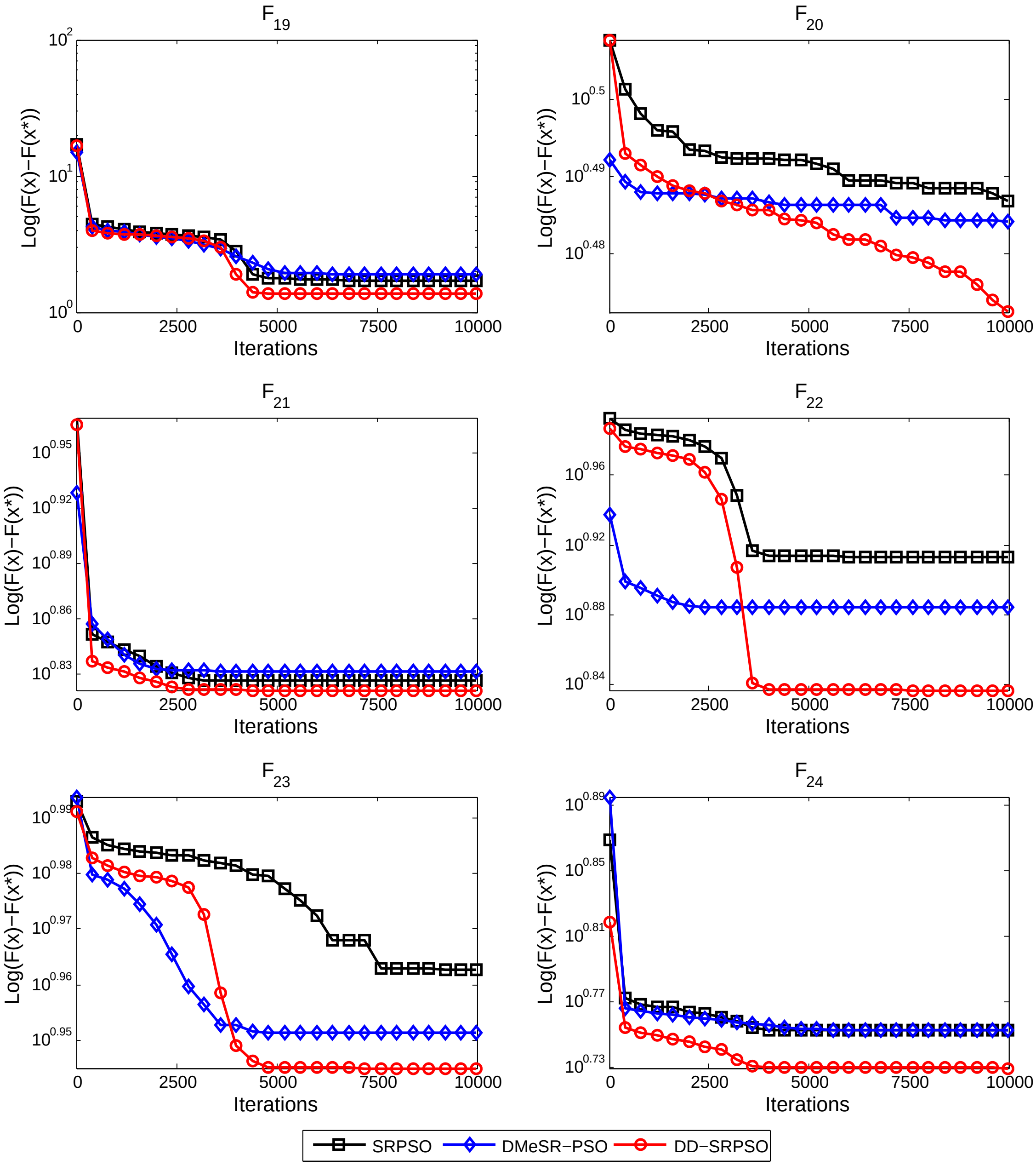

Figure 5.11: Mean Convergence plots for functions F_{19} to F_{24}

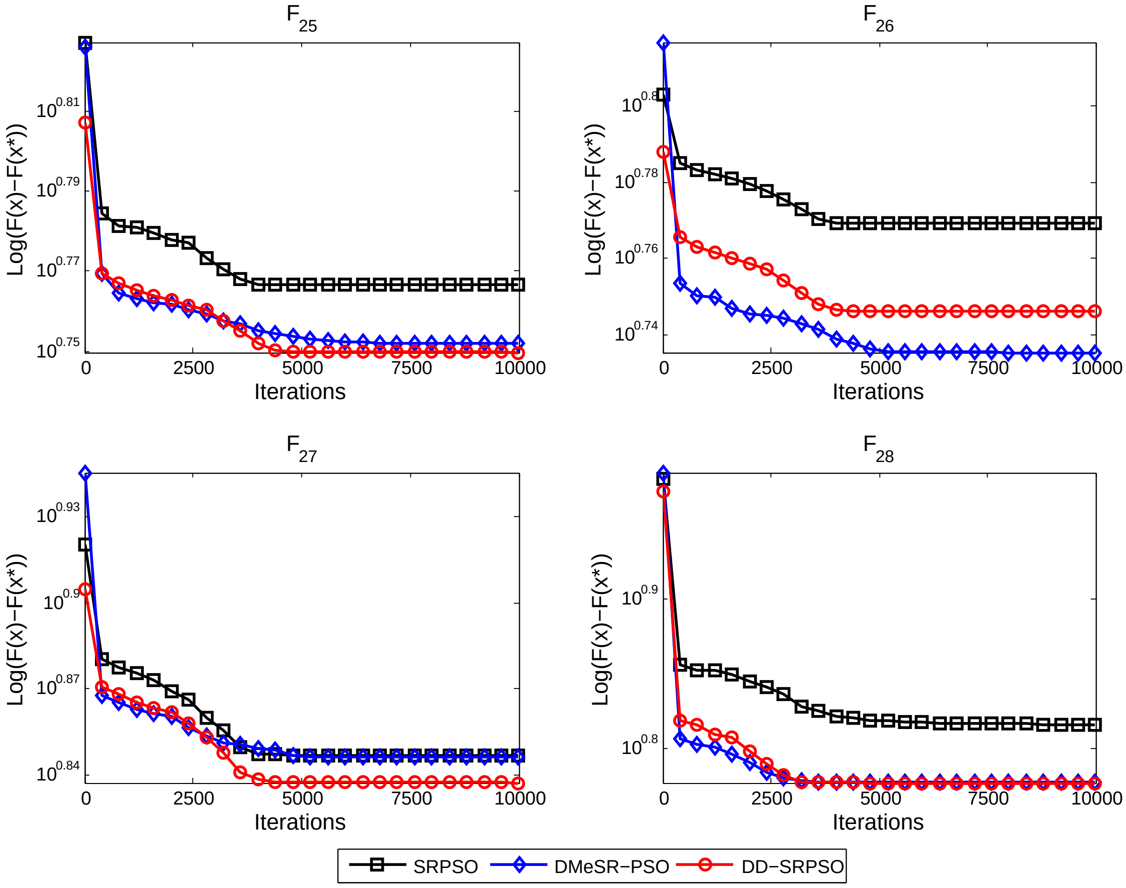

Figure 5.12: Mean Convergence plots for functions F_{25} to F_{18}

5.13. The figure contains box plots for functions F_3, F_6, F_{10}, F_{11}, F_{19} and F_{25} to provide an illustration of the performance on all categories of benchmark functions. In functions F_3 and F_6, SRPSO and DD-SRPSO have exhibited much robust performances with an exception of few outlier as represented by a very small box. On the other hand, in the same functions the performance of DMeSR-PSO is spread over a range of values. This suggests that in most of the runs the algorithm has converged to different values far from each other. The performance of all three algorithms are almost identical on functions F_{10}, F_{19} and F_{25} as the box plots are presenting the robustness of the algorithms. The boxes are very small illustrating that the algorithms have converged closer to the same value most of the time and among them DD-SRPSO has converged to a point which is closer to the optimum solution. A different convergence behaviour has been observed in function F_{11} as here SRPSO and DMeSR-PSO have shown a high level of accuracy across the 51 runs. The performance of DD-SRPSO is spread over a range of values as represented by a bigger box. These observations have been made on all the 28 benchmark functions as summarized below:

i Algorithms have achieved a high level of accuracy across the 51 runs.

ii In most cases, smaller boxes have been observed.

iii The algorithms have succeeded in providing solutions closer to a point in every run.

iv The algorithms have exhibited robustness and consistency on the benchmark functions.

v Among them, DD-SRPSO has provided much improved solutions.

To summarize, the performance of SRPSO, DMeSR-PSO and DD-SRPSO on the set of 28 benchmark functions has been fairly good. The high level level of accuracy across the 51 runs has been observed from the box plots and faster convergence closer to the optimum solution has been observed from the convergence graphs. In most of the cases, the algorithms have provided solutions closer to the optimum solution with robustness and consistency. Among them, the DD-SRPSO algorithm has converged to a point closer to the optimum solution is 35% of total iteration in almost all the functions. Hence, it can be concluded that all the three algorithms have shown consistent performance with sustainability and robustness on the diverse set of benchmark functions whereby DD-SRPSO has provided much improved solutions.

In this chapter, a socially guided interaction among the particles has been implemented in the SRPSO algorithm. The algorithm is referred to as Directionally Driven Self-Regulating Particle Swarm Optimization (DD-SRPSO) algorithm. In DD-SRPSO,

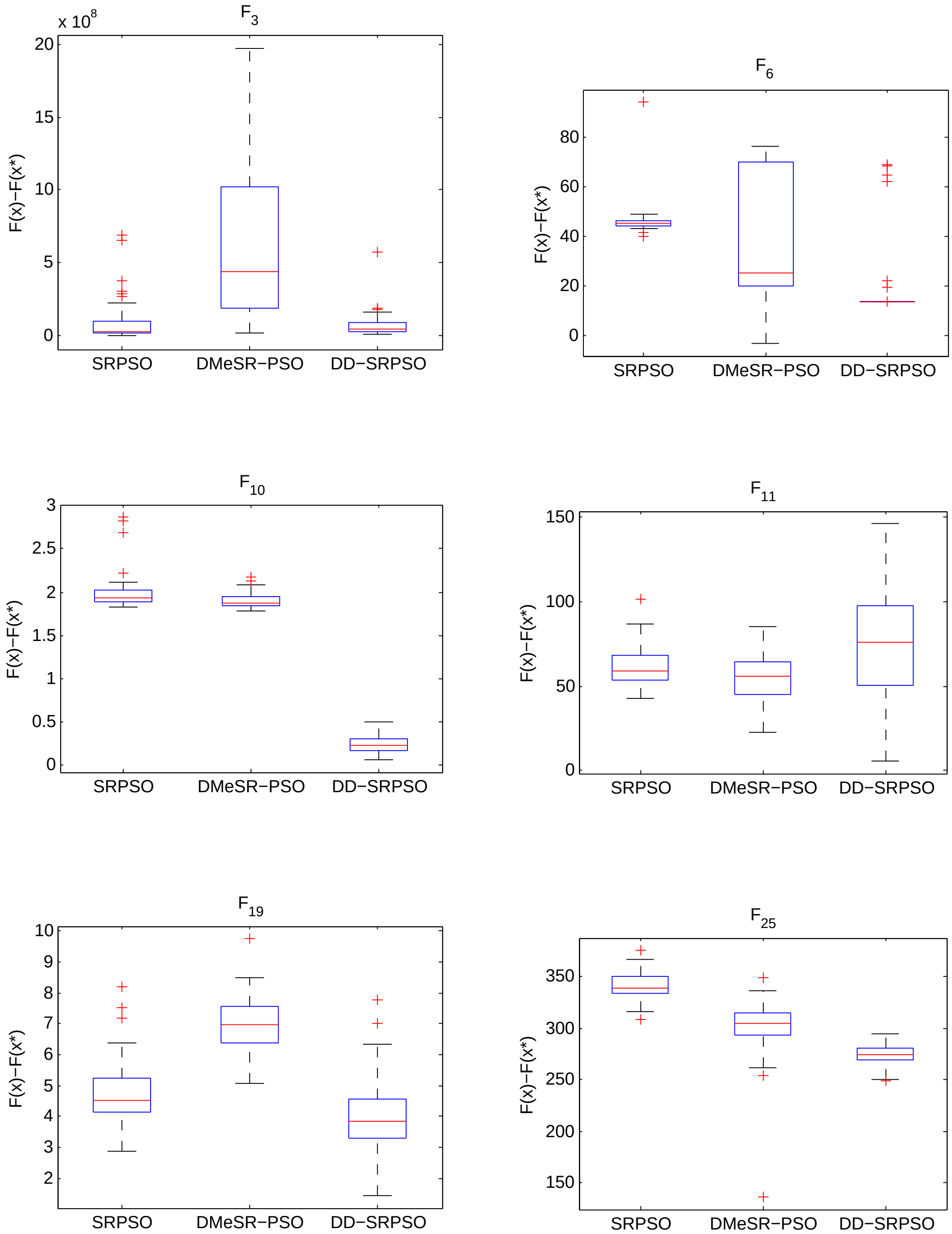

Figure 5.13: Box Plot for Selected Benchmark Functions

top 5% better performing particles are used to guide the bottom 5% poorly performing particles. Also, to tackle the rotated characteristics of the problems a rotational invariant strategy has been introduced for the particles. The particles have utilized the following strategies:

- Directional update strategy,
- Rotational invariant strategy and
- Both the strategies of the SRPSO algorithm.

Using the proposed learning strategies, the algorithm has successfully:

- Improved the convergence characteristics of the SRPSO algorithm.
- Provided better performances on the rotated functions,
- Provided faster convergence closer to the optimum solution with robustness on diverse CEC2013 benchmark problems.
- Statistically outperformed the widely accepted state-of-the-art search based optimization algorithms.

Hence, it may be concluded that the idea of a more stiff scheme, where the lesser performing particles are fully guided by elite particles have indeed helped in achieving improved performance. The DD-SRPSO algorithm has exhibited highly efficient performances on complex composite functions. Further, it has maintained the robustness of SRPSO and DMeSR-PSO algorithms by successfully providing consistent performance over a range of diverse benchmark problems.

In the next chapter, the performance of SRPSO, DMeSR-PSO and DD-SRPSO has been investigated on the complex real-world optimization problems.

Chapter 6

Performance Evaluation on Practical Optimization Problems

In the previous chapters 3, 4 and 5, we proposed three human learning principles inspired PSO variants referred to as SRPSO, DMeSR-PSO and DD-SRPSO respectively. All the three variants has provided better solutions with efficiency and robustness and among them DD-SRPSO has exhibited the best performance.

In this chapter, we present the performance comparison of the proposed SRPSO, DMeSR-PSO and DD-SRPSO with the better performing evolutionary algorithms on the practical optimization problems reported in the literature GA Multi-Parent Crossover (GA-MPC) [160] and Self-Adaptive Multi-Operator DE (SAMODE) [161] on CEC2011 real-world benchmark problems [162].

6.1 Performance Evaluation on Real-World Optimization Problems

The previous analysis on CEC2005 benchmark problems [59] in chapter 3 and 4 has proved that the proposed SRPSO and DMeSR-PSO algorithms are significantly better than state-of-the-art PSO variants as well as the meta-heuristic population based optimization algorithms in providing optimum/near optimum solutions. Furthermore, the analysis on more complex CEC2013 benchmark problems [153] has proved that the DD-

SRPSO algorithm is significantly better than state-of-the-art PSO variants as well as the meta-heuristic population based optimization algorithms in providing optimum/near-optimum solutions with consistency, efficiency and robustness over diverse set of benchmark functions. Next, to prove the effectiveness of the proposed human learning principles inspired PSO algorithms, it is necessary to evaluate its performance on complex real-world optimization problems. First, the performance will be tested on benchmark real-world optimization problems from CEC2011 [162] and then in the next section, DD-SRPSO will be applied to the staging problem of multi-stage launch vehicle trajectory optimization.

Recently, in CEC2011 [162], a comprehensive account of some real-world optimization problems were provided by [162] for testing the performance of optimization algorithms in solving complex practical problems. These problems have been specifically designed to evaluate the performance of different stochastic optimization algorithm on some real-world problems. A set of 8 practical problems have been selected in this study. A brief description of each practical problem as given in [159] is given next.

6.1.1 Parameter estimation for frequency-modulated sound waves

These days, several modern musical systems are highly dependent on Frequency-Modulated (FM) sound synthesis. The modern sound system requires such a FM synthesizer that can generate a sound similar to the target sound. The FM synthesizer is a highly complex multimodal and non-separable optimization problem. The problem consists of six parameters represented by the vector $\vec{X} = \{a_1, \omega_1, a_2, \omega_3, a_3, \omega_3\}$ and the task is to find the optimum value of these parameters such that a sound similar to the target sound is generated. The sound waves using these parameters and the target sound waves are given by:

$$y(t) = a_1.sin(\omega_1.t.\theta + a_2.sin(\omega_2.t.\theta + a_3.sin(\omega_3.t.\theta))) \tag{6.1}$$

$$y_0(t) = (1.0).sin((5.0).t.\theta - (1.5).sin((4.8).t.\theta + (2.0).sin((4.9).t.\theta))) \tag{6.2}$$

respectively, where $\theta = 2\pi/100$ and the six parameters are bounded within the range [-6.4 6.35].

The fitness function for the FM synthesizer is defined as:

$$f_1(\vec{X}) = \sum_{t=0}^{100}(y(t) - y_0(t))^2 \tag{6.3}$$

More details can be found in [162].

