INNOVATIVE HYBRID METAHEURISTIC ALGORITHMS AND THEIR APPLICATIONS UNDER FUZZY OR DETERMINISTIC ENVIRONMENT

Submitted in partial fulfilment of the requirements of the degree of

Doctor of Philosophy

By

Subhabrata Rath

(719098)

Supervisor

Prof. Debashis Dutta

Department of Mathematics

NATIONAL INSTITUTE OF TECHNOLOGY

WARANGAL - 506 004, (T.S.) INDIA

JANUARY 2024

Dedicated To

My beloved Parents

Satyanarayan Rath and Sunita Mishra

Approval Sheet

This Dissertation Work entitled **Innovative Hybrid Metaheuristic Algorithms and their Applications under Fuzzy or Deterministic Environment** by **Subhabrata Rath** is approved for

the degree of Doctor of Philosophy

Examiners

_____

_____

_____

Supervisor

Prof. Debashis Dutta

Chairman (Program Coordinator)

_____

Date_____ Place: _____

Declaration

This is to certify that the work presented in the thesis entitled "**INNOVATIVE HYBRID METAHEURISTIC ALGORITHMS AND THEIR APPLICATIONS UNDER FUZZY OR DETERMINISTIC ENVIRONMENT**" is a bonafide work done by me under the supervision of Dr. D. Dutta, Professor, Department of Mathematics, National Institute of Technology, Warangal and has not been submitted elsewhere for the award of any degree or diploma.

I declare that this written submission represents my ideas in my own words. Where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all academic honesty and integrity principles and have not misrepresented, fabricated, or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will cause disciplinary action by the Institute. I can also evoke penal action from the sources that have thus not been properly cited or from whom proper permission has not been taken when needed.

Subhabrata Rath

Roll No. 719098

Date:

List of Abbreviation

ACO	Ant Colony Optimization
ABC	Artificial Bee Colony
BA	Bat Algorithm
BF	Benchmark Functions
CA	Classification Accuracy
DM	Dominant Member
FS	Feature Selection
FE	Filter Evaluation
FDS	Finite Difference Scheme
FA	Firefly Algorithm
FAR	Friedmann Average Ranking
FOA	Fruit Fly Optimization Algorithm
FIS	Fuzzy Inference Systems
FLC	Fuzzy Logic Controller
FLR	Fuzzy Logical Relation
FN	Fuzzy Number
FYS	Fuzzy Set
FTS	Fuzzy Time Series
GA	Genetic Algorithm
GP	Genetic Programming
GWO	Grey Wolf Optimization
HHO	Harris Hawk Optimization
KNN	K-Nearest Neighbour
MOFS	Multi-Objective Feature Selection
MWUT	Mann Whitney U Test
MSE	Mean Square Error

MP	Multimodal Problem
MOO	Multi-Objective Optimization
MVO	Multi-Verse Optimization
NF	Number of Features
PSO	Particle Swarm Optimization
PFN	Pentagonal Fuzzy Number
PFC	Picture Fuzzy Clustering
PFYS	Picture Fuzzy Set
SSA	Salp Swarm Algorithm
SBISP	State Bank of India Share price
SMA	Slime Mould Algorithm
TSD	Time Series Data
TFN	Trapezoidal Fuzzy Number
UP	Unimodal Problem
UOD	Universe of Discourse
UAE	Enrolments of Alabama University
WRST	Wilcoxon Rank Sum Test
WOA	Whale Optimization Algorithm
WE	Wrapper Evaluation

Abstract

The thesis focuses on developing Innovative metaheuristic algorithms (HPSO, PSOHHO, PSOHHO-V, PSOMHHO, and Fuzzy BTO) inspired by nature. Despite the proliferation of metaheuristic algorithms, a significant void exists in conducting thorough theoretical and mathematical analyses. More studies are needed, particularly in the critical domain of convergence of metaheuristic algorithms. The proposed algorithm's theoretical analysis and mathematical foundation are established to tackle this challenge by introducing the concept of signature and convergence. The stability analysis of HPSO is also discussed to strengthen it mathematically even further.

The thesis also emphasizes the applications of developed metaheuristic algorithms on different real-world problems in fuzzy and deterministic atmospheres. Job scheduling problems on computational grids commonly fall into NP-completeness or NP-hardness, making the quest for optimal solutions notably time-intensive. To address this challenge expediently and effectively, scholars have predominantly turned to the exploration of metaheuristic algorithms. Hence, the developed algorithm HPSO is applied to both single-objective and multi-objective Job Scheduling problems on the computational grid.

Classification problems frequently involve a surplus of features, but not all contribute to the problem's essence. Redundant or irrelevant features may impede classification accuracy. Metaheuristic algorithms are favoured for feature selection due to their simplicity and practical applicability, offering advantages over deterministic optimization algorithms. For this purpose, the real-world application of developed algorithms PSOHHO, PSOHHO-V, PSOMHHO, and Fuzzy BTO are verified on feature selection problems. They are applied to the hybrid Feature Selection problem and compared with other metaheuristic algorithms on seven UCI machine learning repository datasets.

Existing literature highlights the effectiveness of forecasting methods with more diverse set types. The idea of enhancing outcomes by incorporating additional inputs is closely observed. This theoretical basis advocates the efficiency of Picture Fuzzy Sets (PFYSs) in forecasting models. Notably, there needs to be more literature regarding the joint application

of PFYS and swarm intelligence for Fuzzy Time Series (FTS) forecasting. This chapter introduces the innovative EDSPSO-PFTS approach to address this gap.

This thesis is structured into five sections, encompassing ten chapters.

Part I consists of Chapter 1, an introduction, and motivation for the research work carried out. It features a comprehensive literature review emphasizing the importance of the thesis's focal issue.