6.1.2 Lennard-Jones potential problem

The Lennard-Jones (LJ) potential problem is a potential energy minimization problem. The problem is classified as a complex multi-modal optimization problem with an exponential number of local minima. The problem is very challenging because the optimization algorithm has to predict the position of the atom in the molecules such that the molecule has the minimum energy. LJ pair potential for N-atoms is defined as:

$$V_N(p) = \sum_{i=1}^{N-1}\sum_{j=i+1}^{N}(r_{ij}^{-12} - 2r_{ij}^{-6}) \tag{6.4}$$

where

$\vec{p_i}$ in the Cartesian coordinates is $\{\vec{x_i}, \vec{y_i}, \vec{z_i}\}$, $i = 1, \cdots, N$ and $r_{ij} = \|\vec{p_j} - \vec{p_i}\|_2$ with gradient

$$\nabla_j V_N(p) = -12\sum_{i=1, j\neq i}^{N}(r_{ij}^{-14} - r_{ij}^{-8})(\vec{p_j} - \vec{p_i}), \quad j = 1, \cdots, N \tag{6.5}$$

LJ potential has minimum value at a particular distance between the two points. The task of the optimization function is to determine the minimal energy between two atoms as a function of their distance r. The variation in LJ potential with pair distance r is given by:

$$V(r) = r^{-12} - 2 * r^{-6} \tag{6.6}$$

Further details about the problem can be found in [162].

6.1.3 The bi-functional catalyst blend optimal control problem

The Bifunctional catalyst is a chemical process that converts the methylcyclopentane to benzene in a tubular reactor. The problem is classified as a hard multi-modal optimal control problem that consists of numerous local optima (reportedly, there are as many as 300 local optima in this problem). The problem is defined by a set of seven differential equations:

$$\dot{x_1} = -k_1 x_1 \tag{6.7}$$

$$\dot{x_2} = k_1 x_1 - (k_2 + k_3) x_2 + k_4 x_5 \tag{6.8}$$

$$\dot{x_3} = k_2 x_2 \tag{6.9}$$

$$\dot{x_4} = -k_6 x_4 + k_5 x_5 \tag{6.10}$$

$$\dot{x_5} = -k_3 x_2 + k_6 x_4 - (k_4 + k_5 + k_8 + k_9) x_5 + k_7 x_6 + k_{10} x_7 \tag{6.11}$$

$$\dot{x_6} = k_8 x_5 - k_7 x_6 \tag{6.12}$$

$$\dot{x_1} = k_9 x_5 - k_{10} x_7 \tag{6.13}$$

where x_i and k_i are the mole fractions and rate constants respectively. The constants k_i are the function of catalyst blend $u(t)$ such that:

$$k_i = c_{i1} + c_{i2} u + c_{i3} u^2 + c_{i4} u^3, \qquad i = 1, 2, \cdots, 10 \tag{6.14}$$

The optimal control problem is to find $u(t)$ at times $0 \leq t \leq t_f$ where $t_f = 2000g\ h/mol$ such that the concentration index of the reactor:

$$J_1 = x_7(t_f) \times 10^3 \tag{6.15}$$

that represents the benzene concentration at the exit of the reaction is maximized. Further details about the problem are available in [162].

6.1.4 Optimal control of a non-linear stirred tank reactor

One of the benchmark optimization problems in [163] is a multi-modal optimal control problem. The stirred tank reactors are used in the industries to convert reactants into products. The problem is a first-order irreversible chemical reaction in a continuous stirred tank reactor. The operation at highly nonlinear operating point results in a very challenging control problem. As a result, the non-linear structure of the process requires highly efficient tuning processes and extensive testing for desirable outcome that poses a more difficult control optimization problem. The process is modeled by two non-linear differential equations:

$$\dot{x_1} = -(2 + u)(x_1 + 0.25) + (x_2 + 0.5)exp(25x_1/(x_1 + 2)) \tag{6.16}$$

$$\dot{x_2} = 0.5 - x_2 - (x_2 + 0.5)exp(25x_1/(x_1 + 2)) \tag{6.17}$$

where u, x_1 and x_2 are the flow rate of the cooling fluid, dimensionless steady state temperature and deviation from dimensionless steady state concentration respectively. The objective for the optimization algorithm here is to find a value of u such that the performance index is minimized:

$$J_2 = \int_0^{t_f=0.72} (x_1^2 + x_2^2 + 0.1u^2)dt \tag{6.18}$$

All the details about the problem are available in [162].

6.1.5 Tersoff potential function minimization problem

The tersoff potential governs the interaction of silicon atoms that are connected through strong covalent bonding. The problem is defined as the evaluation of the inter atomic potentials and for silicon, the tersoff has given two parameterizations, namely, Sc(B) and Sc(C). The problem is a complex multimodal and differentiable function. The tersoff potential energy function for molecular clusters of N atoms, whose positions are defined as $\vec{X} = \{\vec{X_1}, \vec{X_2}, \cdots, \vec{X_N}\}$ where $\vec{X_i}s\ i \in \{1, 2, \cdots, N\}$ is a three-dimensional vector, is defined as:

$$f_2(\vec{X_1}, \vec{X_2}, \vec{X_3}) = E_1(\vec{X_1}, \vec{X_2}, \vec{X_3}) + \cdots + E_N(\vec{X_1}, \vec{X_2}, \vec{X_3}) \tag{6.19}$$

where the total potential is the sum of individual potentials and the tearsoff potential of an individual atom is defined as:

$$E_i = \frac{1}{2} \sum_{j \neq i} f_c(r_{ij})(V_R(r_{ij}) - B_{ij} V_A(r_{ij})),\ \forall i \tag{6.20}$$

where r_{ij} is the distance between atom i and j. V_R, V_A, $f_c(r_{ij})$ and B_{ij} are the repulsive term, attractive term, switching function and the main-body term respectively. The details of these terms are available in [162]. Using the details, the cluster X of N atoms can be redefined as

$x = \{x_1, x_2, \cdots, x_n\}, x \in IR^{(3N-6)}$

where the search region for both Tersoff potential model, Si(B) and Si(C) is

$\Omega = \{\{x_1, x_2, \cdots, x_n\} | -4.25 \leq x_1, x_i \leq 4.25, i = 4, \cdots, n,\ 0 \leq x_2 \leq 4,\ 0 \leq x_3 \leq \pi\}$.

Finally, the cost function is redefined as:

$$f_2(x) = E_1(x) + E_2(x) + \cdots + E_N(x), \quad x \in \Omega \tag{6.21}$$

From the above practical problem, two different problems are generated as Si(B) and Si(C) that are required to be solved separately by the optimizer.

6.1.6 Spread spectrum radar poly phase code design

Pulse compression technique is the most widely accepted technique in radar systems. Any radar system that uses pulse compression requires an accurate mechanism for the choice of the appropriate waveform. The polyphase codes offer convenience and ease of implementation through digital processing techniques, which is based on the properties of the aperiodic autocorrelation function and the assumption of coherent radar pulse processing in the receiver. The problem is a continuous min-max non-linear non-convex global optimization problem with numerous local optima. The min-max model based on the properties of the aperiodic auto-correlation function is defined as:

$$Global\ min f_3(x) = max\{\Phi_1(X), \Phi_2(X), \cdots, \Phi_{2m}(X)\} \tag{6.22}$$

where $X = \{(x_1, x_2, \cdots, x_D) \epsilon R_D | 0 \leq x_j \leq 2\pi\}$, $m = 2D - 1$ and $\Phi(X)$ is

$$\begin{aligned} \Phi_{2i-1}(X) &= \sum_{j=i}^{D} cos(\sum_{k=|2i-j-1|+1}^{j} x_k), & i = 1, 2, \cdots, D \\ \Phi_{2i}(X) &= 0.5 + \sum_{j=i+1}^{D} cos(\sum_{k=|2i-j-1|+1}^{j} x_k), & i = 1, 2, \cdots, D-1 \\ \Phi_{m+i}(X) &= -\Phi_i(X), & i = 1, 2, \cdots, m \end{aligned} \tag{6.23}$$

The objective of the problem is to minimize the module of the biggest among the samples of the auto-correlation function Φ having a complex envelope of the compressed radar pulse at the optimal receiver output that makes it a NP-hard problem. All the details about the problem are available in [164] and [162].

6.1.7 Transmission network expansion planning problem

The Transmission Network Expansion Planning (TNEP) problem without security constraints is used in power lines expansion for determining the number of new lines required to be constructed. The problem requires the construction in such a way that the cost of expansion plan is minimum and no overloads are produced during the planning horizon. The basic model of TNEP is available in [165] using which the objective function for the expansion cost without security constraint is:

$$f_4 = \sum_{l \in \Omega} c_l n_l + W_1 \sum_{ol} (abs(f_l) - \overline{f_l}) + W_2(n_l - \overline{n_l}) \tag{6.24}$$

where the first term represents the total investment cost and the second and third term represent the violation cases. W_1 and W_2 are constants, l represents the number of lines and ol represents the number of overflow lines. The main objective in TNEP is to find the set of transmission lines that are required to be constructed such that there are no overloads produced and the minimized cost of the expansion plan is achieved. More details about the problem are available in [162].

6.1.8 Comparative performance evaluation

In this section, the mean performances of SRPSO, DMeSR-PSO and DD-SRPSO on the selected real-world optimization problems are compared with the winner and runner-up algorithms of the CEC2011 competition [162]. The two algorithms selected for comparison are:

i **GA Multi-Parent Crossover (GA-MPC) [160]:** GA-MPC an improved version of GA where a randomized operator as multi-parent crossover has been introduced to improve the efficiency of the GA and resulted in providing better solutions.

ii **Self-Adaptive Multi-Operator DE (SAMODE) [161]:** SAMODE is a highly efficient variant of DE where multiple search operators are used in each generation by the DE algorithm using an adaptive approach. Using these multiple search operators better optimal solutions have been achieved by SAMODE.

The experiments are conducted following the guidelines of CEC2011 [162]. The population size for SRPSO, DMeSR-PSO and DD-SRPSO has been selected as 100. The best and mean performances for 25 runs with a maximum of 150,000 function evaluations are presented in table 6.1. From the table, it can be seen that the DD-SRPSO algorithm has successfully outperformed other algorithms by achieving lowest best results in all practical problems. Further, its mean performance is also best in 6 out of 8 problems. The performances of DMeSR-PSO and DD-SRPSO are significantly better than other on these practical problems. Compared to GA-MPC and SAMODE, both the algorithms have achieved either same or better best fitness values in all the functions. Both DMeSR-PSO and DD-SRPSO has outperformed GA-MPC and SAMODE in solving the following problems:

- Non-Linear Stirred Tank Reactor problem.
- Tersoff Potential Function Minimization problem.

- Spread Spectrum Radar Poly phase Code Design problem.

Further, SRPSO has provided comparable performances on the set of practical problems. It has achieved the same best results as that of other algorithms in 5 out of 8 problems and the same mean result in 3 out of 8 problems. Based on both best and mean performances, the algorithms can be ranked as DD-SRPSO followed by DMeSR-PSO, GA-MPC, SAMODE and SRPSO respectively. All these practical problems are bound constraint where constraint handing mechanism is not required to control the search. Nevertheless, there are a lot of real world problems that are governed by equality and inequality constraints. For tackling such problems, the algorithm requires an efficient constraint handling mechanism. Therefore, a new equality constraint handling mechanism has been incorporated in the DD-SRPSO framework.

6.2 Development of an Equality Constraint Handling Mechanism in the DD-SRPSO Algorithm

An important area in the optimization field is the optimization of constrained problems. Most of the problems are governed with constraints because of some physical, geometric and other limitations. Generally, a constraint optimization problem is defined by:

Minimize $f(X)$

subject to:

$$h_i(X) = 0, \qquad i = 1, 2, \cdots, m$$
$$g_j(X) \leq 0, \qquad j = 1, 2, \cdots, p$$

where m and p are the number of equality and inequality constraints respectively. The point X is the feasible point satisfying all the constraints.

Similar to the basic PSO algorithm, all the proposed algorithms in this thesis, SRPSO, DMeSR-PSO and DD-SRPSO are devised for unconstrained problems. In order to make

Table 6.1: Performance on selected CEC2011 Real-World Optimization Problems

Problem	**GA-MPC**		**SAMODE**		**SRPSO**		**DMeSR-PSO**		**DD-SRPSO**	
	Best	Mean	Best	Mean	Best	Mean	Best	Mean	Best	Mean
f_1	**0.000E+00**	**0.000E+00**	**0.000E+00**	1.212E+00	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**	**0.000E+00**
V_N	**-2.842E+01**	**-2.770E+01**	**-2.842E+01**	-2.707E+01	**-2.842E+01**	-2.646E+01	**-2.842E+01**	-2.750E+01	**-2.842E+01**	-2.757E+01
J_1	**1.151E-05**	**1.151E-05**	**1.151E-05**	**1.151E-05**	**1.151E-05**	**1.151E-05**	**1.151E-05**	**1.151E-05**	**1.151E-05**	**1.151E-05**
J_2	**1.377E+01**	**1.382E+01**	**1.377E+01**	1.394E+01	1.380E+01	1.392E+01	**1.377E+01**	**1.382E+01**	**1.377E+01**	**1.382E+01**
$f_2(si(B))$	-3.685E+01	-3.504E+01	-3.684E+01	-3.359E+01	-3.685E+01	-3.471E+01	-6.210E+01	**-5.863E+01**	**-6.602E+01**	-5.525E+01
$f_2(si(C))$	-2.917E+01	-2.749E+01	-2.917E+01	-2.763E+01	-2.917E+01	-2.516E+01	-4.204E+01	-3.895E+01	**-4.480E+01**	**-4.195E+01**
f_3	**5.000E-01**	7.484E-01	**5.000E-01**	8.166E-01	**5.000E-01**	7.968E-01	**5.000E-01**	7.364E-01	**5.000E-01**	**6.235E-01**
f_4	**2.200E+02**	**2.200E+02**	**2.200E+02**	**2.200E+02**	**2.200E+02**	**2.200E+02**	**2.200E+02**	**2.200E+02**	**2.200E+02**	**2.200E+02**

them capable of dealing with constrained problems an efficient constraint handling mechanisms should be incorporated. In literature, there are several constraint handling mechanisms available such as lagrangian [166], projection [166], penalty approach [167, 168], repair approach [169], MO-based separatist approach [170], stochastic ranking [171], hybrid approaches [172] etc. Among them, several methods have already been adopted in the PSO algorithm and successfully applied to several real-world practical problems [173, 174, 175, 176, 177]. A comprehensive survey of available constraint handling strategies in line with there adaptation in the PSO algorithm is provided in [178]. It has also been stated in the literature that penalty approaches are very commonly used [178] but these approaches have many disadvantages like determining a suitable penalty factor, tuning the parameters, higher chances of exploring infeasible regions etc. Therefore, researchers took inspiration from Deb's approach [179] but due to existent overpressure towards feasible regions this approach leads towards premature convergence [178]. Therefore, this study explores a new constraint handling mechanism for solving constrained problems.

6.2.1 A new constraint handling mechanism

This constraint handling approach is inspired from the repair approach [169]. In the repair approach, the infeasible individuals are converted to feasible ones in a separate repair process. This approach can be easily applied to the equality constraints of any problems as repairing infeasible individuals can be conducted easily. Hence, an equality constraint handling mechanism incorporated within the DD-SRPSO framework. This approach is different from repair approach as there is no separate repair process but the repair is performed before the position update of particles. As soon as the velocities of the particles are calculated, the equality constraints are applied to the velocities an then the new positions of the particles are calculated. As a result, the new calculated positions (X) never violates the constraints and hence convergence faster.

The mechanism is incorporated in such a way that m number of equality constraints must satisfy the condition of being equal to $h_i(X)$. For any problem of equality constraints, the total number of constraints can be treated as the dimensions of the problem. Any mathematical operator can govern the relationship between m and $h_i(X)$. For simplicity, an example of multiplication operator can be taken for a problem of three constraints. During the entire search process, the algorithm must follow the rule that whenever there is an update the multiplication of three constraints is equal to a constant k. Then, the calculated position (X) can be applied to the objective function $f(X)$ for performance evaluation. Such a mechanism has been incorporated in the DD-SRPSO algorithm and the algorithm is referred to as DD-SRPSO with constraint handing mechanism (DD-SRPSO-CHM).

6.2.2 The DD-SRPSO-CHM Pseudocode

Incorporating the proposed equality constraint handling mechanism, the pseudo-code for DD-SRPSO-CHM is summarized in Algorithm 4.

```
Initialization:
for the particles (i) do
    Initialize the position X_i randomly in the search range (X_min, X_max)
    Randomly initialize velocity V_i
end
Using the fitness function, calculate the fitness values;
From X_i, determine the personal best position;
The DD-SRPSO Loop:
Apply the entire loop as that in 3 in chapter 5
The CHM Loop:
Normalize the velocity with respect to constant k;
Apply the operator for satisfying the equality constraint;
Update the position of each particle using equation (4.6);
```

Algorithm 4: The DD-SRPSO-CHM Algorithm

Next, the performance of DD-SRPSO-CHM has been evaluated on finding the optimal configurations of multi-stage launch vehicle design problem.

6.3 Application of DD-SRPSO-CHM in Optimizing Multi-Stage Launch Vehicle Configuration

The modern approach of selecting missions for multi-stage launch vehicle is largely based on cost-to-performance ratio, i.e., getting the most for the least within an acceptable level of risk. One of the most important tasks during the mission is the design and optimization of suitable/optimal mission trajectories. In case of multi-launch vehicles, the suitable/optimal mission trajectory is defined by maximizing the payload with respect to overall propellant consumption. Here, the problem of finding the optimal mission trajectory is multi-objective in nature; the multiple objectives are converted into a single objective with mission constraints. The primary objective is considered as main objective and the secondary objectives are converted as mission constraints. Therefore, in multi-scale launch vehicle, one need to calculate the total velocity required to place the satellite at appropriate orbit. In this process, one should consider the losses due to atmosphere, gravity and gain in speed due to earth rotation. The objective in multi-stage launch vehicle design is to design stages such that the payload fraction is a maximum. More details about the design and analysis of multi-stage launch vehicle can be found in [46, 180].

6.3.1 Problem definition: Multi-stage launch vehicle configuration

The launch of a satellite or space vehicle consists of a period of powered flight during which the vehicle is lifted above the Earth's atmosphere and accelerated to orbital velocity by a rocket, or launch vehicle. Powered flight concludes at burnout of the rocket's last stage at which time the vehicle begins its free flight. During free flight the space vehicle is assumed to be subjected only to the gravitational pull of the Earth. If the vehicle moves far from the Earth, its trajectory may be affected by the gravitational influence of the sun, moon, or another planet.

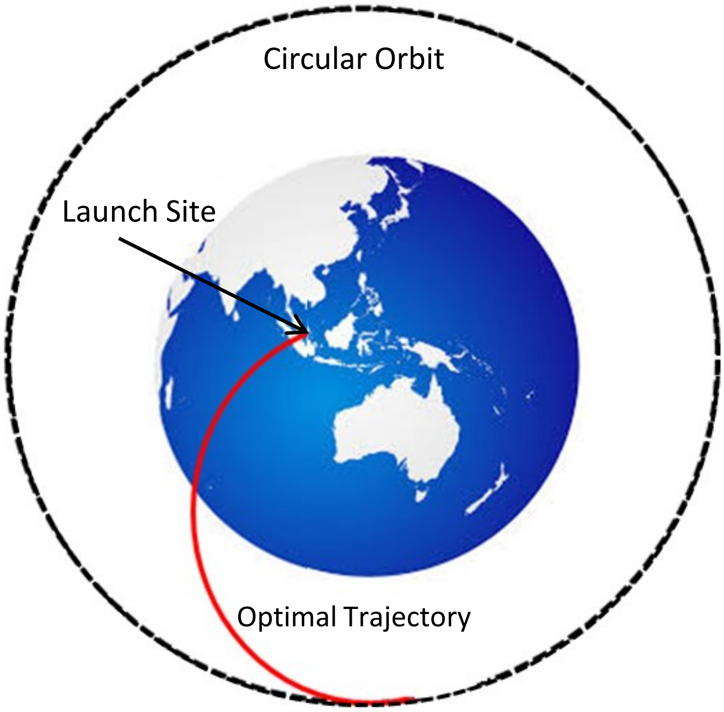

Figure 6.1: Typical Launching of Satellite to Circular Orbit

To place a satellite in a given orbit, one should accelerate the launch vehicle from standstill to orbital velocity (v_{orbit}) by overcoming the drag losses. During the acceleration process, the vehicle experience forces due to gravitation pull and atmospheric drag. Since, Earth is rotating at a constant speed it will add a component in the acceleration process. Figure 6.1 shows the typical trajectory from the earth surface to circular orbit. Figure 6.2 shows the various components influencing the acceleration process. In single-stage-to-orbit, the launch vehicle must be accelerated from the ground to the orbital velocity. The rocket should overcome the losses and place the satellite in the orbit. The total velocity (ΔV) required to place the satellite in a given orbital altitude is

$$\Delta V = \sqrt{(v_{orbit} - v_{boost} + v_{drag})^2 + (v_g)^2} \tag{6.25}$$

where v_{orbit} is the orbital velocity, v_{boost} is the gain in velocity, v_{drag} is the loss in velocity due to drag force and v_g is the loss in velocity due to gravity.

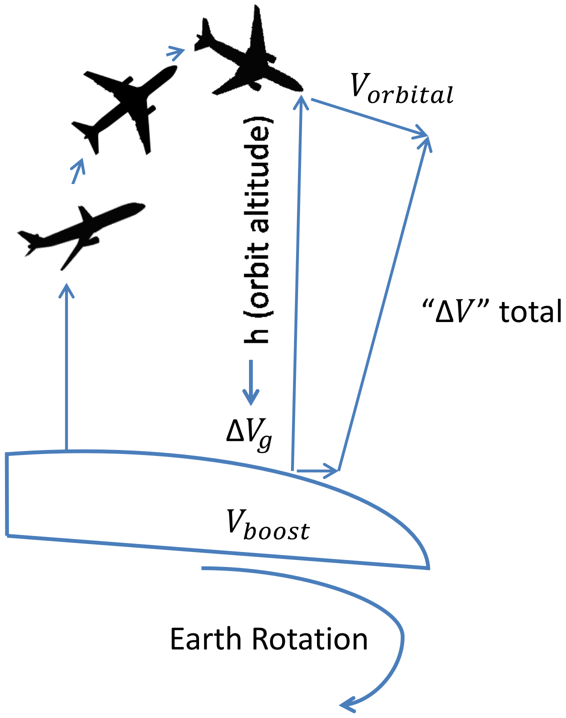

Figure 6.2: Various Components Influencing the Satellite Launch

Since, the single stage rocket carry the empty weight till orbital height they require more propellant mass. Multistage rockets allow improved payload capability for vehicles with a high ΔV requirement such as launch vehicles or interplanetary spacecraft. In a multistage vehicle, propellant is stored in smaller, separate tanks rather than a larger single tank as in a single-stage vehicle. Since each tank is discarded when empty, energy is not expended to accelerate the empty tanks, so a higher total ΔV is obtained. Alternatively, a larger payload mass can be accelerated to the same total ΔV . For convenience, the separate tanks are usually bundled with their own engines, with each discard-able unit called a stage. In multistage vehicle, the stages are designed such that the payload fraction is a maximum. The payload fraction (PF) is defined as:

$$PF = \frac{m_{payload}}{m_{oi}} \tag{6.26}$$

where $m_{payload}$ is the final mass payload and m_{oi} is the total vehicle weight when stage

Table 6.2: Parameters required for Staging

Payload	Structural Fraction	I_{sp}
For all stages 450 KG 180 KM	Stage 1 - 14% Stage 2 - 09% Stage 3 - 17%	Stage 1 - 293 (S) Stage 2 - 290 (S) Stage 3 - 293 (S)

i is ignited. For a multistage vehicle with dissimilar stages, the overall vehicle payload fraction depends on how the ΔV requirement is partitioned among stages. Payload fractions will be reduced if the ΔV is partitioned sub-optimally.

6.3.2 Performance evaluation, comparison and discussion

For this study, a three stage launch vehicle capable of placing the satellite at the Earth orbit has been considered. The first and second stages are having solid motors and last stage is having liquid motor. The vehicle is launched from the pad near equatorial region. The payload, specific Impulse (I_{sp}) and structural fractions are given in Table 6.2. The objective of staging is to find the optimal velocity contribution such that the payload fraction is a maximum and the constraints are satisfied. Using staging results, one can find the vehicle configuration and find the pitch program to place the satellite in appropriate orbit. The objective here is to place the payload of mass 450 KG in the Earth orbit 180 KM away. The velocity contribution due to atmospheric effect is considered as 12% of orbital velocity.

The DD-SRPSO-CHM algorithm has been applied to the staging design for determining the optimal configurations of the launch vehicle. The configurations are summarized in table 6.3. For the given payload, the orbital and ideal velocities and maximum paylaod are set as 7796.1 m/s, 8774.8 m/s 23795 KGs respectively. The configurations in terms of velocity contribution ΔV) , propellant mass (M_p) and gross weight (M_g) in every stage is presented in the table. These results match to the stage configuration data published in the literature [46, 180]. This suggests that DD-SRPSO-CHM has successfully provided valid stage configurations they can be used for the initial design purpose.

Table 6.3: Optimal Staging design configurations proposed by the DD-SRPSO-CHM algorithm

Vehicle Configuration				
Stage	Orbit: 180 KM, v_{orbit} = 7796.1, ΔV = 8774.8			
	ΔV in m/s	M_p in KG	M_g in KG	m_{oi} in KG
1	2718.6	14557	16927	23795
2	3895.3	5122.4	5629	
3	2160.9	655.04	789.21	

Next, the performance of DD-SRPSO and DD-SRPSO-CHM has been evaluated on the problem to study the impact of proposed constraint handling mechanism. Performance of both the algorithms is presented in figure 6.3. From the figure, it can be seen that DD-SRPSO without any constraint handling mechanism is stuck at a point throughout the run whereas the DD-SRPSO-CHM has converged to the optimum payload fraction value.

Further, the performance over limited budget settings has been performed and compared with a well-known Real Coded GA (RCGA) [181, 182]. The population size has been kept 50 for both the algorithms. The algorithms have been tested 25 times and in run the total iteration were set to 20. The convergence graphs for both the functions are presented in figure 6.4. The figure contains the convergence graphs for the best and average performances. It can been that both the algorithms have converged to the optimum payload fraction value but the DD-SRPSO-CHM algorithm have exhibited faster convergence characteristics on the problem.

In this chapter, the performance of SRPSO, DMeSR-PSO and DD-SRPSO has been evaluated on the practical optimization problems. The first study on CEC2011 practical problems clearly indicates that DMeSR-PSO and DD-SRPSO are better performers on the practical real-world problems and among them DD-SRPSO is the best algorithm. Next, new constraint handling mechanism has been proposed for the DD-SRPSO algorithm to solve problems consisting of equality constraints. The performance is then

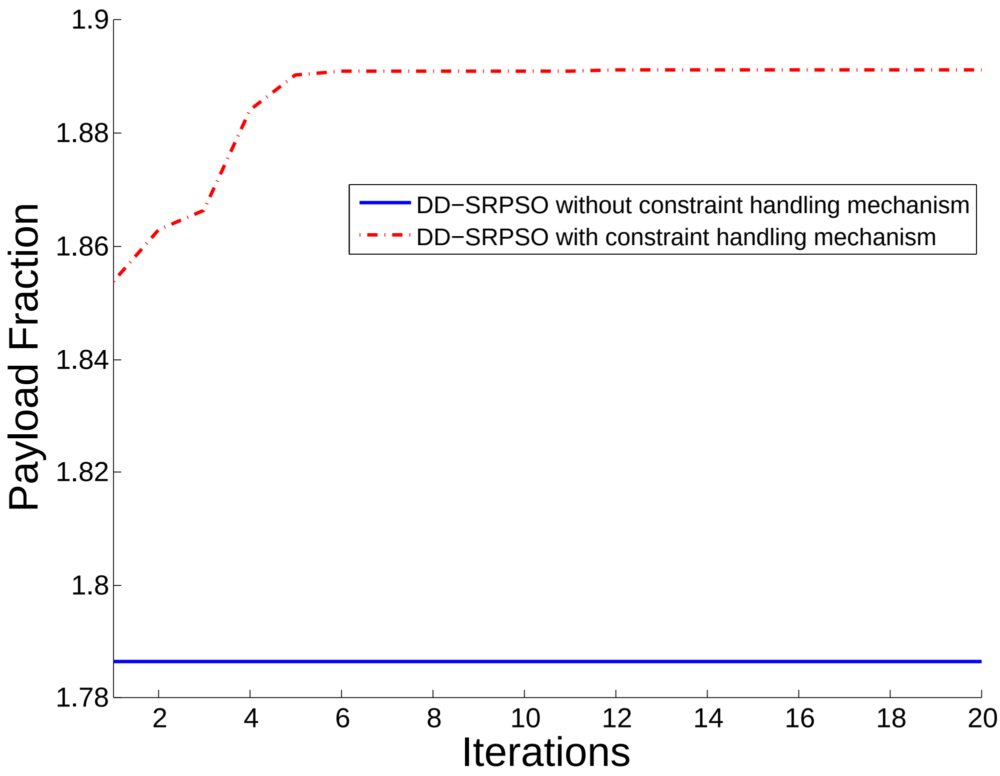

Figure 6.3: Performance of DD-SRPSO on Multi-Stage Launch Vehicle Configuration

tested on optimizing the multi-stage launch vehicle configurations. Using the constraint handling mechanism, the DD-SRPSO-CHM algorithm have successfully

- Provided optimal solutions similar to that proposed in the literature,
- Exhibited superior performances compared to that of DD-SRPSO algorithm and
- Shown faster convergence characteristics compared to the RCGA algorithm.

Therefore, it can be concluded that the proposed human learning principles inspired SRPSO, DMeSR-PSO and DD-SRPSO algorithms have exhibited better performances on the set of benchmark functions and real-world practical problems. Among them, DD-SRPSO have successfully outperformed state-of-the-art PSO variants and other efficient evolutionary algorithms. In the next chapter, the conclusions drawn from this thesis and recommendations for future work are presented.

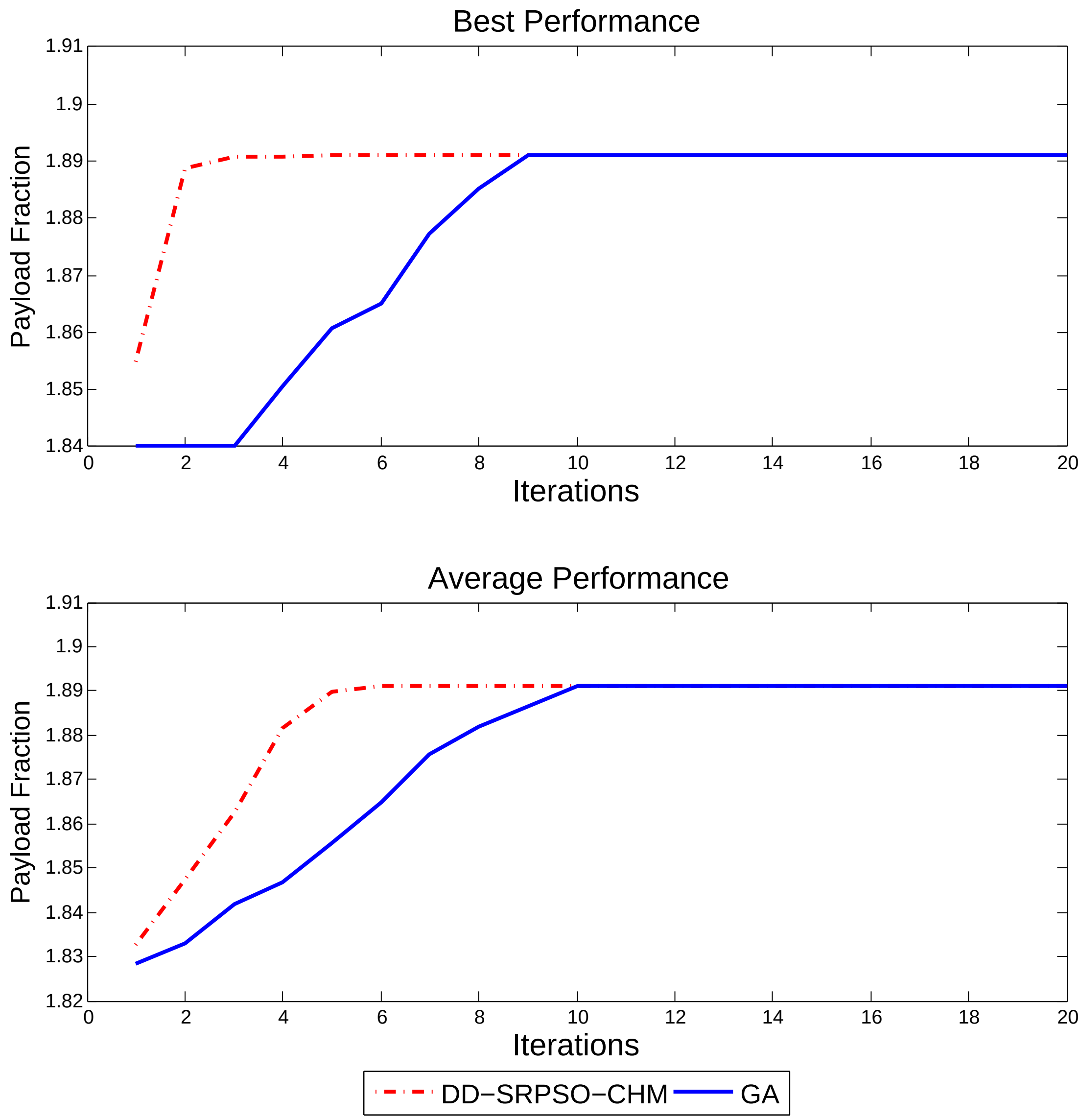

Figure 6.4: Performance comparison of DD-SRPSO-CHM and GA on Multi-Stage Launch Vehicle Configuration

Chapter 7

Conclusions and Future Work

This chapter presents the main conclusions from the studies carried out in this thesis and also presents the recommendations for future work.

7.1 Conclusions

This thesis has focussed on the development of human learning principles inspired particle swarm optimization algorithms. Human like self-regulated learning and socially shared information processing integrated into the PSO framework have introduced effective search for achieving the desired performance. In this thesis, three such human learning principles inspired algorithms, viz., SRPSO, DMeSR-PSO and DD-SRPSO have been developed. Further, a new constraint handling mechanism has been introduced in SRPSO for handling equality constraints. Its performance has been tested on a practical staging design problem of a multi-scale launch vehicle design. In a nut-shell, the major contributions of this thesis are:

(a) Development of human self-learning principles inspired PSO referred to as Self Regulating Particle Swarm Optimization (SRPSO) algorithm.

(b) Development of human socially shared information processing inspired PSO referred to as Dynamic Mentoring and Self-Regulation based Particle Swarm Optimization (DMeSR-PSO) algorithm.

(c) Development of a Directionally Driven Self Regulating Particle Swarm Optimization (DD-SRPSO) algorithm to tackle the rotated characteristics of any optimization problem.

(d) Development of a new constraint handling mechanism in SRPSO and its application to the staging problem of multi-scale launch vehicle design.

The major conclusions from the above studies are:

- **The SRPSO Algorithm:**
  A new self-regulating particle swarm optimization algorithm has been developed based on the human self-learning principles. Two strategies have been introduced in the algorithm, viz., self-regulating inertia weight for the best particle to improve exploration and self-perception in global search direction for the rest of the particles for better exploitation. With the hep of these strategies significant performance improvement has been observed in the SRPSO algorithm. The performance of SRPSO has been studied in comparison with six widely accepted PSO variants using 25 benchmark functions from CEC2005. The results clearly prove that the performance of SRPSO over the set of benchmark functions is much better than others. Finally, the statistical analysis clearly indicates that SRPSO is better than other PSO variants with 95% confidence level.