In **Part II**, the emphasis shifts to the real-world deployment of the newly created hybrid metaheuristic algorithms, specifically in the context of Job Scheduling on a computational grid. There are two chapters in this part, which are 2 and 3.

Chapter 2 applies a proposed fuzzy particle swarm optimization to address single-objective job scheduling on computational grids. Focused on minimizing the makespan value, indicative of maximum task completion time across the grid, this chapter explores the impact of trapezoidal and pentagonal fuzzy numbers, presenting a detailed comparative analysis.

Chapter 3 explores the application of fuzzy particle swarm optimization for multi-objective job scheduling on computational grids. The foundation established in the previous chapter is expanded upon in this chapter, which moves from single-objective to multi-objective optimization. This chapter aims to maintain a delicate balance between makespan and flowtime objectives, which are conflicting.

The practical application of the proposed hybrid metaheuristic algorithms in the hybrid feature selection problem domain is the main emphasis of **Part III** of the thesis. There are five chapters in this section: Chapters 4, 5, 6, 7, and 8.

In **Chapter 4**, a Hybrid Particle Swarm Optimization (HPSO) algorithm is introduced and applied to address a Hybrid Feature Selection problem. This chapter discusses the stability of the proposed HPSO algorithm through the Von Neumann stability criterion and Fourier series concepts. The convergence of HPSO is explained using the Markov chain concept. The chapter thoroughly compares results against other metaheuristic algorithms, employing statistical tests like the Friedman and Mann-Whitney U test to estimate the algorithm's statistical significance.

In **Chapter 5**, an improved version of the Particle Swarm Optimization algorithm, HPSO, is thoroughly explored for its application in multi-objective feature selection. This section represents a notable progression from single-objective to multi-objective optimization, leveraging the foundation in the previous chapter. The effectiveness of HPSO is meticulously evaluated across seven UCI datasets, with robust statistical analysis facilitated by the Wilcoxon rank sum test.

In **Chapter 6**, two innovative hybrid optimization algorithms, PSOHHO and its variant PSOHHO-V, are introduced. These algorithms enhance their exploration capabilities by integrating a dual-swarm strategy and an exponential mutation operator. The evaluation of PSOHHO and PSOHHO-V involves rigorous analysis of statistical metrics and convergence rates, with extensive testing on benchmark functions. Furthermore, the algorithms are applied to feature selection problems, and their performance is benchmarked against alternative approaches using seven UCI datasets.

In **Chapter 7**, a ground-breaking Hybrid Swarm Optimization algorithm, PSOMHHO, is introduced, seamlessly integrating Pentagonal and Trapezoidal Fuzzy Numbers. The chapter explores the mathematical foundations by proving the algorithm's convergence using the Markov Chain property and introducing the algorithm's signature. The effectiveness of PSOMHHO is meticulously evaluated through extensive benchmark function testing, establishing its superiority over established metaheuristic algorithms. Rigorous statistical tests, including the Mann-Whitney U test and the Friedman test, affirm the exceptional performance of PSOMHHO across various metaheuristic algorithms.

Chapter 8 introduces innovative hybrid swarm optimization algorithms (Fuzzy BTO and its variants). Emphasizing the critical role of parameters in optimization, this chapter introduces fuzzy concepts for dynamic parameter adaptation. The efficiency of Fuzzy BTO and its variants are rigorously evaluated through extensive benchmark functions, followed by a comprehensive comparison with established metaheuristic algorithms. The robust statistical analysis, employing the Kruskal-Wallis Test (KWT), validates the superior performance of Fuzzy BTO across various metaheuristic algorithms.

In **Part – IV** of this thesis, the emphasis shifts to applying the proposed hybrid metaheuristic algorithms and Picture Fuzzy Set on Forecasting. This critical part is summarized in a singular chapter, namely Chapter 9.

A novel picture fuzzy time series (PFTS) forecasting model built on the foundations of picture fuzzy sets (PFYSs) is presented in **Chapter 9**. This chapter develops a unique hybrid EDSPSO-PFTS forecasting approach by integrating PFYS and EDSPSO. To illustrate the applicability and utility of the proposed forecasting method, it is applied to data sets from the Alabama University and the State Bank of India share price at the Bombay Stock Exchange, India. Average forecasting error (AFE) and mean square error (MSE) are used to evaluate the efficiency of the suggested approach. Thorough statistical validation and performance analysis are carried out to guarantee the validity and dependability of the proposed approach.

The thesis summary for **Part V** is included in a single Chapter 10. It outlines the key findings from the study and points out the issues that still need to be addressed to further this field of study.

Contents

List of Figures

List of Tables

PART 1

INTRODUCTION

Chapter 1

Introduction

1.1 Motivation

Metaheuristic algorithms represent a cornerstone in optimization, offering versatile and adaptive solutions to diverse complex problems. These algorithms, characterized by their ability to navigate vast solution spaces efficiently, have emerged as indispensable tools in various research and application domains. They draw inspiration from natural phenomena or utilize heuristic strategies to discover near-optimal or satisfactory solutions. In general, two types of metaheuristic algorithms are present: a single solution-based (i.e., Simulated Annealing (SA)) and population-based (i.e., GA). In a single solution-based optimization algorithm, only one solution is processed. In the Population-based metaheuristic (P-metaheuristics) algorithms, a set of solutions is processed in each iteration of the optimization process. P-metaheuristics mostly take motivation from nature and mimic their behavior[1]. P-metaheuristics can be categorized mainly into three groups, based on the motivation it takes[2]: Evolutionary Algorithms (EAs), Swarm Intelligence (SI), and Physics-based algorithms as presented in Fig. 1.1. EAs mimic biological evolutionary behaviors like mutation, cross-over, and selection. Some popular EAs are GA, Differential Evolution (DE)[3], and Genetic Programming (GP). SI mimics the social behaviors of organisms living in swarms, flocks, or herds[4]. The bird flocking behavior is the main inspiration for the PSO proposed by Eberhart and Kennedy[5]. In PSO, each particle is a possible solution to a given optimization problem. Some popular SI are Cuckoo Search (CS)[6], Ant Colony Optimization (ACO)[7], and ABC. Physics-based algorithms are motivated by physical laws. Central force optimization and gravitational search algorithms (GSA)[8] are some examples of physics-based algorithms.