- **The DMeSR-PSO Algorithm:**
  The socially shared information processing strategies in human beings has been explored and a new dynamic mentoring and self-regulation based particle swarm optimization algorithm has been developed. Here, the particles are divided in mentor, mentee and independent learner groups based on their performances. The algorithm mimics a mentoring based learning environment whereby the learning

schemes have resulted in better exploration and intelligent exploitation of the search space for effective search. As a result, the DMeSR-PSO algorithm has generated optimum/near-optimum solutions with faster convergence characteristics. The performance of DMeSR-PSO has been compared with SRPSO and eight other evolutionary optimization algorithms on the CEC2005 benchmark functions whereby DMeSR-PSO has provided faster convergence closer to the optimum solution with robustness on diverse problems. Finally, the statistical analysis clearly indicates that DMeSR-PSO is better than other evolutionary optimization algorithms with 95% confidence level.

- **The DD-SRPSO Algorithm:**
  The socially guided interaction scheme has been explored together with the rotational invariant design scheme and a new directionally updated and rotationally invariant SRPSO algorithm referred to as a directionally driven self-regulating particle swarm optimization algorithm has been developed. The two new strategies, directional update and rotational invariance characteristics have provided optimum/near-optimum solutions with faster convergence characteristics. These learning strategies have accelerated the convergence because more particles are gathered around the potential solutions. Performance of the DD-SRPSO algorithm has been evaluated on CEC2013 benchmark functions and compared with seven other PSO variants and three evolutionary optimization algorithms. The results clearly highlight that DD-SRPSO has provided better solutions with efficiency and robustness on the set of benchmark problems. Further, the statistical analysis guarantees that DD-SRPSO is statistically better than all the other selected algorithms with a confidence level of 95%. The convergence analysis proves that DD-SRPSO possess faster convergence characteristics closer to the true optimum solution on a wide range of problems compared to SRPSO and DMeSR-PSO.

- **Constraint Handling Technique for DD-SRPSO Algorithm:**
  A DD-SRPSO with constraint handing mechanism (DD-SRPSO-CHM)has been developed. The DD-SRPSO framework has been modified with equality constraint handling mechanism. Based on this modification, the particles are bound to search within the specified region. As a result, the particles converged faster towards the optimum solution. The performance of DD-SRPSO-CHM has been tested in optimizing multi-stage launch vehicle configuration. The constraint handling mechanism has accelerated the convergence process in determining the optimal configurations.

- **Performance Evaluation on Practical Optimization Problems:**
  In this thesis, the three aforementioned algorithms, SRPSO, DMeSR-PSO and DD-SRPSO algorithm has been applied to eight different practical problems form the CEC2011 problems. The performance has been compared with GA Multi-Parent Crossover (GA-MPC) algorithm and Self-Adaptive Multi-Operator DE (SAMODE) algorithm that are proven better performers on the selected problems. From the comparative analysis, it has been observed that DD-SRPSO has achieved either the same solutions or better solutions compared to all other algorithms. Further, DMeSR-PSO has also provided better results compared to SRPSO, GA-MPC and SAMODE on the problems.

To summarize, compared to the existing solutions available in the literature algorithms developed in this thesis have provided better solutions. Performance of SRPSO has been found to be significantly better than other PSO variants. Further, the DMeSR-PSO has shown much improved performance compared to SRPSO as well as other evolutionary optimization algorithms. Finally, equipped with the directional update and rotational invariant strategies, DD-SRPSO has provided faster convergence closer to the optimum solutions in diverse set of benchmark as well as practical optimization problems.

7.2 Future Works

Various extensions can be studied based on the studies conducted in this thesis. A few recommendations for possible research directions are given below:

7.2.1 Incorporating more human learning principles

The thesis addresses only a few of the human learning principles applied to the PSO framework. The human brain is one of the most complex organ and it consists of several interconnected components. Each of these components have specific functionalities associated with them [183]. As indicated in human learning psychology, humans utilize multiple hierarchical inter-related layers of information processing and adopt several strategies for enhancing the decision making abilities for attaining the maximum gain from the environment [1, 129]. Different human abilities and learning principles are discussed in detail in [184]. It would an interesting area to explore the concepts described in [184] for incorporating in the algorithms' framework. Further, the competitive nature of human has not been explored in this thesis for integration in the algorithm. It has been indicated in [185] that humans always have a desire to win that initiates competitive nature in them and leads to innovation and advances. Such a competitive mechanism can be explored for integrating into the algorithm. Similarly, there are other human learning principles that can be integrated in the algorithm.

7.2.2 Extending SRPSO, DMeSR-PSO and DD-SRPSO for multi-objective problems

The algorithms developed in this thesis are designed to solve single objective optimization problems. Many real-world problems are multi-objective whereby there is no single solution for such type of problems as the objectives conflict with each other. The PSO has already been design for solving multi-objective optimization problems [186] which has successfully handled problems with several objective functions [187]. A recent survey on Multi-Objective PSO (MOPSO) [188] provides a comprehensive study about the application areas in aerospace, biological sciences, data mining, industrial engineering etc. where MOPSO has been successfully applied. In this thesis, it has been proven that SRPSO, DMeSR-PSO and DD-SRPSO are fast, efficient, and robust and provide a much better convergence compared to the basic PSO algorithm. Therefore, the proposed algorithms can be extended to solve multi-objective problems.

7.2.3 Incorporating human learning Principles on other population based optimization algorithms

In this thesis, PSO has been chosen as the candidate for exploring human learning principles in an optimization algorithm. It is well-known that there are numerous population based optimization algorithms available in the literature. The Artificial Bee Colony (ABC) algorithm introduced in [15] can be a strong candidate for integration of human learning principles as it has recently proven good for solving multimodal and real-world problems [189]. It will be an interesting area to explore the behaviour of employed bees, onlookers and scouts when equipped with human like self-regulatory mechanism. Further, the concept of mentoring and social guidance can be implemented among the bees so that the onlookers can choose more potential areas by identifying the bees they should follow from the employed bees. In the same way, the scouts can be mentored and guided for selection of good food sources. Similarly, one can explore impact on the performance of other population based optimization algorithms using the human learning principles described in this thesis.

7.2.4 Theoretical analysis on the convergence of SRPSO, DMeSR-PSO and DD-SRPSO

The algorithms developed in this thesis are shown to be faster in convergence when compared to other algorithms through extensive numerical simulation studied on various problems. However, a theoretical analysis on the faster convergence compared to other algorithms will provide some additional insight in understanding the behaviour of the particles during evaluation. Researchers have carried out convergence analysis on convergence speed of the PSO algorithm [190] and most recently a comprehensive analysis on stability, local convergence and transformation sensitivity has been carried out on SPSO2011 [29]. Here, estimate variance convergence boundaries algorithm has been utilized for estimation of the convergence boundaries of SPSO2011 [152]. Similar technique can be implemented for convergence analysis of SRPSO, DMeSR-PSO and DD-SRPSO for better understanding of the faster convergence characteristics of the proposed algorithms.

7.2.5 Handing the inequality constraints

The constraint handling mechanism introduced in the thesis only incorporate the equality constraint in the PSO framework. The associated complexities with the real-world applications are not just limited to the equality constraints. In fact, most of the problems consist of inequality constraints that are highly difficult to handle. In recent past, researchers have tried several methods for handling the inequality constraints in PSO [178]. It will be an interesting problem to investigate the integration of inequality constraint handling mechanism in the structure of the PSO algorithm.

As far as applications are considered, the DD-SRPSO algorithm has shown much promising performance on the practical optimization problems and is definitely a potential candidate for providing effective, optimized solutions to the real-world applications. Thus, one can utilize the DD-SRPSO algorithm for solving complex real-world optimization problems by performing the necessary modifications in the algorithm as per the requirements of the problem.

Publications List

Journals

1. M. R. Tanweer, S. Suresh and N. Sundararajan, "Self Regulating particle swarm optimization algorithm", *Information Sciences*, vol. 294, 2015, pp. 182 - 202.

2. M. R. Tanweer, S. Suresh and N. Sundararajan, "Dynamic Mentoring and Self-Regulation based Particle Swarm Optimization Algorithm for solving Complex Real-world Optimization Problems", *Information Sciences*, vol. 326, 2016, pp. 1 - 24.

3. M.R. Tanweer, R. Auditya, S. Suresh, N. Sundararajan and N. Srikanth, "Directionally Driven Self Regulating particle swarm optimization algorithm", *Swarm and Evolutionary Computation*, vol. 28, 2016, pp. 98 - 116.

Conference Proceedings

1. M. R. Tanweer and S. Suresh, "Human cognition inspired particle swarm optimization algorithm", *IEEE 9th Int. Conf. on Intelligent Sensors, Sensor Networks and Information Processing*, Singapore, pp: 1 - 6, 2014.

2. M. R. Tanweer, S. Suresh, and N. Sundararajan, "Human meta-cognition inspired collaborative search algorithm for optimization", *IEEE Int. Conf. on MFI Integration for Intelligent Systems*, Beijing (China), pp: 1 - 6, 2014.

3. M. R. Tanweer, S. Suresh and N. Sundararajan, “Improved SRPSO algorithm for solving CEC2015 computationally expensive numerical optimization problems”, *IEEE Cong. on Evolutionary Computation*, Sendai (Japan), pp: 1943 - 1949, 2015.

4. M. R. Tanweer, S. Suresh and N. Sundararajan, “Mentoring based Particle Swarm Optimization Algorithm for faster convergence”, *IEEE Cong. on Evolutionary Computation*, Sendai (Japan), pp: 196 - 203, 2015.

5. M. R. Tanweer, Abdullah Al-Dujaili and S. Suresh, “Empirical Assessment of human learning principles inspired particle swarm optimization algorithms on Continuous Black-Box optimization testbed”, *IEEE 6th Int. Conf. on Swarm, Evolutionary and Memtic Computing*, Hyderabad (India), 2015.

6. Abdullah Al-Dujaili, M. R. Tanweer and S. Suresh, “On the performance of Particle Swarm Optimization Algorithms in Solving Cheap Problems”, *IEEE Symposium Series on Computational Intelligence*, Cape Town (South Africa), pp: 1318 - 1325, 2015.

7. Abdullah Al-Dujaili, M. R. Tanweer and S. Suresh, “DE vs. PSO: A Performance Assessment for Expensive Problems”, *IEEE Symposium Series on Computational Intelligence*, Cape Town (South Africa), pp: 1711 - 1718, 2014.

Bibliography

[1] T.O. Nelson and L. Narens. Metamemory: A theoretical framework and new findings. *The psychology of learning and motivation*, 26:125–141, 1990.

[2] P. M. Pardalos and M. GC. Resende. *Handbook of applied optimization.* Oxford university press, 2001.

[3] Z. Beheshti and S. M. H. Shamsuddin. A review of population-based meta-heuristic algorithms. *Int. Jour. of Advanced Soft Computing Appl.*, 5:1 – 35, 2013.

[4] Engelbrecht and P. Andries. *Computational intelligence: An introduction.* John Wiley and Sons, 2007.

[5] C. A. Floudas and P. M. Pardalos. *Encyclopedia of optimization.* Springer Science & Business Media, 2001.

[6] J. H. Holland. *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence.* U Michigan Press, 1975.

[7] K. Price, R. Storn, and J. Lampinen. *Differential evolution: A practical approach to global optimization.* Springer-Verlag, 2005.

[8] N. Hansen and A. Ostermeier. Completely derandomized self adaptation in evolution strategies. *Evolutionary computation*, 9, 2001.

[9] L. Xie, J. Zeng, and Z. Cui. General framework of artificial physics optimization algorithm. In *Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on*, pages 1321–1326, 2009.

[10] H. Shah-Hosseini. Principal components analysis by the galaxy-based search algorithm: a novel metaheuristic for continuous optimisation. *International Journal of Computational Science and Engineering*, 6(1-2):132–140, 2011.

[11] H. Mo, L. Liu, and L. Xu. A power spectrum optimization algorithm inspired by magnetotactic bacteria. *Neural Computing and Applications*, 25(7-8):1823 – 1844.

[12] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi. Gsa: a gravitational search algorithm. *Information sciences*, 179(13):2232–2248, 2009.

[13] J. Kennedy and R.C. Eberhart. Particle swarm optimizatio. In *Proc. of IEEE Int. Conf. on Neural Networks*, pages 1942–1948, 1995.

[14] M. Dorigo, V. Maniezzo, and A. Colorni. The ant system: Optimization by a colony of cooperating agents. *IEEE Trans. on System, Man and Cybernetics, Part B: Cybernetics*, 26(2):29 – 41, 1996.

[15] D. boga and B. Basturk. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. *Jour. of global optimization*, 39(3):459 – 471, 2007.

[16] D. Simon. Biogeography-based optimization. *IEEE Trans. on Evolutionary Computation*, 12(6):702 – 713, 2008.

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

[18] A.D. Belegundu and T.R. Chandrupatla. *Optimization concepts and applications in engineering.* Prentice Hall, 1999.

[19] B. Schmenk. Globalizing learner autonomy. *Tesol Quarterly*, 39(1):107–118, 2005.

[20] C. R. McConnell. Succeeding with succession planning. *The Health Care Manager*, 15(2):69–78, 1996.

[21] Gergely Csibra and György Gergely. Social learning and social cognition: The case for pedagogy. *Processes of change in brain and cognitive development. Attention and performance XXI*, 21:249–274, 2006.

[22] A. Bandura. Social cognitive theory: An agentic perspective. *Annual review of psychology*, 52(1):1–26, 2001.

[23] Y. Song, Z. Chen, and Z. Yuan. New chaotic PSO-based neural network predictive control for nonlinear process. *IEEE Trans. on Neural Networks*, 18(5):595–601, 2007.

[24] A.Banks, J.Vincent, and C.Anyakoha. A review of particle swarm optimization. PartII: Hybridization, combinatorial, multicriteria and constrained optimization and indicative applications. *Natural Computing*, 7:109–124, 2008.

[25] R. Poli. Analysis of the publications on the applications of particle swarm optimization. *Artificial Evolution and Applications*, 28:1–10, 2008.

[26] S. Suresh, P.B. Sujit, and A.K. Rao. Particle swarm optimization approach for multi-objective composite box-beam design. *Composite Structures*, 81(4):598–605, 2007.

[27] R.C. Eberhart and Y. Shi. Particle swarm optimization: developments, applications and resources. In *Proc. of Cong. on Evolutionary Computation*, pages 81–86, 2001.

[28] Y. del Valle, G.K. Venayagamoorthy, S. Mohagheghi, J.C. Hernandez, and R.G. Harley. Particle swarm optimization: basic concepts, variants and applications in power systems. *IEEE Trans. on Evolutionary Computation*, 12(2):171–195, 2008.

[29] M. R. Bonyadi and Z. Michalewicz. Analysis of stability, local convergence, and transformation sensitivity of a variant of the particle swarm optimization algorithm. *IEEE Transactions on Evolutionary Computation*, 20(3):370–385, 2016.

[30] Y. Shi and R.C. Eberhart. A modified particle swarm optimizer. In *Proc. of WCCI*, pages 69–73, 1998.

[31] M. Clerc and J. Kennedy. The particle swarm - Explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans. on Evolutionary Computation*, 6(1):58–73, 2002.

[32] R. Mendes, J. Kennedy, and J. Neves. The fully informed particle swarm: simpler, maybe better. *IEEE Trans. on Evolutionary Computation*, 8(3):204–210, 2004.

[33] K.E. Parsopoulos and M.N. Vrahatis. Empirical study of particle swarm optimizer with an increasing inertia weight. In *Proc. of Cong. on Evolutionary Computation*, pages 868–873, 2004.

[34] A. Ratnaweera, S.K. Halgamuge, and H.C. Watson. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *IEEE Trans. on Evolutionary Computation*, 8(3):240–255, 2004.

[35] J.J. Liang, A.K. Qin, P.N. Suganthan, and S. Baskar. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Trans. on Evolutionary Computation*, 10(3):281–295, 2006.

[36] H. Wang, H. Sun, C. Li, S. Rahnamayan, and J.S. Pan. Diversity enhanced particle swarm optimization with neighborhood search. *Information Sciences*, 223:119–135, 2013.

[37] W.H. Lim and N.A.M. Isa. Teaching and peer-learning particle swarm optimization. *Applied Soft Computing*, 18:39–58, 2014.

[38] L. Wang, B.Yang, and Y. Chen. Improving particle swarm optimization using multi-layer searching strategy. *Information Sciences*, 274:70–94, 2014.

[39] Z. Beheshti and S. M. Hj Shamsuddin. CAPSO: centripetal accelerated particle swarm optimization. *Information Sciences*, 258:54–79, 2014.

[40] R. Cheng and Y. Jin. A social learning particle swarm optimization algorithm for scalable optimization. *Information Sciences*, 291:43–60, 2015.

[41] S. L. Ho, J. Yang, S. Yang, and Y. Bai. Integration of directed searches in particle swarm optimization for multi-objective optimization. *IEEE Transactions on Magnetics*, 51(3):1–4, 2015.

[42] Y. Gao, W. Du, and G. Yan. Selectively-informed particle swarm optimization. *Scientific reports*, 5:1–7, 2015.

[43] S. U. Khan, S. Yang, L. Wang, and L. Liu. A modified particle swarm optimization algorithm for global optimizations of inverse problems. *IEEE Transactions on Magnetics*, 52(3):1–4, 2016.

[44] Qiaohe Yang, Jinpeng Tian, and Wen Si. An improved particle swarm optimization based on difference equation analysis. *Journal of Difference Equations and Applications*, pages 1–18, 2016.