The no-free-lunch (NFL) theorem[9] states that no optimization algorithm can be the most efficient for every optimization problem. Some algorithms may perform better for specific situations than others. The trend of hybridizing metaheuristic algorithms is on the rise, as it can improve their performance in solving real-world optimization problems that are often non-linear and high-dimensional. Metaheuristic algorithms are preferred over deterministic optimization algorithms due to their simplicity and ease of implementation in real-life scenarios. Various metaheuristic algorithms, such as the Genetic Algorithm (GA)[10], social spider algorithm[11], Bat algorithm (BA)[12], Slime Mould algorithm (SMA)[13], Whale Optimization algorithm (WOA), Firefly algorithm (FA)[14], Salp swarm algorithm (SSA)[15], Grey wolf optimizer[16], Multi-verse optimization algorithm (MVO)[17], Fruit Fly optimization algorithm (FOA), and others have been applied to various real-world problems. Many hybrid optimization techniques that combine Artificial Bee Colony (ABC) and Particle Swarm Optimization (PSO) have also been used for path optimization problems.

1.2 Theoretical Analysis

Proper theoretical and mathematical analysis of many metaheuristic algorithms has yet to be done. Very little work has yet to be done regarding studying the convergence of metaheuristic algorithms. Since many metaheuristic algorithms suffer from premature convergence or get stuck in local optima, proper mathematical analysis of any given metaheuristic algorithm is very important. Different algorithms have been proposed to prove the convergence of metaheuristic algorithms, such as multi-objective PSO[18] and Markov Chain for Chicken Swarm Optimization[19]. Markov Chain is a random process with a strong capability for probabilistic analysis and convergence analysis of randomized algorithms. It has been successfully implemented on the ABC algorithm, the PSO, the ACO, and the SA. Stability analyses have been conducted on a range of algorithms, such as PSO and ABC, using Von Neumann stability analysis, Differential Evolution (DE) employing both Von Neumann and Lyapunov stability criteria, Gravitational Search Algorithm (GSA) concerning Lyapunov stability criterion, and Bacterial Foraging Optimization (BFO).

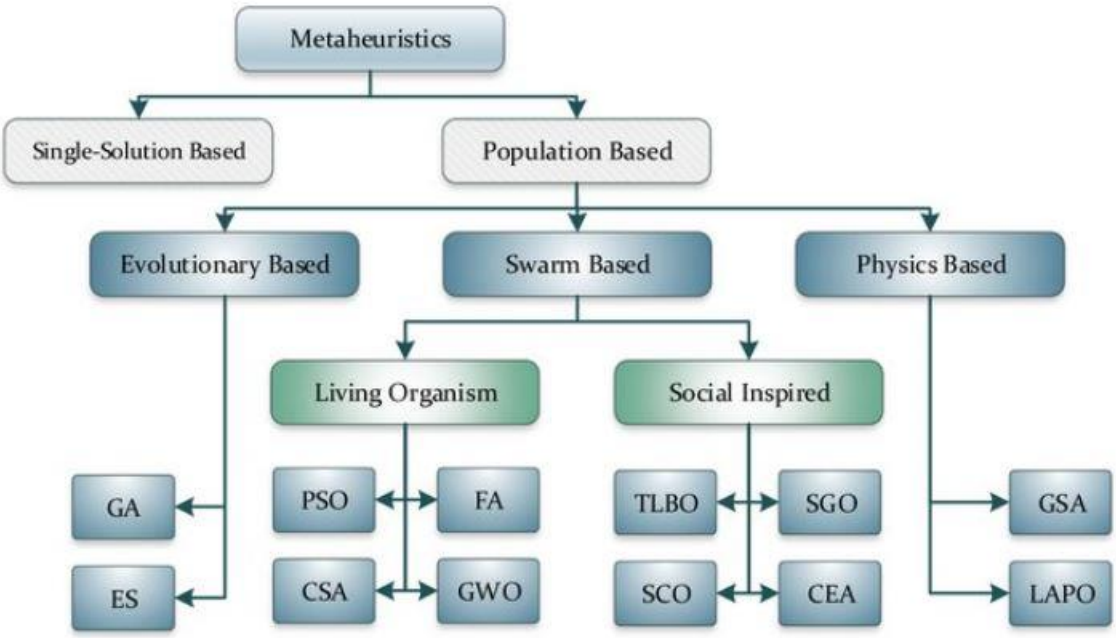

Fig 1.1. Classification of P-metaheuristic algorithm

1.3 Preliminaries

In this section, we lay out the fundamental concepts crucial for a thorough comprehension of the forthcoming chapters. Section 1.3.1 provides insights into the fundamental ideas related to Fuzzy Set (FYS). Subsequently, Sections 1.3.2, 1.3.3, and 1.3.4 elaborate on the concepts of Harris Hawk Optimization (HHO), Particle Swarm Optimization (PSO), and GA, respectively.