[45] Y. F. Zhang and H. D. Chiang. A novel consensus-based particle swarm optimization-assisted trust-tech methodology for large-scale global optimization. *IEEE Transactions on Cybernetics*, pages 1–16, 2016.

[46] S. J. Isakowitz. International reference guide to space launch systems. *NASA STI/Recon Technical Report A*, 91:51425, 1991.

[47] S. Suresh, H. J. Rong, and N. Sundararajan. Bio-inspired computing for launch vehicle design and trajectory optimization. In *IEEE Symposium on Computational Intelligence and Security and Defence Applicaations*, pages 1–6, 2009.

[48] Y J. Zheng, H. F. Ling, J. Y. Xue, and S. Y. Chen. Population classification in fire evacuation: a multiobjective particle swarm optimization approach. *IEEE Transactions on Evolutionary Computation*, 18(1):70–81, 2014.

[49] G. Sermpinis, K. Theofilatos, A. Karathanasopoulos, E. F. Georgopoulos, and C. Dunis. Forecasting foreign exchange rates with adaptive neural networks using radial-basis functions and particle swarm optimization. *European Journal of Operational Research*, 225(3):528–540, 2013.

[50] M. C. Chen, Y. H. Hsiao, R. H. Reddy, and M. K. Tiwari. The self-learning particle swarm optimization approach for routing pickup and delivery of multiple products with material handling in multiple cross-docks. *Transportation Research Part E: Logistics and Transportation Review*, 91:208–226, 2016.

[51] G. Liu, Z. Chen, H. W. F. Yeung, Y. Y. Chung, and W.C. Yeh. A new weight adjusted particle swarm optimization for real-time multiple object tracking. In *International Conference on Neural Information Processing*, pages 643–651, 2016.

[52] P. Patel and M. Patel. Tumor detection using particle swarm optimization to initialize fuzzy c-means. *International Journal for Scientific Research & Development*, 4(3):255–259, 2016.

[53] S. Jiang, Z. Ji, and Y. Wang. A novel gravitational acceleration enhanced particle swarm optimization algorithm for wind–thermal economic emission dispatch problem considering wind power availability. *International Journal of Electrical Power & Energy Systems*, 73:1035–1050, 2015.

[54] A. Modiri, X. Gu, A. Hagan, and A. Sawan. Radiotherapy planning using an improved search strategy in particle swarm optimization. *IEEE Transactions on Biomedical Engineering*, pages 1–16, 2016.

[55] P. Li, D. Xu, Z. Zhou, W. J. Lee, and B. Zhao. Stochastic optimal operation of microgrid based on chaotic binary particle swarm optimization. *IEEE Transactions on Smart Grid*, 7(1):66–73, 2016.

[56] B. Ufnalski, A. Kaszewski, and L. M. Grzesiak. Particle swarm optimization of the multioscillatory lqr for a three-phase four-wire voltage-source inverter with an output filter. *IEEE Transactions on Industrial Electronics*, 62(1):484–493, 2015.

[57] R. Koad, A. F. Zobaa, and A. El Shahat. A novel mppt algorithm based on particle swarm optimisation for photovoltaic systems. *IEEE Transactions on Sustainable Energy*, pages 1–16, 2016.

[58] W. Han, T. T. Tran, J. W. Kim, Y. J. Kim, and S. Y. Jung. Mass ionized particle optimization algorithm applied to optimal fea-based design of electric machine. *IEEE Transactions on Magnetics*, 52(3):1–4, 2016.

[59] P.N. Suganthan, N. Hansen, J.J. Liang, Y.P. Chen K. Deb, A. Auger, and S. Tiwari. Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. *Technical Report: Nanyang Technological University AND KanGAL Report*, (2005005), 2005.

[60] Y. Shi and R.C. Eberhart. Empirical study of particle swarm optimization. In *Proc. of Cong. on Evolutionary Computation*, pages 1945–1950, 1999.

[61] J. Kennedy. Small worlds and mega-minds: Effects of neighborhood topology on particle swarm performance. In *Proc. of Cong. on Evolutionary Computation*, 1999.

[62] J. Kennedy and R. Mendes. Population structure and particle swarm performance. In *Proc. of Cong. on Evolutionary Computation*, pages 1671–1676, 2002.

[63] Y. L. Zheng, L. H. Ma, L. Y. Zhang, and J. X. Qian. A unified particle swarm optimization scheme. In *Proc. of the IEEE Int. Conf. of Computational Methods in Sciences and Engineering*, pages 221–226, 2003.

[64] W. A. Hassan, M. B. Fayek, and S. I. Shaheen. PSOSA: An optimised particle swarm technique for solving the urban planning problem. In *Proc. of the IEEE Int. Conf. of Computer Engineering and systems*, pages 401–405, 2006.

[65] M. Pant, T. Radha, and V. P. Singh. Particle swarm optimization using Gaussian inertia weight. In *Proc. of the IEEE Int. Conf. on Computataional intelligence and multimedia*, pages 97–102, 2007.

[66] G. Y Lin and D. Y. Hong. A new particle swarm optimization algorithm with random inertia weight and evolution strategy. In *Proc. of the IEEE Int. Conf. on CIS*, pages 199–203, 2007.

[67] L. Li, B. Xue, B. Niu, Y. Chai, and J. Wu. The novel nonlinear strategy of inertia weight in particle swarm optimization. In *Proc. of the IEEE Int.Conf. on Bio-inspired computation*, pages 1–5, 2009.

[68] W. Jianxin, H.X. Xin, Z. Weiguo, and W. Rui. Exponential inertia weight particle swarm algorithm for dynamic optimization of electromechanical coupling system. In *Proc. of the IEEE Int. Conf. on Intelligent computing and intelligent systems*, pages 479–483, 2009.

[69] L. Zhang, Y. Tang, C. Hua, and X. Guan. A new particle swarm optimization algorithm with adaptive inertia weight based on bayesian techniques. *Applied Soft Computing*, 28:138–149, 2015.

[70] M. Taherkhani and R. Safabakhsh. A novel stability-based adaptive inertia weight for particle swarm optimization. *Applied Soft Computing*, 38:281–295, 2016.

[71] P. Bajpai and S.N. Singh. Fuzzy adaptive particle swarm optimization for bidding strategy in uniform price spot market. *IEEE Trans. on Power systems*, 22(4):2152–2160, 2007.

[72] C. Liu, C. Ouyang, P. Zhu, and W. Tang. An adaptive fuzzy weight PSO algorithm. In *Proc. of the IEEE Conf. on Genetic and Evolutionary computation*, pages 8–10, 2010.

[73] W. Han, P. Yang, H. Ren, and J. Sun. Comparison study of several kinds of inertia weight for PSO. In *Proc. of the IEEE Int. Conf. on progress in informatics and computing*, pages 280–284, 2010.

[74] E. Ozcan and C.Mohan. Particle swarm optimization: Surfing the waves. In *Proc. of Cong. on Evolutionary Computation*, pages 1939–1944, 1999.

[75] G.Q. Bao and K.F. Mao. Particle swarm optimization algorithm with asymmetric time-varying acceleration coefficients. In *Proc. of IEEE Int. Conf. on robotics and biomimetics*, pages 2134–2139, 2009.

[76] L. Guo and X.Chen. A novel particle swarm optimization based on the self-adaptation strategy of acceleration coefficients. In *Proc. of the IEEE Int. Conf. on Computational intelligence and security*, pages 277–281, 2009.

[77] T. Ziyu and Z. Dingxue. A modified particle swarm optimization with adaptive acceleration coefficients. In *Proc. of the IEEE Int. Conf. on Information processing*, pages 330–332, 2009.

[78] Z.H. Zhan, J. Zhang, Y. Li, and H.S.H. Chung. Adaptive particle swarm optimization. *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, 39(6):1362–1381, 2009.

[79] B. Alatas, E. Akin, and A.B. Ozer. Chaos embedded particle swarm optimization algorithms. *Chaos, Solitons and Fractals*, 40(4):1715–1734, 2009.

[80] M.S. Leu and M.F. Yeh. Grey particle swarm optimization. *Applied Soft Computing*, 12(9):2985–2996, 2012.

[81] J. Ding, J. Liu, K.R. Chowdhury, W. Zhang, Q. Hu, and J. Lei. A particle swarm optimization using local stochastic search and enhancing diversity for continuous optimization. *Neurocomputing*, 137:261–267, 2014.

[82] F.V.D. Bergh, Engelbrecht, and P. Andries. A cooperative approach to particle swarm optimization. *IEEE Trans. on Evolutionary Computation*, 8(3):225–239, 2004.

[83] S. Janson and M. Middendorf. A hierarchical particle swarm optimizer and its adaptive variant. *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, 35(6):1272–1282, 2005.

[84] D. Parrott and X. Li. Locating and tracking multiple dynamic optima by a particle swarm model using speciation. *IEEE Trans. on Evolutionary Computation*, 10(4):440–458, 2006.

[85] H. Wang, Z. Wu, S. Rahnamayan, C. Li, S. Zeng, and D. Jiangand. Particle swarm optimization with simple and efficient neighbourhood search strategies. *International Journal of Innovative Computing and Applications*, 3(2):97–104, 2011.

[86] X. Xu, Y. Tang, J. Li, C. Hua, and X. Guan. Dynamic multi-swarm particle swarm optimizer with cooperative learning strategy. *Applied Soft Computing*, 29:169–183, 2015.

[87] J. Rada-Vilela, M. Johnston, and M. Zhang. Population statistics for particle swarm optimization: Hybrid methods in noisy optimization problems. *Swarm and Evolutionary Computation*, 22:15–29, 2015.

[88] J. Park and K-Y. Kim. Instance variant nearest neighbor using particle swarm optimization for function approximation. *Applied Soft Computing*, 40:331–341, 2016.

[89] J.J. Liang and P.N. Suganthan. Dynamic multi-swarm particle swarm optimizer. In *Proc. of IEEE SIS*, pages 210–224, 2005.

[90] P.J. Angeline. Using selection to improve particle swarm optimization. In *Proc. of the IEEE Int. Conf. on Evolutionary computation*, pages 84–89, 2010.

[91] M. Lovbjerg, T. K. Rasmussen, and T. Krink. Hybrid particle swarm optimizer with breeding and subpopulations. In *Proc. of the IEEE Int. Conf. on Genetic and Evolutionary computation*, pages 469–476, 2001.

[92] N. Higashi and H. Iba. Particle swarm optimization with Gaussian mutation. In *Proc. of IEEE SIS*, pages 72–79, 2003.

[93] W.J. Zhang and X.F. Xie. DEPSO: hybrid particle swarm with differential evolution operator. In *IEEE Int. Conf. Systems, Man, and Cybernetics*, pages 3816–3821, 2003.

[94] M.G. Epitropakis, V.P. Plagianakos, and M.N. Vrahatis. Evolving cognitive and social experience in particle swarm optimization through differential evolution. In *Proc. of IEEE Cong. on Evolutionary Computation*, pages 1–8, 2010.

[95] M.G. Epitropakis, V.P. Plagianakos, and M.N. Vrahatis. Evolving cognitive and social experience in Particle Swarm Optimization through Differential Evolution: A hybrid approach. *Information Sciences*, 216(1):50–92, 2012.

[96] B. Xin, J. Chen, J. Zhang, H. Fang, and Z.H. Peng. Hybridizing differential evolution and particle swarm optimization to design powerful optimizers: a review and taxonomy. *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, 42(5):744–767, 2012.

[97] J. Sun, B.W. Xu, and W. Fang. Diversity-guided quantum-behaved particle swarm optimization algorithm. In *Proc. of Int. Conf. on Simulated Evolution and Learning*, pages 497–504, 2006.

[98] M. Lovbjerg and T. Krink. Extending particle swarm optimizers with self-organized criticality. In *Proc. of IEEE Cong. Evolutionary computation*, pages 1588–1593, 2002.

[99] X. Xie, W. Zhang, and Z. Yang. A dissipative particle swarm optimization. In *Proc. of IEEE Cong. Evolutionary computation*, pages 1456–1461, 2002.

[100] K.E. Parsopoulos and M.N. Vrahatis. On the computation of all global minimizers through particle swarm optimization. *IEEE Trans. on Evolutionary Computation*, 8(3):211–224, 2004.

[101] G-G. Wang, A. H. Gandomi, A. H. Alavi, and S. Deb. A hybrid method based on krill herd and quantum-behaved particle swarm optimization. *Neural Computing and Applications*, 27(4):989–1006, 2016.

[102] M.J. Mahmoodabadi, Z.M. Salahshoor, and A. Bagheri. HEPSO: High exploration particle swarm optimization. *Information Sciences*, 273:101–111, 2014.

[103] R. Thangaraj, M. Pant, A. Abraham, and P. Bouvry. Particle swarm optimization: Hybridization perspectives and experimental illustrations. *Applied Mathematics and Computation*, 217(12):5208–5226, 2011.

[104] M. Eslami, H. Shareef, M. Khajehzadeh, and A. Mohamed. A survey of the state of the art in particle swarm optimization. *Research Jour. of Applied Sciences, Engineering and Technology*, 4(9):1181–1197, 2012.

[105] T. Peram, K. Veeramachaneni, and C.K. Mohan. Fitness-distance-ratio based particle swarm optimization. In *Proc. of IEEE SIS*, pages 174–181, 2003.

[106] C. Li, S. Yang, and T.T. Nguyen. A self-learning particle swarm optimizer for global optimization problems. *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, 42(3):627–646, 2012.

[107] K. Tang, Z. Li, L. Luo, and B. liu. Multi-strategy adaptive particle swarm optimization for numerical optimization. *Engineering Applications of Artificial Intelligence*, 37:9 – 19, 2015.

[108] H.M. Wang, Z.W. Qiao, C.L. Xia, and L.Y. Li. Self-regulating and self-evolving particle swarm optimizer. *Engineering Optimization*, 47(1):129–147, 2015.

[109] Y. Li, Z.H. Zhan, S. Lin, J. Zhang, and X. Luo. Competitive and cooperative particle swarm optimization with information sharing mechanism for global optimization problems. *Information Sciences*, 293:370–382, 2015.

[110] C. Zeng. A particle swarm optimization algorithm with rich social cognition. In *IEEE 5th International Conference on natural computation*, pages 186–189, 2009.

[111] Z. Zhi-Hui, Z. Jun, L. Yun, and S. Yu-Hui. Orthogonal learning particle swarm optimization. *IEEE Trans. on Evolutionary Computation*, 15(6):832–847, 2011.

[112] Q. Qin, S. Cheng, Q. Zhang, Y. Wei, and Y. Shi. Multiple strategies based orthogonal design particle swarm optimizer for numerical optimization. *Computers & Operations Research*, 60:91–110, 2015.

[113] W.N. Chen, J. Zhang, Y. Lin, N. Chen, Z.H. Zhan, H.S.H. Chung, Y. Li, and Y.H. Shi. Particle swarm optimization with an aging leader and challengers. *IEEE Trans. on Evolutionary Computation*, 17(2):241–258, 2013.

[114] B.Y. Qu, P.N. Suganthan, and S. Das. Distance-based locally informed particle swarm model for multimodal optimization. *IEEE Trans. on Evolutionary Computation*, 17(3):387–402, 2013.

[115] Y. Shi, H. Liu, L. Gao, and G. Zhang. Cellular particle swarm optimization. *Information Sciences*, 181(20):4460–4493, 2013.

[116] X. Sheng, M. Xi, J. Sun, and W. Xu. Quantum-behaved particle swarm optimization with novel adaptive strategies. *Journal of Algorithms & Computational Technology*, 9(2):143–162, 2015.

[117] C. Sun, J. Zeng, J. Pan, S. Xue, and Y.Jin. A new fitness estimation strategy for particle swarm optimization. *Information Sciences*, 221:355–370, 2013.

[118] J. Kennedy. Bare bones particle swarms. In *Proc. of IEEE SIS*, pages 80–87, 2003.

[119] K.E. Parsopoulos and M.N. Vrahatis. UPSO: A unified particle swarm optimization scheme. In *Lecture Series on Computer and Computational Sciences: Proc. of Int. Conf. of Computational Methods in Sciences and Engineering*, pages 868–873, 2004.

[120] W. Wang, H. Wang, and S. Rahnamayan. Improving comprehensive learning particle swarm optimiser using generalized opposition-based learning. *Int. Journ. of Modelling, Identification and Control*, 14(4):310–316, 2011.

[121] S.K. Goudos, V. Moysiadou, T. Samaras, K. Siakavara, and J.N. Sahalos. Application of a comprehensive learning particle swarm optimizer to unequally spaced linear array synthesis with sidelobe level suppression and null control. *IEEE Antennas and Wireless Propagation Letters*, 9:125–129, 2010.

[122] M. R. Bonyadi and Z. Michalewicz. Particle swarm optimization for single objective continuous space problems: a review. *Evolutionary computation*, 2016.

[123] J. P. Gollub, M. W. Bertenthal, J. B. Labov, and P. C. Curtis. *Learning and Understanding: Improving Advanced Study of Mathematics and Science in US High Schools*. National Acedemic Press, 2002.

[124] H. S. Terrace. *The Missing Link in Cognition: Origins of Self-Reflective Consciousness*. Oxford University Press, USA, 2004.

[125] X. Yang and S. Liu. Dynamic adjustment strategies of inertia weight in particle swarm optimization algorithm. *Int. Jour. of Control and Automation*, 7(5):353 – 364, 2014.

[126] R. S. Wyer and T. K. Srull. Human cognition in its social context. *Psychological review*, 93(3):322, 1986.

[127] A. Bandura. *Social foundations of thought and action: A social cognitive theory*. Prentice-Hall, Inc., 1986.

[128] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *Machine Learning Research*, 7:1–30, 2006.

[129] T. Iiskala, M. Vauras, E. Lehtinen, and P. K. Salonen. Socially shared metacognition in dyads of pupils in collaborative mathematical problem-solving processes. *Learning and Instruction*, 21(1):379 – 393, 2011.