1.3.1 Fuzzy Set (FYS)

Zadeh's ground-breaking proposition of FYS in the literature marked a paradigm shift, employing human logic to tackle engineering challenges while algorithmically summarising human decision-making and evaluation processes[20]. FYS theory is used for transposing human logical and adaptable thought processes into the domain of computational intelligence. Consequently, it emerges as an indispensable tool for confronting complexities stemming from incompleteness, unreliability, vagueness, randomness, and imprecision within artificial intelligence applications. This fusion of FYS with artificial intelligence has catalyzed advancements across various products. An illustrative instance is in self-driving cars, where fuzzy logic-based artificial intelligence takes the lead. FYS furnishes us with mathematical constructs to emulate human reasoning, assuming a pivotal role in applications rooted in

human-like cognition, including humanoid robots and human decision-making. The literature review attests to the surging adoption of intelligent techniques integrated with FYSs, a trend that sees growing traction with each passing year. The landscape of FYSs has undergone a transformative phase through the introduction of extensions that offer complex details about membership functions. This has sparked notable research interest, particularly in areas like Picture Fuzzy Set (PFYS), Spherical FYS, Fermatean FYS, Circular Intuitionistic FYS, and Decomposed FYS. Fig 1.2 serves as an illustrative overview of these extensions, signifying a substantial expansion beyond the conventional FYS.

Fuzzy Number

A FYS is called a fuzzy number when the following properties are satisfied

- A must be a normal FYS.
- All α-cuts of A must be in a closed interval.
- The support of A must be bounded.

Trapezoidal Fuzzy Number (TFN)

A fuzzy number is called a TFN if the following conditions are satisfied-

- Let the TFN be denoted by (a, b, c, u) with membership function (x).
- (x) must be a continuous membership function whose interval is [0,1].
- (x) must be a strictly non-decreasing function that is continuous on the intervals $[a, b]$.
- $(x) = 1$, in the interval $[b, c]$.
- (x) must be a strictly non-increasing function that is continuous on the intervals $[c, u]$.

Pentagonal Fuzzy Number (PFN)

A fuzzy number is called a PFN if the following conditions are satisfied-

- Let the PFN be denoted by $(\mathrm{a}, \mathrm{b}, \mathrm{c}, \mathrm{d}, \mathrm{e})$ with membership function (x).
- (x) must be a continuous membership function whose interval is [0,1].
- (x) must be a strictly non-decreasing function that is continuous on the intervals $[a, b]$ and $[b, c]$.

- (x) must be a strictly non-increasing function that is continuous on the intervals $[c, d]$ and $[d, e]$.

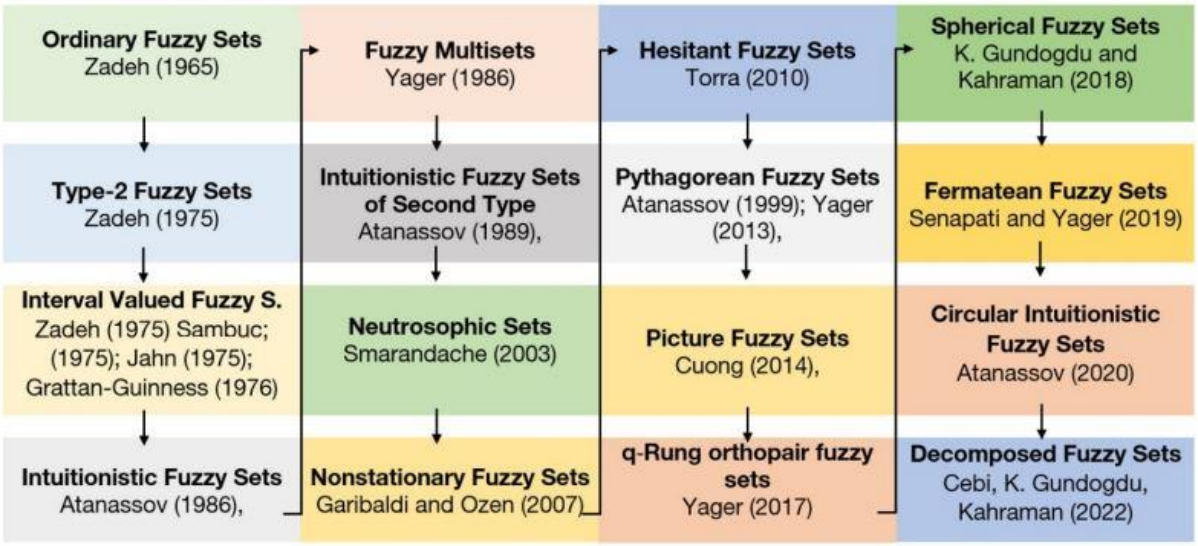

Fig 1.2. Extension of FYSs

1.3.2 Harris Hawk Optimization (HHO)

HHO is an optimization technique based on swarm Intelligence. HHO consists of two phases: exploration and exploitation. The value of E_E helps us to know whether to be in the exploration or exploitation phase. In the exploration stage, HHO is first randomly located. There are two cases in the Exploration stage. Each case has an equal chance of selection based on the random value, as in Table 1.1. Again, in the exploitation phase, there are four cases. The updating in the exploitation stage is based on the importance of E_E and a random value, as given in Table 1.1. The notations of HHO are shown in Table 1.1.

Exploration

In this phase, the particles move according to Eq. (1.1). Again, the value of c_0 determines which equation should be used. Finally, the value of $z_{araverage}(t)$ is computed using Eq. (1.2).

$$z(t+1) = \begin{cases} z_r(t) - c_1|z_r(t) - 2c_2 z(t)|, & c_0 \geq 0.5 \\ (z_{target}(t) - z_{araverage}(t)) - c_3 \times (c_4(u-l)+l), & c_0 < 0.5 \end{cases} \quad (1.1)$$

$$z_{average}(t) = \left(\left(\frac{1}{N}\right) \times \left(\sum_{i=1}^{N} z_i(t)\right)\right) \quad (1.2)$$

Phase change between exploitation and exploration

It depends on the value of E_E. Whenever the value of $|E\_E|$ is greater than one, HHO performs exploration. Otherwise, it performs exploitation. The value of E_E is computed using Eq. (1.3).

$$E\_E = 2E_0\left(1 - \frac{t}{T}\right) \quad (1.3)$$

Here $E_0 = (2 \times rand) - 1$ is the initial Energy updated in each iteration.

Table 1.1. Notations of the HHO algorithm

Notation	**Description**
E_E	Escaping Energy
$z(t), z_r(t)$	Current position, the position of a random individual at iteration t, respectively
$c_0, c_1, c_2, r, v, c, c_3, c_4, c_5$	Random numbers in the range [0,1]
$z_{average}(t)$	average position of the population set at iteration t
$z_{target}(t)$	global best position
u, l	Upper and lower bound
T, t	Maximum iteration and current iteration
E_0	Initial Energy
D	Dimension of the problem

Exploitation

Four cases are given to study the exploitation stage. Each case depends on the E_E and random number c.

Case A

is When $|E\_E| \geq 0.5$ and $c \geq 0.5$. The following equation updates the current locations-

$$z(t+1) = \Delta z(t) - E\_E \times |(Jump\_strength) \times z_{target}(t) - z(t)| \quad (1.4)$$

Where,

$$\Delta z(t) = z_{target}(t) - z(t) \quad (1.5)$$

$$Jump\_strength = 2(1 - c_5) \quad (1.6)$$

Case B

When $|E\_E| < 0.5$ and $c \geq 0.5$. The following equation updates the current locations-

$$z(t+1) = z_{target}(t) - E\_E \times |\Delta z(t)| \quad (1.7)$$

Case C

When $|E\_E| \geq 0.5$ and $c < 0.5$. The search agents follow the next move according to the following rule

$$Y = z_{target}(t) - E\_E \times |(Jump\_strength) \times z_{target}(t) - z(t)| \quad (1.8)$$

The Levy flight concept is used here. The rule is as follows-

$$a = Y + \left(r \times v \times levy(D)\right) \quad (1.9)$$

The levy flight distribution is as follows-

$$levy(x) = \left(\frac{u \times \sigma}{|v|^{\frac{1}{\beta}}}\right) \quad (1.10)$$

$$\sigma = \left[\frac{\Gamma(1+\beta) sin sin \frac{(\Pi\beta)}{2}}{\Gamma\left(\frac{(1+\beta)}{2}\right) * \beta * 2^{\frac{(\beta-1)}{2}}}\right]^{\frac{1}{\beta}} \quad (1.11)$$

Hence, the mechanism for this case is as follows-

$$z(t) = \begin{cases} Y, & if\ F(Y) < z(t) \\ a, & if\ F(a) > z(t) \end{cases} \quad (1.12)$$

Case D

When | *E_E* |< 0.5 and $c < 0.5$, The search agents follow the next move according to the following rule

$$Y` = z_{target}(t) - E\_E \times |(Jump\_strength) \times z_{target}(t) - z_{araverage}(t)| \quad (1.13)$$

$$a` = Y` + \left(r * v * levy(D)\right) \quad (1.14)$$

$$z(t) = \begin{cases} Y`, & if\ F\left(Y`\right) < z(t) \\ a`, & if\ F\left(a`\right) > z(t) \end{cases} \quad (1.15)$$

1.3.3 PSO

Kennedy and Eberhart introduced the PSO technique, which is nature-inspired optimization[5]. It is a metaheuristic technique that works well for challenges encountered in daily life. The methodology is based on the swarming, and other variations have already been created. Each particle in PSO has its position and velocity initialized first. The position and velocity are then updated after each iteration using the mathematical Eq. (1.16) and Eq. (1.17).

$$s(t+1) = \left((h_1 \times c_6) \times \left(z^p(t) - z(t)\right)\right) + \left(h \times s(t)\right) + \left((h_2 \times c_7) \times \left(z^g(t) - z(t)\right)\right) \quad (1.16)$$

$$z(t+1) = z(t) + s(t+1) \quad (1.17)$$

Table 1.2 contains all of the PSO notations. They are essential in this situation. While a small value of h aids in local exploitation, a big value of h aids in exploration. Moreover, it influences the PSO's convergence behaviour. The social experience of that particle, as well as the social experience of every particle, is very important in this strategy. For PSO to be implemented successfully, there must be a proper relationship between the problem solutions and swarm particles. This method converges more quickly than previous global optimization

techniques[21]. Several PSO variations have a wide range of uses[22,23]. A hybrid FTS forecasting algorithm is built using PSO[24,25].

Table 1.2. Notations of the PSO method

Notation	Description
$s(t)$, $s(t+1)$	The velocity of the particle at (t) and (t + 1) iteration respectively
c_6, c_7	Two random numbers in the range [0,1]
$z(t)$,$z^p(t+1)$ and $z^g(t+1)$	Current position, personal best position, and global best position, respectively at (t) iteration
h_1 and h_2	Two positive constants (acceleration coefficients)
h	Inertia weight

1.3.3 Genetic Algorithm

Natural selection and genetics are the foundation for the GA search technique[26]. Genes are binary string representations of a solution (chromosomes) encoded by the GA. An allele is the value of a gene[26]. Genetic recombination (crossover), a low risk of mutation, and fitness proportionate selection are used in the GA to create good solutions[27]. The population is changed, and new generating solutions are assessed. Because of its competitive character, which permits the survival of solutions suited to favourable settings, GA is frequently used for complex and large-scale problems[28,29]. It shows promise for quickening convergence and improving the caliber of final solutions[30]. Because of GA's versatility, more study has been done to improve its performance by changing its processes, including initialization[30,31], and genetic (crossover and mutation) operators[32,33]. In n-dimensional GA-based problems, each chromosome or individual is represented by an n-bit binary sting. Here, the concepts of Mutations, Crossover, and other reproduction operators are applied to the problem to obtain optimal solutions.