[130] S. Pinker. The cognitive niche: Coevolution of intelligence, sociality, and language. In *Proc. of the National Academy of Sciences, 107(Supplement 2)*, pages 8993–8999, 2010.

[131] R. Boyd, P. J. Richerson, and J. Henrich. The cultural niche: Why social learning is essential for human adaptation. In *Proc. of the National Academy of Sciences, 108(Supplement 2)*, pages 10918–10925, 2011.

[132] R. C. Zwart, T. Wubbels, T. C. Bergen, and S. Bolhuis. Experienced teacher learning within the context of reciprocal peer coaching. *Teachers and Teaching: theory and practice*, 13(2):165–187, 2007.

[133] G. G. McGee, M. C. Almeida, B. Sulzer-Azaroff, and R. S. Feldman. Promoting reciprocal interactions via peer incidental teaching. *Journal of applied behavior analysis*, 25(1):117–126, 1992.

[134] S. L. Odom and P. S. Strain. A comparison of peer-initiation and teacher-antecedent interventions for promoting reciprocal social interaction of autistic preschoolers. *Journal of applied behavior analysis*, 19(1):59–71, 1986.

[135] F. W. Kohler, H. K. Ezell, and M. Paluselli. Promoting changes in teachers' conduct of student pair activities an examination of reciprocal peer coaching. *The journal of special education*, 33(3):154–165, 1999.

[136] H. Law. *The psychology of coaching, mentoring and learning.* John Wiley and Sons, 2013.

[137] B. Bozeman and M. K. Feeney. Toward a useful theory of mentoring a conceptual analysis and critique. *Administration & Society*, 39(6):719–739, 2007.

[138] L. A. Daloz. *Mentor: Guiding the Journey of Adult Learners. The Jossey-Bass Higher and Adult Education Series.* Jossey-Bass Publishers, 350 Sansome Street, San Francisco, CA 94104, 1999.

[139] C.C. Mincemoyer and J.S. Thomson. Establishing effective mentoring relationships for individual and organizational success. *Jour. of Extension*, 36(2):1–8, 1998.

[140] M. R. Tanweer and S. Suresh amd N. Sundararajan. Self regulating particle swarm optimization algorithm. *Information Sciences*, 294:182 – 202, 2014.

[141] R. Storn and K. Price. Differential evolution-A simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341 – 359, 1997.

[142] J. Ronkkonen, S. Kukkonen, and K. V. Price. Real-parameter optimization with differential evolution. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 506–513, 2005.

[143] L. T. Bui, Y. Shan, F. Qi, and H. A. Abbass. Comparing two versions of differential evolution in real parameter optimization. In *IEEE Congress on Evolutionary Computation (CEC)*, 2005.

[144] S. Alonso et al. Performance of a flexible evolutionary algorithm. In *IEEE Congress on Evolutionary Computation (CEC)*, 2005.

[145] A. Sinha, S. Tiwari, and K. Deb. A population-based, steady-state procedure for real-parameter optimization. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 514–521, 2005.

[146] B. Yuan and M. Gallagher. Experimental results for the special session on real-parameter optimization at cec 2005: a simple, continuous eda. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 1792–1799, 2005.

[147] C. García-Martínez and M. Lozano. Hybrid real-coded genetic algorithms with female and male differentiation. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 896–903, 2005.

[148] D. Molina, F. Herrera, and M. Lozano. Adaptive local search parameters for real-coded memetic algorithms. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 888–895, 2005.

[149] A. Auger and N. Hansen. A restart cma evolution strategy with increasing population size. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 1769–1776, 2005.

[150] R. V. Rao and V. J. Savsani. *Mechanical design optimization using advanced optimization techniques*. London: Springer, 2012.

[151] W.M. Spears, D. Green, and D.F. Spears. *Innovations and Developments of Swarm Intelligence Applications*, chapter Biases in particle swarm optimization. 2010.

[152] M. Zambrano-Bigiarini, M. Clerc, and R. Rojas. Standard particle swarm optimisation 2011 at CEC-2013: A baseline for future PSO improvements. In *IEEE Cong. on Evolutionary Computation (CEC)*, pages 2337 – 2344, 2011.

[153] J.J. Liang, Y.B. Qu, P.N. Suganthan, and A.G. Hernández-Díaz. Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization. *Technical Report: Zhengzhou University and Nanyang Technological University*, (201212), 2005.

[154] M. Clerc. The particle swarm central optimisation, standard particle swarm.

[155] F. Caraffini, G. Iacca, F. Neri, L. Picinali, and E. Mininno. A cma-es super-fit scheme for the re-sampled inheritance search. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 1123–1130, 2013.

[156] A. K. Qin and P. N. Suganthan. Self-adaptive differential evolution algorithm for numerical optimization. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 1785–1791, 2005.

[157] J. Brest, B. Boskovic, A. Zamuda, I. Fister, and E. Mezura-Montes. Real parameter single objective optimization using self-adaptive differential evolution algorithm with more strategies. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 377–383, 2013.

[158] S. M. Elsayed, R. A. Sarker, and D. L. Essam. A genetic algorithm for solving the cec2013 competition problems on real-parameter optimization. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 356–360, 2013.

[159] M. R. Tanweer, S. Suresh, and N. Sundararajan. Dynamic mentoring and self-regulation based particle swarm optimization algorithm for solving complex real-world optimization problems. *Information Sciences*, 326:1–24, 2016.

[160] S. M. Elsayed, R. A. Sarker, and D. L. Essam. GA with a new multi-parent crossover for solving ieee-cec2011 competition problems. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 1034 – 1040, 2011.

[161] S. M. Elsayed, R. A. Sarker, and D. L. Essam. Differential evolution with multiple strategies for solving CEC2011 real-world numerical optimization problems. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 1041 – 1048, 2011.

[162] S. Das and P.N. Suganthan. Problem definitions and evaluation criteria for the CEC 2011 competition on testing evolutionary algorithms on real world optimization problems. *Technical Report: Jadavpur University, India AND Nanyang Technological University, Singapore*, 2010.

[163] C.A. Floudas et al. *Handbook of test problems in local and global optimization.* Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996.

[164] A.M. Perez-Bellido, S. Salcedo-Sanz, E.G. Ortiz-Garcia, and A. Portilla-Figueras. A hybrid evolutionary programming algorithm for spread spectrum radar polyphase codes design. In *Proc. of the 9th annual conf. on Genetic and evolutionary computation*, pages 682–688, 2007.

[165] I. De J. Silva, M.J. Rider, R. Romero, A.V. Garcia, and C.A. Murari. Transmission network expansion planning with security constraints. *IEEE Proc.-Generation, Transmission and Distribution*, 152(6):828–836, 2005.

[166] E. KP. Chong and S. H. Zak. *An introduction to optimization*, volume 76. John Wiley & Sons, 2013.

[167] S .Singiresu Rao and SS. Rao. *Engineering optimization: theory and practice*. John Wiley & Sons, 2009.

[168] T. Bäck, D. B. Fogel, and Z. Michalewicz. Handbook of evolutionary computation. *Release*, 97:B1, 1997.

[169] Z. Michalewicz and G. Nazhiyath. Genocop iii: A co-evolutionary algorithm for numerical optimization problems with nonlinear constraints. In *Evolutionary Computation, 1995., IEEE International Conference on*, volume 2, pages 647–651, 1995.

[170] P .D. Surry, N. J. Radcliffe, and I. D. Boyd. A multi-objective approach to constrained optimisation of gas supply networks: The COMOGA method. In *AISB Workshop on Evolutionary Computing*, pages 166–180, 1995.

[171] T. P. Runarsson and X. Yao. Stochastic ranking for constrained evolutionary optimization. *IEEE Transactions on evolutionary computation*, 4(3):284–294, 2000.

[172] G. Zhang, J. Cheng, M. Gheorghe, and Q. Meng. A hybrid approach based on differential evolution and tissue membrane systems for solving constrained manufacturing parameter optimization problems. *Applied Soft Computing*, 13(3):1528–1542, 2013.

[173] J. Zheng, Q. Wu, and W. Song. An improved particle swarm algorithm for solving nonlinear constrained optimization problems. In *Third international conference on natural computation (ICNC 2007)*, volume 4, pages 112–117, 2007.

[174] Shanhe Jiang, Zhicheng Ji, and Yanxia Shen. A novel hybrid particle swarm optimization and gravitational search algorithm for solving economic emission load dispatch problems with various practical constraints. *International Journal of Electrical Power & Energy Systems*, 55:628–644, 2014.

[175] M. Bonyadi, X. Li, and Z. Michalewicz. A hybrid particle swarm with velocity mutation for constraint optimization problems. In *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, pages 1–8, 2013.

[176] É. D. C. R. Carvalho, J. P. G. Carvalho, H. S. Bernardino, P. H. Hallak, and A. C. D. C. Lemonge. An adaptive constraint handlig tecnique for particle swarm in constrained optimization problems. *Revista CIATEC-UPF*, 8:39–56, 2016.

[177] A. H. Gandomi and A. R. Kashani. Evolutionary bound constraint handling for particle swarm optimization. In *Computational and Business Intelligence (ISCBI), 2016 4th International Symposium on*, pages 148–152, 2016.

[178] A. R. Jordehi. A review on constraint handling strategies in particle swarm optimisation. *Neural Computing and Applications*, 26(6):1265–1275, 2015.

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

[180] W. J. Larson and J. R. Wertz. Space mission analysis and design. Technical report, Microcosm, Inc., Torrance, CA (US), 1992.

[181] V Mani, Sundaram Suresh, and Hyong Joong Kim. Real-coded genetic algorithms for optimal static load balancing in distributed computing system with communication delays. In *International Conference on Computational Science and Its Applications*, pages 269–279, 2005.

[182] M. Thakur, S.S. Meghwani, and H. Jalota. A modified real coded genetic algorithm for constrained optimization. *Applied Mathematics and Computation*, 235:292–317, 2014.

[183] A. R. Luria. *The working brain: An introduction to neuropsychology.* Basic Books, 1976.

[184] M. JA. Howe. *Principles of abilities and human learning.* Psychology Press, 1998.

[185] D. Malhotra. The desire to win: The effects of competitive arousal on motivation and behavior. *Organizational Behavior and Human Decision Processes*, 111(2):139–146, 2010.

[186] CAC. Coello and M. S. Lechuga. MOPSO: A proposal for multiple objective particle swarm optimization. In *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 IEEE Congress on*, volume 2, pages 1051–1056, 2002.

[187] CAC. Coello, G. T. Pulido, and M. S. Lechuga. Handling multiple objectives with particle swarm optimization. *IEEE Transactions on evolutionary computation*, 8(3):256–279, 2004.

[188] S. Lalwani, S. Singhal, R. Kumar, and N. Gupta. A comprehensive survey: Applications of multi-objective particle swarm optimization (mopso) algorithm. *Transactions on Combinatorics*, 2(1):39–101, 2013.

[189] S .Wang, Y. Zhang, G. Ji, J. Yang, J. Wu, and L. Wei. Fruit classification by wavelet-entropy and feedforward neural network trained by fitness-scaled chaotic ABC and biogeography-based optimization. *Entropy*, 17(8):5711–5728, 2015.

[190] I. C. Trelea. The particle swarm optimization algorithm: convergence analysis and parameter selection. *Information processing letters*, 85(6):317–325, 2003.

[191] Q. Chen, B. Liu, Q. Zhang, J. J. Liang, P. N. Suganthan, and B. Y. Qu. Problem definition and evaluation criteria for CEC 2015 special session and competition on bound constrained single-objective computationally expensive numerical optimization. *Technical Report*, 2014.

[192] M. Arya, K. Deep, and J. C. Bansal. A nature inspired adaptive inertia weight in particle swarm optimization. *International Journal of AI and Soft Computing*, 4(2-3):228–248, 2014.

[193] H. Huang, H. Qin, Z. Hao, and A. Lim. Example-based learning particle swarm optimization for continuous optimization. *Information Sciences*, 182(1):125–138, 2012.

[194] R. Cheng and Y. Jini. A competitive swarm optimizer for large scale optimization. *Cybernetics, IEEE Transactions on*, 45(2):191–204, 2015.

[195] R. Sawant, D. Hatton, and D.P. O'Donoghue. An ancestor based extension to differential evolution (AncDE) for single-objective computationally expensive numerical optimization. In *IEEE Cong. on Evolutionary Computation (CEC)*, pages 3228 – 3234, 2015.

[196] I. Erlich, G. K. Venayagamoorthy, and W. Nakawiro. A mean-variance optimization algorithm. In *IEEE Cong. on Evolutionary Computation (CEC)*, pages 1 – 6, 2010.

[197] J.L. Rueda and I. Erlich. MVMO for bound constrained single-objective computationally expensive numerical optimization. In *IEEE Cong. on Evolutionary Computation (CEC)*, pages 1011 – 1017, 2015.

[198] V. Berthier. Experiments on the CEC 2015 expensive optimization testbed. In *IEEE Cong. on Evolutionary Computation (CEC)*, pages 1059 – 1066, 2015.

Appendix

A CEC2005 Benchmark Functions

To evaluate different kinds of optimization algorithms in a more systematic manner by specifying a common termination criterion, size of problems, initialization scheme, linkages/rotation, etc. a set to 25 test benchmark function has been introduced [59]. These problems are categorized as Unimodal (F_1 - F_5), Basic Multimodal (F_6 - F_{12}), Expanded Multimodal (F_{13} and F_{14}) and Hybrid Composition (F_{15} - F_{25}). A summary of these problems is given below:

Name	Objective Function	Search Range	Bias(f_b)
Shifted Sphere	$F_1=\sum_{i=1}^{D} z_i^2 + f_b$	$[-100,100]^D$	-450
Shifted Schwefel's Problem 1.2	$F_2=\sum_{i=1}^{D}(\sum_{j=1}^{D} z_j)^2 + f_b$	$[-100,100]^D$	-450
Shifted Rotated High Conditioned Elliptic	$F_3=\sum_{i=1}^{D}(10^6)^{\frac{i-1}{D-1}} z_i^2 + f_b$	$[-100,100]^D$	-450
Shifted Schwefel's Problem 1.2 with Noise in Fitness	$F_4=(\sum_{i=1}^{D}(\sum_{j=1}^{D} z_j)^2 * (1+0.4\|N(0,1)\|) + f_b$	$[-100,100]^D$	-450
Schwefel's Problem 2.6 with Global Optimum on Bounds	$F_5=\max\|A_i x - B_i\| + f_b$ *where* A_i is the ith row of a D*D matrix A andB_i is the ith element of D*1 vector B	$[-100,100]^D$	-310
Shifted Rosenbrock's Function	$F_6=\sum_{i=1}^{D}(100(z_i^2 - z_{i+1})^2 + (z_i - 1)^2) + f_b$	$[-100,100]^D$	390
Shifted Rotated Griewank's Function without Bounds	$F_7=\sum_{i=1}^{D}\frac{z_i^2}{4000} - \prod_{i=1}^{D} cos(\frac{z_i}{\sqrt{i}}) + 1 + f_b$	$[0,600]^D$	-180

Name	Objective Function	Search Range	Bias(f_b)
Shifted Rotated Ackley's Function with Global Optimum on Bounds	$F_8 = -20exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} z_i^2}) - exp(\frac{1}{D}\sum_{i=1}^{D} cos(2\pi z_i)) + 20 + e + f_b$	$[-32, 32]^D$	-140
Shifted Rastrigin's Function	$F_9 = \sum_{i=1}^{D} (z_i^2 - 10cos(2\pi z_i) + 10) + f_b$	$[-5, 5]^D$	-330
Shifted Rotated Rastrigin's Function	$F_{10} = \sum_{i=1}^{D} (z_i^2 - 10cos(2\pi z_i) + 10) + f_b$	$[-5, 5]^D$	-330
Shifted Rotated Weierstrass Function	$F_{11} = \sum_{i=1}^{D} (\sum_{k=0}^{kmax} [a^k cos(2\pi b^k (z_i + 0.5))]) - D \sum_{k=0}^{kmax} [a^k cos(2\pi b^k * 0.5)] + f_b \quad where \quad a = 0.5, b = 3 \;\&\; kmax = 20.$	$[-0.5, 0.5]^D$	90
Schwefel's Problem 2.13	$F_{12} = \sum_{i=1}^{D} (\lvert A_i - B_i(x) \rvert)^2 + f_b \quad where \quad A_i = \sum_{j=1}^{D} (a_{ij} sin\alpha_j + b_{ij} cos\alpha_j) \quad and \quad B_i(x) = \sum_{j=1}^{D} (a_{ij} sinx_j + b_{ij} cosx_j)$	$[-\pi, \pi]^D$	-460
Shifted Expanded Griewank's plus Rosenbrock's Function (F8F2)	$F_{13} = F8(F2(z_1, z_2)) + F8(F2(z_2, z_3)) + \cdots + F8(F2(z_{D-1}, z_D)) + F8(F2(z_D, z_1)) + f_b$	$[-5, 5]^D$	-130
Shifted Rotated Expanded Scaffer's F6 Function	$F_{14} = F(z_1, z_2) + F(z_2, z_3) + \cdots + F(z_{D-1}, z_D) + F(z_D, z_1) + f_b$ $F(x, y) = 0.5 + \frac{(sin^2(\sqrt{x^2+y^2}) - 0.5)}{(1 + 0.0001(x^2+y^2))^2}$	$[-100, 100]^D$	-300
Hybrid Composition Function	$F_{15} = \sum_{i=1}^{n} w_i * [f'_i(z) + bias_i] + f_b \quad where \; n = 10$ $f'_i = [F9, F9, F11, F11, F7, F7, F8, F8, F1, F1] \; without \; f_b$	$[-5, 5]^D$	120
Rotated Hybrid Composition Function	$F_{16} = \sum_{i=1}^{n} w_i * [f'_i(z) + bias_i] + f_b \quad where \; n = 10$ $f'_i = [F9, F9, F11, F11, F7, F7, F8, F8, F1, F1] \; without \; f_b$	$[-5, 5]^D$	120
F_{16} with Noise in Fitness	$F_{17} = G(x) * (1 + 0.2\lvert N(0,1) \rvert) + f_b \quad where \; G(x) = F_{16} - f_b(16)$	$[-5, 5]^D$	120
Rotated Hybrid Composition Function	$F_{18} = \sum_{i=1}^{n} w_i * [f'_i(z) + bias_i] + f_b \quad where \; n = 10$ $f'_i = [F8, F8, F9, F9, F1, F1, F11, F11, F7, F7] \; without \; f_b$	$[-5, 5]^D$	10
Rotated Hybrid Composition Function with a Narrow Basin for the Global Optimum	$F_{19} = \sum_{i=1}^{n} w_i * [f'_i(z) + bias_i] + f_b \quad where \; n = 10$ $f'_i = [F8, F8, F9, F9, F1, F1, F11, F11, F7, F7] \; without \; f_b$	$[-5, 5]^D$	10
Rotated Hybrid Composition Function with the Global Optimum on the Bounds	$F_{20} = \sum_{i=1}^{n} w_i * [f'_i(z) + bias_i] + f_b \quad where \; n = 10$ $f'_i = [F8, F8, F9, F9, F1, F1, F11, F11, F7, F7] \; without \; f_b$	$[-5, 5]^D$	10

Name	Objective Function	Search Range	Bias(f_b)
Rotated Hybrid Composition Function	$F_{21}=\sum_{i=1}^{n} w_i * [f_i'(z) + bias_i] + f_b \quad where\ n = 10$ $f_i' = [F14, F14, F9, F9, F13, F13, F11, F11, F7, F7]\ without\ f_b$	$[-5, 5]^D$	360
Rotated Hybrid Composition Function with High Condition Number Matrix	$F_{22}=\sum_{i=1}^{n} w_i * [f_i'(z) + bias_i] + f_b \quad where\ n = 10$ $f_i' = [F14, F14, F9, F9, F13, F13, F11, F11, F7, F7]\ without\ f_b$	$[-5, 5]^D$	360
Non-Continuous Rotated Hybrid Composition Function	$F_{23}=\sum_{i=1}^{n} w_i * [f_i'(z) + bias_i] + f_b \quad where\ n = 10$ $f_i' = [F14, F14, F9, F9, F13, F13, F11, F11, F7, F7]\ without\ f_b$	$[-5, 5]^D$	360
Rotated Hybrid Composition Function	$F_{24}=\sum_{i=1}^{n} w_i * [f_i'(z) + bias_i] + f_b \quad where\ n = 10$ $f_i' = [F11, F14, F13, F8, F9, F7, F13_{nc}, F9_{nc}, F3, F1]$ $without\ f_b\ (nc = non-continuous)$	$[-5, 5]^D$	260
Rotated Hybrid Composition Function without Bounds	$F_{25}=\sum_{i=1}^{n} w_i * [f_i'(z) + bias_i] + f_b \quad where\ n = 10$ $f_i' = [F11, F14, F13, F8, F9, F7, F13_{nc}, F9_{nc}, F3, F1]$ $without\ f_b\ (nc = non-continuous)$	$[2, 5]^D$	260

B CEC2013 Benchmark Functions

The CEC2013 benchmark functions are a set of improved, more complex set of real parameter single objective optimization problems. All these problems have been rotated and shifted to test the capabilities of any optimization algorithm in solving complex problems. There are 28 benchmark functions categorized as Unimodal (F_1 - F_5), Basic Multimodal (F_6 - F_{20}) and Composition functions (F_{21} - F_{28}). The global optimum solutions for all the test functions are shifted to $\boldsymbol{o} = [o_1, o_2, \cdots, o_D]$ and all of them are defined in the same search range of $[-100, 100]^D$. The characteristics of the functions are made more complex by introducing $\mathbf{M_1}, \mathbf{M_2}, \cdots, \mathbf{M_3}$ which are the orthogonal (rotation) matrices used to rotate the functions. A summary of the functions as provided in [153] is given next:

No.	Function Name	Bias(f_b)
1	Sphere Function	-1400
2	Rotated High Conditioned Elliptic Function	-1300
3	Rotated Bent Cigar Function	-1200
4	Rotated Discus Function	-1100
5	Different Powers Function	-1000
6	Rotated Rosenbrock's Function	-900
7	Rotated Schaffers F7 Function	-800
8	Rotated Ackley's Function	-700
9	Rotated Weierstrass Function	-600
10	Rotated Griewank's Function	-500
11	Rastrigin's Function	-400
12	Rotated Rastrigin's Function	-300
13	Non-Continuous Rotated Rastrigin's Function	-200
14	Schwefel's Function	-100
15	Rotated Schwefel's Function	100
16	Rotated Katsuura Function	200
17	Lunacek Bi-Rastrigin Function	300
18	Rotated Lunacek Bi-Rastrigin Function	400

No.	Function Name	Bias(f_b)
19	Expanded Griewank's plus Rosenbrock's Function	500
20	Expanded Scaffer's F6 Function	600
21	Composition Function 1 (n=5,Rotated)	700
22	Composition Function 2 (n=3,Un-rotated)	800
23	Composition Function 3 (n=3,Rotated)	900
24	Composition Function 4 (n=3,Rotated)	1000
25	Composition Function 5 (n=3,Rotated)	1100
26	Composition Function 6 (n=5,Rotated)	1200
27	Composition Function 7 (n=5,Rotated)	1300
28	Composition Function 8 (n=5,Rotated)	1400

C Performance Comparison on CEC2015 Problems

The CEC2015 benchmark problems [191] are highly competitive and require efficient optimization algorithms to provide fast solutions with accuracy in the limited allocated budgets. This section compares and contrasts the capabilities of proposed human learning principles inspired PSO algorithms; the SRPSO, DMeSRPSO and DD-SRPSO on the numerically expensive optimization problems with the recently proposed PSO variants and well-known evolutionary algorithms. The selected algorithms for comparison include the Adaptive Inertia Weight PSO (AIWPSO) [192], the Teaching and Peer Learning PSO (TPLPSO) [37], the Example-based Learning PSO (ELPSO) [193], the Competitive Swarm Optimizer (CSO) [194], the Ancestral Differential Evolution (AncDE) algorithm [195], the Mean Variance Mapping Optimization (MVMO) algorithm [196] and the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [8]. The results for AncDE are taken from [195] and the results for MVMO and CMA-ES are taken from [197] and [198] respectively. In [198], twenty seven different variant of CMA-ES have been tested on the expensive optimization problems from CEC2015 [191] and the reported results are for the best variant among them. The results for all the other algorithms are generated on MATLAB R2013a on Dell Precision T3600 machine having 16Gb RAM and 64-bit operating system. To provide a fair comparison, all the guidelines of CEC2015 [191] has been strictly followed and the experiments were conducted on the 30-dimensional functions using 50 particles. The experiments were conducted 20 times and for each run the exact function evaluations were set to 1500 as given in [191].

The mean and standard deviation of the performances for all the algorithms are provided in Table 1. From the table, it can be easily seen that the performance of DD-SRPSO and CMA-ES are much better than the other selected algorithms. Among the other algorithms, only MVMO has provided the best solution on two simple multimodal functions out of fifteen functions. DD-SRPSO and CMA-ES performances are closer to each other where DD-SRPSO is leading with solutions closer to the optimum in seven functions out of fifteen compared to six out of fifteen for CMA-ES. The performance of DD-SRPSO is better than the other selected algorithms on simple multimodal and composition functions, whereas among them CMA-ES is better for the hybrid functions.

Performance of DD-SRPSO, MVMO and CMA-ES on the expensive optimization problems are closely similar to each other which suggest that these three algorithms may have a common strategy among them for solving any optimization problem. If one closely studies these three algorithms, it will be observed that the concept of intelligent swarms has been implemented in all the three algorithms. In CMA-ES, the update strategy is performed in such a way that the likelihood of previous successful candidate is maximized. Similarly, in DD-SRPSO, the increased inertia weight for the best particle ensures maximum exploration of the best solution. Usage of memory for each swarm in

Table 1: Mean and Standard Deviation Performances on 30D CEC2015 Benchmark Functions

BFs	DD-SRPSO		DMeSR-PSO		SRPSO		AIWPSO		TPLPSO	
	Mean	STD.	Mean	STD.	Mean	STD.	Mean	STD.	Mean	STD.
F_1	1.960E+07	1.165E+07	1.396E+08	5.764E+08	1.277E+09	7.253E+08	2.813E+10	6.332E+09	3.483E+10	6.424E+09
F_2	**2.559E+04**	1.057E+04	6.051E+04	4.209E+03	7.566E+04	1.378E+04	8.665E+04	1.703E+04	1.780E+05	4.093E+04
F_3	2.243E+01	2.124E+00	3.330E+01	2.406E+00	3.048E+01	3.614E+00	3.523E+01	3.468E+00	3.917E+01	1.883E+00
F_4	1.833E+03	3.375E+02	5.548E+03	4.482E+02	6.309E+03	5.735E+02	7.150E+03	4.718E+02	8.410E+03	3.700E+02
F_5	**2.719E+00**	7.580E-01	4.107E+00	6.332E-01	4.331E+00	6.893E-01	4.334E+00	6.858E-01	4.707E+00	4.979E-01
F_6	**5.616E-01**	1.007E-01	7.162E-01	8.565E-01	6.344E-01	1.110E-01	1.082E+00	2.281E-01	4.394E+00	5.250E-01
F_7	5.432E-01	2.031E-01	7.210E-01	6.150E-01	7.938E-01	1.054E+00	4.808E+01	1.178E+01	4.204E+01	2.055E+01
F_8	3.322E+02	2.163E+02	1.691E+03	1.070E+03	3.197E+03	5.093E+03	1.010E+06	1.013E+06	2.548E+07	2.150E+07
F_9	**1.358E+01**	8.546E-01	1.366E+01	9.258E-01	1.364E+01	2.338E-01	1.369E+01	2.135E-01	1.380E+01	2.536E-01
F_{10}	**3.084E+05**	4.535E+05	3.546E+06	2.540E+06	1.222E+07	5.380E+06	2.481E+07	1.075E+07	3.931E+07	1.742E+07
F_{11}	4.596E+01	2.229E+01	5.642E+01	5.290E+01	6.839E+01	3.282E+01	1.439E+02	3.785E+01	2.866E+02	1.070E+02
F_{12}	6.923E+02	2.125E+02	8.056E+02	3.524E+02	8.387E+02	2.324E+02	1.364E+03	2.762E+02	1.749E+03	3.923E+02
F_{13}	**3.348E+02**	1.075E+01	4.077E+02	3.520E+02	4.116E+02	2.173E+01	6.247E+02	8.884E+01	9.533E+02	1.450E+02
F_{14}	**2.444E+02**	4.710E+00	2.668E+02	1.892E+02	2.591E+02	1.581E+01	3.226E+02	3.032E+01	3.524E+02	3.974E+01
F_{15}	9.236E+02	1.914E+02	9.952E+02	5.986E+02	1.018E+03	1.329E+02	1.301E+03	1.051E+02	1.365E+03	5.579E+01

	ELPSO		CSO		AncDE		MVMO		CMA-ES	
	Mean	STD.	Mean	STD.	Mean	STD.	Mean	STD.	Mean	STD.
F_1	3.147E+10	3.804E+09	8.1949E+09	2.804E+09	5.210E+08	3.540E+08	3.008E+07	2.865E+07	**2.032E+05**	6.117E+05
F_2	9.624E+04	1.869E+04	1.6026E+05	3.749E+04	1.180E+05	2.490E+04	1.389E+05	3.419E+04	1.174E+05	2.199E+04
F_3	4.103E+01	1.790E+00	3.5342E+01	3.879E+00	2.670E+01	4.480E+00	3.543E+01	3.892E+00	**2.131E+01**	5.063E+00
F_4	7.962E+03	2.709E+02	7.6389E+03	4.569E+02	6.780E+03	4.940E+02	**9.685E+02**	2.865E+02	7.588E+03	2.186E+03
F_5	4.004E+00	6.324E-01	5.1738E+00	7.864E-01	4.240E+00	5.910E-01	3.912E+00	8.133E-01	5.437E+00	5.919E-01
F_6	4.235E+00	2.003E-01	2.4349E+00	9.620E-01	5.890E-01	1.390E-01	6.335E-01	1.240E-01	6.854E-01	2.357E-01
F_7	6.349E+01	1.027E+01	3.2993E+01	1.467E+01	5.770E-01	2.080E-01	**4.566E-01**	1.801E-01	6.363E-01	2.860E-01
F_8	2.378E+06	1.032E+06	4.7463E+04	4.188E+04	4.480E+02	5.510E+02	1.753E+02	3.125E+02	**2.291E+01**	1.093E+01
F_9	1.377E+01	1.551E-01	1.3790E+01	6.072E+00	1.380E+01	2.030E-01	1.367E+01	2.898E-01	1.385E+01	3.030E-01
F_{10}	5.661E+07	2.096E+07	4.1404E+06	9.561E+05	1.810E+07	7.950E+06	5.821E+06	3.243E+06	2.774E+06	2.213E+06
F_{11}	2.175E+02	4.785E+01	4.9924E+01	3.457E+01	4.240E+01	3.440E+01	6.424E+01	4.575E+01	**2.783E+01**	1.982E+01
F_{12}	2.104E+03	3.706E+02	1.0548E+03	6.753E+02	1.370E+03	1.510E+02	7.053E+02	1.985E+02	**6.720E+02**	2.825E+02
F_{13}	7.905E+02	9.303E+01	5.2289E+02	1.042E+02	3.680E+02	9.110E+00	3.658E+02	2.255E+01	3.771E+02	1.191E+01
F_{14}	3.828E+02	2.204E+01	2.9894E+02	4.568E+01	2.710E+02	2.240E+01	2.796E+02	3.204E+01	2.457E+02	1.567E+01
F_{15}	1.401E+03	4.871E+01	1.3177E+03	3.452E+02	9.710E+02	1.230E+02	9.619E+02	2.702E+02	**7.581E+02**	7.910E+01

MVMO for tracking their behaviour and the dynamic reduction of swarm size capitalize the services of efficient candidates in the search space. All the above three algorithms give more importance to better performing candidates and the less efficient candidates are either guided (as in DD-SRPSO) or discarded (as in MVMO) or given lower likelihood (as in CMA-ES). Using these strategies, the algorithms ensure that potential areas of the search space are explored resulting in a faster convergence. Next, a comparative analysis using rank based and statistical test has been conducted to provide the significance of performance.

Table 2: Rank based analysis of Mean Performances

Algorithm	Individual Ranking of Benchmark Functions															Avg. RANK	RANK
	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}		
DD-SRPSO	2	1	2	2	1	1	2	3	1	1	3	2	1	1	2	1.667	1
DMeSR-PSO	4	2	5	3	4	6	5	5	3	3	5	4	5	4	5	4.2	4
SRPSO	6	3	4	4	6	4	6	6	2	6	7	5	6	3	6	4.933	6
AIWPSO	8	4	6	6	7	7	9	8	5	8	8	7	8	8	7	7.067	8
TPLPSO	10	10	10	10	8	10	8	10	8	9	10	9	10	9	9	9.333	10
ELPSO	9	5	9	9	3	9	10	9	6	10	9	10	9	10	10	8.467	9
CSO	7	9	7	8	9	8	7	7	7	4	4	6	7	7	8	7	7
AncDE	5	7	3	5	5	2	3	4	9	7	2	8	3	5	4	4.8	5
MVMO	3	8	8	1	2	3	1	2	4	5	6	3	2	6	3	3.8	3
CMAES	1	6	1	7	10	5	4	1	10	2	1	1	4	2	1	3.733	2

C.1 Rank Based Analysis and Statistical Comparison

In this section, the mean performances on the functions of all the selected algorithms are ranked and the algorithm with lowest average rank is considered as the best performing algorithm. These average ranks are then used to perform a statistical comparison of performances as performed for CEC2005 and CEC2013 benchmark Problems in Chapter 3, 4 and 5.

In table 2, the individual ranks, average ranks and the final ranks of all the evolutionary algorithms are presented. From the table, it is clear that DD-SRPSO has consistently achieved better ranks compared to the other algorithms. The average rank of DD-SRPSO is low compared to all the other algorithms and therefore it has been ranked as best algorithm. Based on the average ranks, DD-SRPSO is top performer followed by CMA-ES, MVMO, DMeSR-PSO, AncDE, SRPSO, CSO, AIWPSO, ELPSO and TPLPSO respectively. In the average ranks, the compared PSO variants are all low performer whereas the DD-SRPSO, DMeSR-PSO and SRPSO are all performing comparable with the highly competitive evolutionary algorithms. This suggests that the performance of proposed human learning principles inspired PSO algorithms are comparatively better than other PSO algorithms and are also comparable with other evolutionary algorithms. To have a better understanding of the performance of selected algorithms a statistical test has been performed.

Next, a statistical comparison has been conducted to provide the significance of DD-SRPSO over other evolutionary algorithms using the non-parametric Friedman test on the average rank followed by the pairwise post-hoc Bonferroni-Dunn test [128]. The average rank difference with respect to DD-SRPSO is 2.067, 2.133, 2.533, 3.133, 3.267, 5.333, 5.400, 6.800 and 7.667 for CMA-ES, MVMO, DMeSR-PSO, AncDE, SRPSO, CSO, AIWPSO, ELPSO and TPLPSO respectively. First, the Friedman test has been conducted for rejecting the null hypothesis. The computed F-score is 23.4039 and for ten algorithms and fifteen problems the F-statistics value at a confidence level of 95% according to the F-Table is 1.9549. Comparing the F-score and the F-statistics, the null hypothesis can be rejected and it can be concluded that the performance of the algorithms is statistically different. Therefore, the pairwise post-hoc Bonferroni-Dunn test has been conducted in which the critical difference in average ranks at 95% is 1.3852. Hence, it can be concluded that the performance of DD-SRPSO is significantly better than the selected algorithms with 95% confidence level.