1.4 Job Scheduling on Computational Grids

A computational grid is a large-scale and heterogeneous collection of autonomous systems. The sharing of computational jobs among the grid is one of the significant applications of the

grids. Several metaheuristic algorithms are developed to minimize the average completion time of jobs on each grid node through optimal job allocation. The field of job scheduling is complex since it calls for overcoming several efficiencies, resource utilization, and workload management challenges, as well as considering various resources, task dependencies, and confidentiality difficulties. Most scheduling problems for jobs are either NP-complete or NP-hard. As a result, compared to other options, it takes a very lengthy time to find an optimal solution[34]. To find a rapid yet efficient solution to this scheduling problem, most academics have been driven to explore an acceptable scheduling algorithm.

1.4.1 Need of Proposed Algorithms for Job Scheduling Problems

Heuristic and meta-heuristic algorithms are applied to obtain optimal or nearly optimal solutions to explain job scheduling because traditional algorithms frequently fall short of fully understanding conditions. Metaheuristic algorithms are the most effective means of escaping the local minima issue from which heuristic solutions typically suffer, as noted in[35]. To solve NP-complete problems, metaheuristic algorithms effectively search the search space for a sub- or near-optimal solution. The scheduling procedure assigns the tasks in the workflow to the appropriate resources based on predetermined scheduling criteria. Makespan is defined as the total amount of time needed to complete the workflow, considering both the time the tasks were completed and the time they were submitted.

1.5 Feature Selection

Classification problems often contain a large number of features (NF). Feature Selection involves selecting a subset of original features that can achieve high accuracy in a classification problem. However, only some of the features are helpful for classification-type problems. The features that are reductant and Irrelevant may reduce the classification accuracy. Assuming we choose every feature, the task becomes more complicated and time-consuming. The primary goals of the Feature Selection problems are to improve accuracy and reduce the NF. Several fields, including text mining, image processing, computer vision, industrial applications, bioinformatics, and others, use Feature Selection problems in various ways.

A typical feature selection algorithm typically consists of five basic steps. The process begins with an initialization step based on all the unique properties in the issue domain. After that, candidate feature subsets are created using a discovery technique. This process, which functions like a search mechanism, might begin with none, all, or a random subset of features. The best subset of features is found using a variety of search strategies, including both traditional and evolutionary ones. An evaluation function then enters the picture to determine how valuable the feature subset is. This function is crucial in directing the algorithm to the best subset. The crucial next phase, guided by predetermined criteria, is deciding when to stop the procedure. These standards could be based on the evaluation function or the search process. They could include requirements like achieving a predetermined number of chosen features or a predetermined number of iterations. Until the halting requirement is satisfied, the loop continues. The validity of the chosen subset is then established by instituting a validation method. Although it isn't a direct step in the feature selection procedure, it is a crucial step that guarantees the effectiveness of the selected algorithm. The chosen feature subset is validated against a test set, and the results are compared to those from earlier studies or from benchmarking methodologies that have been predetermined.

The three categories into which feature selection procedures are subdivided for evaluation purposes are wrapper approaches, embedding techniques, and filter approaches. In wrapper techniques, a learning mechanism is employed to evaluate the worth of selected feature subsets, most typically classification accuracy. Specifically, wrapper techniques repeatedly generate several candidate feature subsets by specified strategies, and then they use a classification algorithm to evaluate the corresponding classification accuracy. Embedded techniques always use a classification algorithm, even though the features are chosen during classifier training[36]. Filtering methods, in contrast, examine candidate subsets without using a classification system. The evaluation is primarily based on a dataset's inherent qualities. Since no learning is involved, the filter technique is the oldest and is thought to be the simplest of the three. On the other hand, because they take into account how the chosen features and the classification algorithm interact, the wrapper and embedding strategies typically provide higher classification results.

First and foremost, representation scalability is crucial. Real-world datasets frequently include various features, numbering thousands or even millions. Therefore, any suggested method must be scalable. This calls for including a reliable, scalable classifier in the algorithm's architecture to manage such big datasets effectively.

Second, the classifier selection significantly influences the quality of outcomes in a wrapper feature selection approach. Metaheuristic algorithms have been used to address feature selection issues with a variety of classifier types, including but not limited to K-nearest neighbour (KNN), Support Vector Machine (SVM), Naive Bayesian (NB), Random Forest (RF), Artificial Neural Network (ANN), Fuzzy rule-based (FR), and Decision Tree (DT).

Last but not least, the key step in a wrapper feature selection approach is choosing the best feature subset by maximizing a designated objective function. The particular categorization issue at hand will determine how this objective function behaves. Its function may include the elimination of unnecessary features or the improvement of classification precision. A multi-objective function is frequently developed to resolve competing objectives in the feature selection problem. This combines the two goals and streamlines the multi-objective function into a single goal using a weighted learning approach. The most favourable feature subset has been determined using such multi-objective functions in research on a large scale[37,38].

1.5.1 Need of Proposed Algorithms for Feature Selection Problems

Achieving the desired accuracy in time series forecasting has become a binding domain, and developing a forecasting framework with a high degree of accuracy is one of the most challenging tasks in this area. Combining different forecasting methods to construct efficient hybrid models regarding this challenge has been widely reported in the literature. Various hybrid models have been developed and successfully employed to improve forecasting accuracy. Despite the significant successes of hybrid models, efforts to access more accurate results face continued growth.

1.5.2 Benchmark Datasets

The proposed approaches for FS problems are assessed throughout this thesis using a variety of benchmark classification tasks with differing degrees of complexity. Table 1.3 provides a summary of the datasets. The UCI Repository of Machine Learning Databases provided these carefully selected datasets[39]. Different quantities of features, classes, instances, and data types (continuous and categorical) are present in the chosen datasets. The datasets serve as illustrative examples of the types of issues that the suggested algorithms can handle.

Table 1.3. Detailed information of all seven datasets.

Sl. No.	Datasets	Features	Instances
1	Wine	13	178
2	WDBC	30	569
3	Ionosphere	34	351
4	Sonar	60	208
5	Libras Movement	90	360
6	Hill Valley	100	606
7	Musk 1	166	476

1.6 Forecasting

Forecasting challenges have been addressed through numerous approaches, with a notable portion employing FYS or FYS-related methodologies. FTS approaches, Mamdani-type Fuzzy Inference Systems (FIS), and Sugeno-type FIS have all been widely utilized forecasting techniques. Song and Chissom[40] introduced the foundational definition of FTS. There have been more studies about FTS approaches in recent years. Kocak established an ARMA-style FTS forecasting technique. Güler Dincer and Akkuş concentrated on the fuzzification stage and proposed a robust clustering-based FTS approach. Based on fuzzy c-regression, Güler Dincer suggested an FTS approach. In the FTS approach, Bas et al. incorporated the pi-sigma neural network to identify fuzzy relations. A subtractive clustering technique and an algorithm for artificial bee colonies were employed in Zeng et al.'s[41] proposal for a method of FTS forecasting. Jiang et al. introduced an innovative forecasting methodology for wind speed data using a hybrid approach that integrated Multi-Objective Optimization (MOO) and FTS

techniques. A multivariate FTS technique that uses long short-term memory to define fuzzy relations was proposed by Tran et al. A combination strategy using a convolutional neural network and an FTS was proposed by Sadaei et al.[42]. Statistical inferences, establishing confidence intervals, and obtaining forecast distributions have been the main goals of some investigations on FTS. Yolcu et al. suggested using a statistical fuzzy inference method to derive statistical findings from Time Series Data (TSD) within an FS. Various nature-inspired optimization algorithms have applications in other fields[43,44] and can also be modified to obtain the optimal length of the intervals in FTS forecasting.

1.6.1 Need of Proposed Approach for Forecasting Problems

Intuitionistic Fuzzy Set (IFYS) can be regarded as an extended version of FYS, offering a more inclusive and flexible framework. The intuitionistic fuzzy c-means algorithm was utilized to introduce the modeling and implementation of intuitionistic FTS. Various Intuitionistic FTS approaches were proposed. Bisht and Kumar created hesitant FYSs using triangle membership functions with equal and unequal intervals. A high-order intuitionistic FTS technique was put out by Abhishekh[45]. Novel intuitionistic FTS definitions and a novel high-order intuitionistic FTS forecasting method were introduced by Egrioglu[46].

Picture Fuzzy Set (PFYS) is an upgraded version of IFYS that provides a more inclusive and flexible foundation. Thong and Son employed IFYS and picture fuzzy clustering (PFC) to make medical diagnoses. The concept of a PFYS set was incorporated into the clustering model by Thong and Son, who presented the idea of PFC. Son et al. provided a control theory application with an idea for a picture FIS. An analysis of the literature demonstrates that approaches for forecasting have been found extremely useful with more general set types. According to the literature, adding more inputs to the model with latent variables has improved inference outcomes. Membership values can be viewed as latent variables that give extra inputs for the models. This theory suggests that employing PFYSs in a forecasting model can be helpful.

1.7 Scope of the Thesis

The present thesis aims to propose different novel metaheuristic algorithms with critical applications in job scheduling, feature selection, and time series forecasting. Parameter adaptation of the proposed algorithms is another critical aspect addressed in the thesis. There is a considerable research gap as theoretical and mathematical analysis of most of the metaheuristic algorithms has yet to be proved or even discussed, so this issue has also been addressed in the form of stability, convergence, and signature. Multi-objective metaheuristic algorithms are developed to address the issue of job scheduling and feature selection. Integration of PFYS with an improved version of PSO is done to address the FTS forecasting problem.

1.8 Organization of the Thesis

This thesis comprises five parts, which consist of ten chapters.

Part- I has a single Chapter 1, which acts as an introductory section, motivating the research carried out. It includes an extensive literature review highlighting the significance of the issue addressed in the thesis.

Part – II of the thesis focuses on the practical implementation of the developed hybrid metaheuristic algorithms in the domain of Job Scheduling on a computational grid. This part consists of two chapters, namely Chapters 2 and 3.

Chapter 2 presents the application of fuzzy PSO with single-objective job scheduling on the computational grid. This chapter presents a specialized fuzzy PSO technique designed to tackle the job scheduling problem in computational grids. The central goal is to minimize the makespan value, indicating the maximum completion time of all tasks across the grid. The chapter systematically investigates the influence of trapezoidal and pentagonal fuzzy numbers on the optimization process and conducts a comparative analysis.

Chapter 3 provides a detailed exploration of utilizing fuzzy PSO for multi-objective job scheduling on the computational grid. This chapter builds upon the groundwork laid in the previous chapter, advancing from single-objective to MOO. The conflicting nature of makespan and flowtime as objective functions is addressed, necessitating a delicate balance between the two.

Part III of the thesis focuses on practically implementing the developed hybrid metaheuristic algorithms in the hybrid Feature Selection problems domain. This part consists of five chapters, namely Chapters 4, 5, 6, 7, and 8.