To summarize, the performance of DD-SRPSO on numerically expensive CEC2015 benchmark functions compared to other PSO variants and evolutionary algorithms has validated the capabilities of DD-SRPSO algorithm in solving expensive budget problems effectively. The rotational invariance strategy has provided the particles better capabilities to tackle with the rotated problems and converge to optimum/ near-optimum solutions. Similarly, the direction update strategy has diverged the particles towards the potential solutions which has resulted in faster convergence. The rank based analysis and statistical comparison has proved that DD-SRPSO has provided highly competitive solutions with 95% confidence level. Further, it can also be seen from table 2, that the difference in average rank of selected PSO variants compared to SRPSO and DMeSR-PSO are greater than the critical difference value. This suggests that the performance of SRPSO and DMeSR-PSO are significantly better than selected PSO variants with 95% confidence level. Also, it has been observed that DD-SRPSO is best suited for solving simple multimodal and highly complex composition functions with better accuracy whereas the performance is not significant on the hybrid functions.

D Comparative Analysis of SRPSO, DMeSR-PSO and DD-SRPSO using Box Plots

Figure 1 contains the box plots of the performance on functions F_1 and F_8. The figure demonstrates the behaviour of all the three algorithms in solving the benchmark functions. From the box plots, one can see all the three algorithms have exhibited similar performances in functions F_1, F_5, F_7 and F_8. In function F_3, SRPSO and DD-SRPSO have shown similar behaviour and have achieved solutions with robustness (all the solutions are close to each other) whereas the performance on DMeSR-PSO is spread over a large range of values. The performance of both DMeSR-PSO and DD-SRPSO is much better than SRPSO in functions F_2 whereby DD-SRPSO has a smaller median value. In functions F_4, DMeSR-PSO have shown a high level of accuracy across the 51 runs and in function F_6 the same has been observed for DD-SRPSO. With an exception of few outliers, the performance of DD-SRPSO on function F_6 is much better than others.

The box plots for functions F_9 and F_{16} are presented in figure 2. Form the figure, it can be seen that the performance of DD-SRPSO is much better than others in functions F_{10}, F_{12}, F_{13}, F_{15} and F_{16}. With the exception of functions F_{11} and F_{12}, all the algorithms have exhibited similar performances whereby following observations have been made:

- SRPSO has a slightly better performance in function F_9.
- SRPSO and DMeSR-PSO are almost identical in solving function F_{10} where more outliers are observed for SRPSO.
- DMeSR-PSO has shown much better performance than others in function F_{14}.

The box plots for the next eight functions F_{17} to function F_{24} are shown in figure 3. From the box plots, one can easily identify the best performer as in almost every function

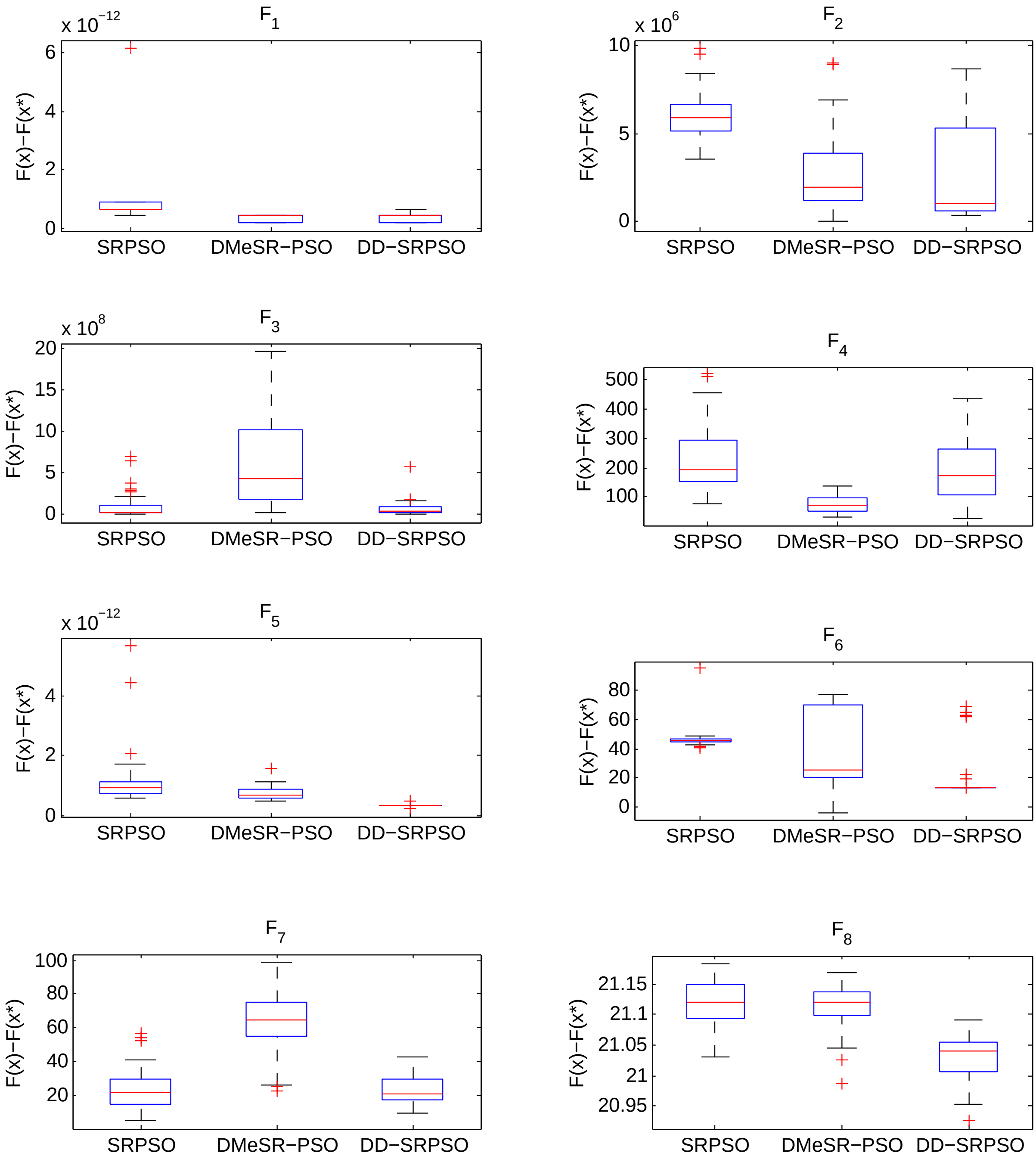

Figure 1: Box Plot for functions F_1 to F_8

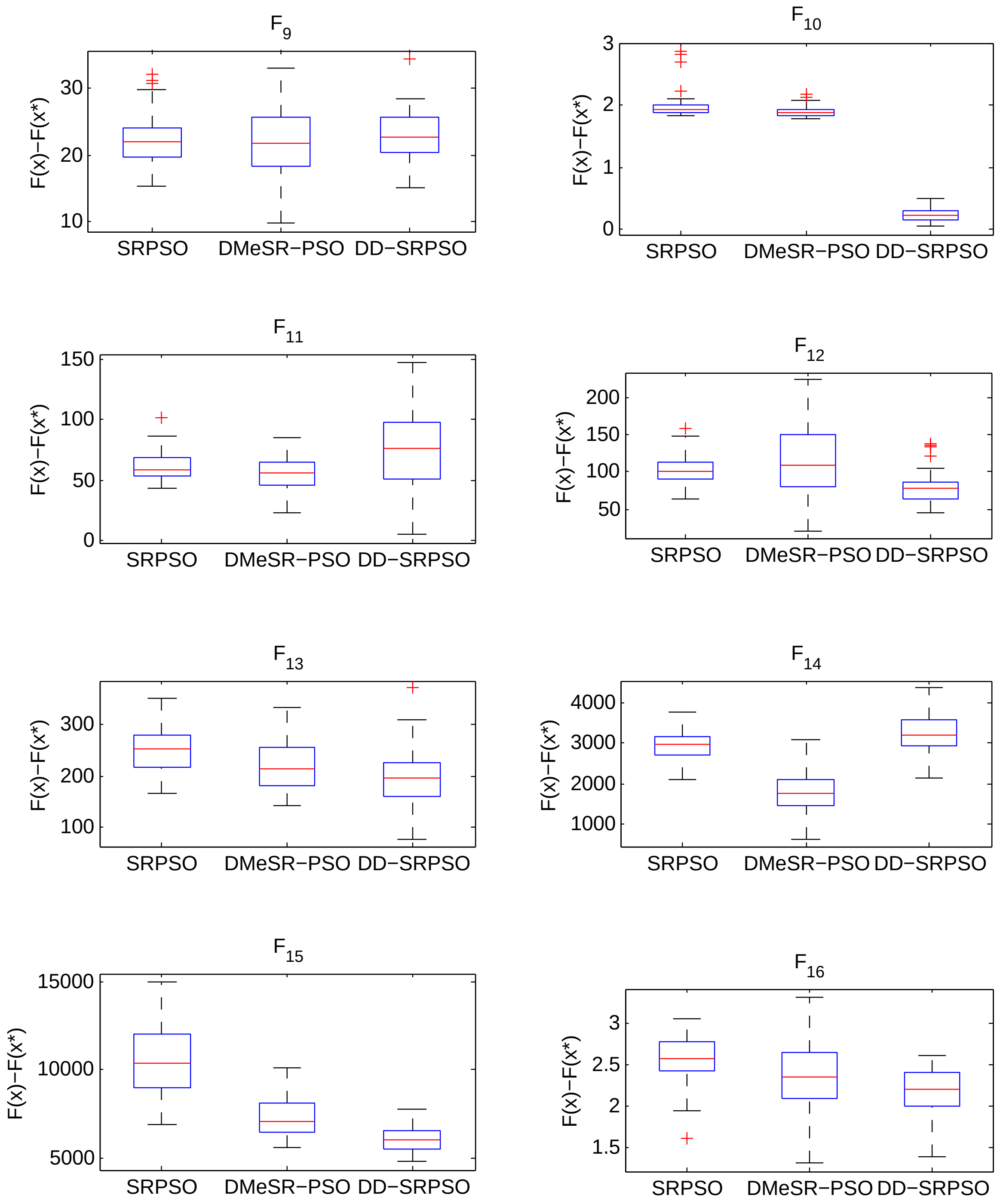

Figure 2: Box Plot for functions F_9 to F_{16}

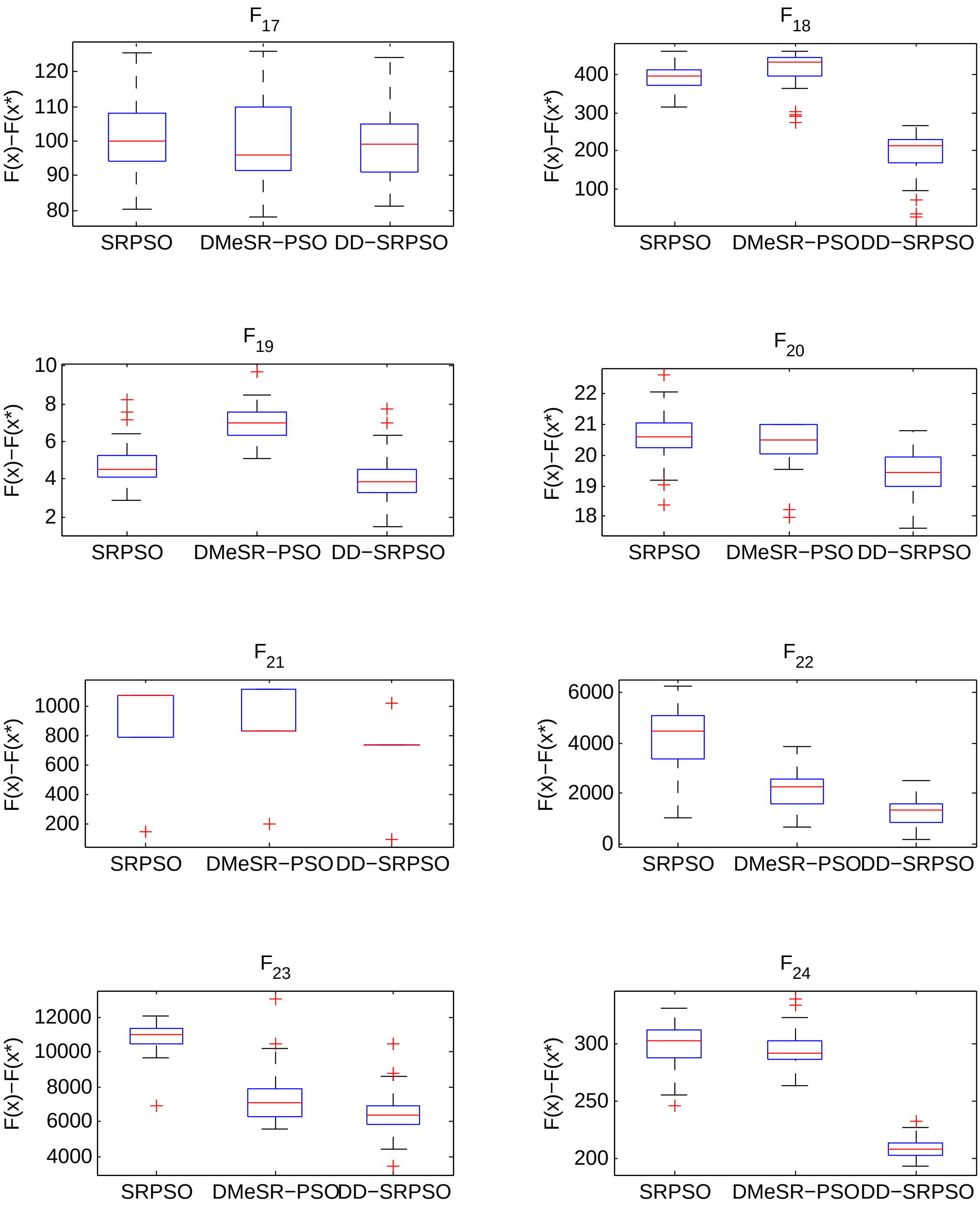

Figure 3: Box Plot for functions F_{17} to F_{24}

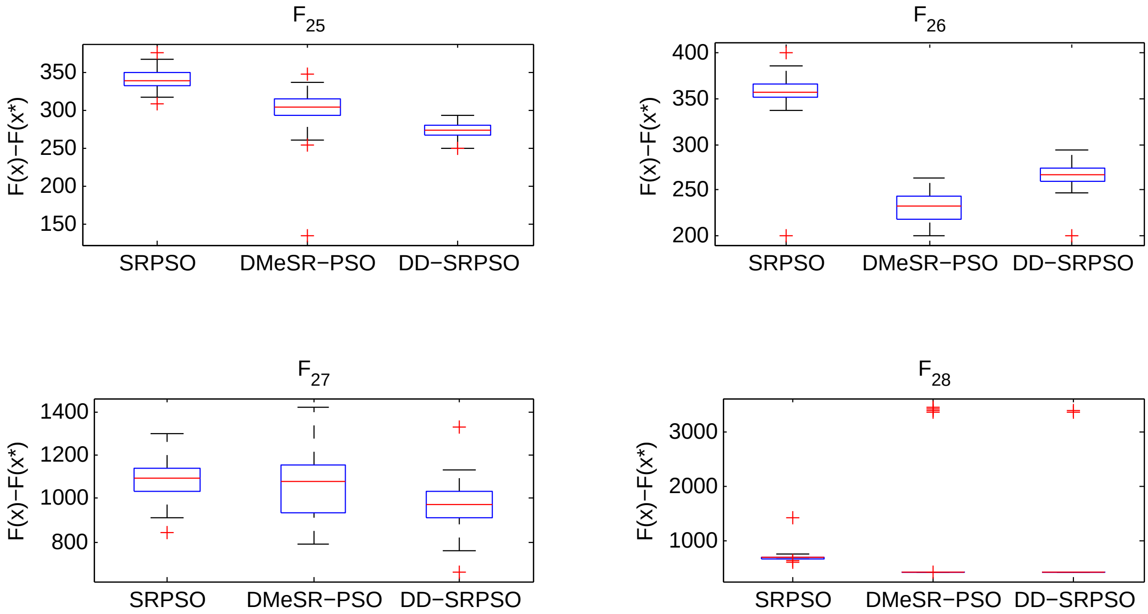

Figure 4: Box Plot for functions F_{25} to F_{28}

the performance of DD-SRPSO is better than other two algorithms. In function F_{21}, DD-SRPSO has converged to exactly the same point most of the times. The most blots for the final four functions F_{25} to function F_{28} are provided in figure 4. The algorithms have shown mixed performances on these four functions and narrow plots have been observed in almost all functions that suggest the algorithms have solved the problems with robustness. In these 12 functions F_{17} to F_{28}, following observations have been made:

- DD-SRPSO has much better performance in almost all functions.
- All 3 functions have identical performance on functions F_{17}, F_{20}, F_{25} and F_{27}.
- DMeSR-PSO and DD-SRPSO both have identical performances in function F_{28} .

As a whole, the performance of SRPSO, DMeSR-PSO and DD-SRPSO on the set of 28 benchmark functions has been fairly good as in most cases smaller boxes have been observed that represent the robustness and consistency of the algorithms in providing solutions closer to a point in every run. In fact, with an exception of few outliers the algorithms have converged to almost the same point in functions F_5, F_7, F_{10}, F_{18} and F_{28}. Hence, it can be concluded that all the three algorithms have shown consistent performance with sustainability and robustness on the diverse set of benchmark functions whereby DD-SRPSO has provided much improved solutions.