Chapter 4 introduces and applies a hybrid PSO algorithm to solve a Hybrid Feature Selection problem. This chapter provides a detailed mathematical explanation of the stability of the proposed HPSO algorithm, employing Von Neumann stability criterion and Fourier series concepts substantiated by rigorous proof. Furthermore, the convergence of the proposed HPSO algorithm is elucidated using the Markov chain concept. This chapter concludes with a comprehensive graphical and statistical comparison of results with other meta-heuristic algorithms, employing Friedman and Mann-Whitney U tests to assess the statistical significance of the proposed algorithm.

Chapter 5 comprehensively explores an enhanced version of the PSO algorithm, HPSO, and its application in multi-objective feature selection. This chapter marks a significant advancement from single-objective to MOO, building upon the framework established in the prior chapter. The effectiveness of HPSO is rigorously assessed across seven UCI datasets, with the Wilcoxon rank sum test employed for robust statistical analysis.

Chapter 6 introduces two innovative nature-inspired Hybrid optimization algorithms: PSOHHO and its variant PSOHHO-V. These algorithms incorporate the concepts of dual-swarm strategy and an exponential mutation operator (EMO) to amplify their exploration capabilities. PSOHHO and PSOHHO-V were evaluated based on statistical metrics and convergence rates, with extensive testing on ten benchmark functions. Additionally, the algorithms were applied to feature selection problems, and their effectiveness was benchmarked against other approaches using seven UCI datasets.

Chapter 7 unveils a novel Hybrid Swarm Optimization algorithm that integrates Pentagonal Fuzzy Numbers (PFN) and Trapezoidal Fuzzy Numbers (TFN). This innovative approach integrates PSO and Harris Hawk Optimization (HHO) principles to increase exploration opportunities. The mathematical foundations signature is introduced to obtain an idea of the optimization algorithm's intrinsic bias, and convergence is proved using the Markov Chain (MC) property. To evaluate the efficiency of PSOMHHO, it undergoes rigorous testing on benchmark functions and is compared against other established metaheuristic algorithms. Statistical significance is assessed using the Mann-Whitney U test and the Friedman test, affirming the prowess of PSOMHHO against various metaheuristic algorithms.

Chapter 8 introduces a ground-breaking hybrid optimization strategy termed Fuzzy PSOHHO, including its different variants. By melding the principles of Fuzziness, Escape Energy from Harris Hawk Optimization (HHO), and PSO, this technique is engineered to amplify exploration capabilities while maintaining a delicate equilibrium between exploration and exploitation. Recognizing the pivotal role of parameters in optimization, this chapter introduces the fuzzy concept for dynamic parameter adaptation within the framework of Fuzzy PSOHHO. The effectiveness of Fuzzy PSOHHO is meticulously assessed through extensive testing on benchmark functions and subsequent comparison with well-established metaheuristic algorithms. Rigorous statistical analysis employing the Kruskal-Wallis Test (KWT) decisively validates the superior performance of PSOMHHO over a range of metaheuristic algorithms.

Part – IV of the thesis focuses on the practical implementation of the developed hybrid metaheuristic algorithms in the domain of Forecasting. This part consists of one chapter, namely Chapter 9.

Chapter 9 presents a ground-breaking picture fuzzy time series (PFTS) forecasting model constructed based on the principles of picture fuzzy sets (PFYSs). This article presents a novel variant of the PSO (EDSPSO) algorithm, enhancing the PSO algorithm with the EMO and a dual-swarm strategy. This article integrates PFYS and EDSPSO to develop a novel hybrid EDSPSO-PFTS forecasting method. The suggested forecasting method is used on data sets from the Enrolments of Alabama University (UAE) and the State Bank of India Share Price

(SBISP) at the Bombay Stock Exchange, India, to demonstrate its applicability and usefulness. Mean square error (MSE) and average forecasting error (AFE) are used to gauge the effectiveness of the proposed method. The significant reduction in both MSE and AFE is solid evidence of the superior performance of the proposed EDSPSO-PFTS method compared to various existing methods. Rigorous statistical validation and performance analysis are conducted to ensure the reliability and validity of the proposed method.

Part V consists of a single Chapter 10, serving as the thesis summary. It highlights the main conclusions drawn from the research work and identifies the remaining challenges that require attention in this particular area of research in the future.

Part – II

Application of the Developed Hybrid Metaheuristic Algorithms in the domain of Job Scheduling on a Computational Grid.

(Chapter 2 and Chapter 3)

Chapter 2

Scheduling of Jobs on Computational Grids by Fuzzy Particle Swarm Optimization Algorithm using Trapezoidal and Pentagonal Fuzzy numbers

2.1 Introduction

A computational grid is a large-scale and heterogeneous collection of autonomous systems. The sharing of computational jobs among the grid is one of the significant applications of the grids. Its resources may be distributed among different owners, who may have some constraints and various access policies. Several metaheuristic methods are developed to minimize the average completion time of jobs on each Grid node through optimal job allocation[47]. A more complete analysis of the scheduling on the grid was provided by Dong and Akl, which is known as a N-P complete problem[48]. Every grid node has a processing speed and requirements of its own. So here we are using fuzzy PSO, a job scheduling problem on computational grids. Then, we will compare the TFN results with those obtained with PFN. This problem aims to minimize the time complexity and efficient use of grid nodes. The success of a PSO problem depends on the mapping between the PSO particle and the possible solution.

The rest of the chapter is organized as follows. Section 2.2 explains some basic concepts required to understand this chapter properly. Section 2.3 explains the problem we have tackled in this chapter. In Sections 2.4 and 2.5, the proposed algorithm and Numerical experiment are given, respectively, which fulfills our objective in this chapter. In Section 2.6, the conclusion of the above approach is presented.

2.2 Background

For a proper understanding of the work explained in this chapter, the concepts of fuzzy numbers, TFN, and PFN are required, which are explained in Section 1.3. Here, the particles are guided using the concepts of PSO, which are also described in Section 1.3.

2.3 Problem Formulation

Here, in this problem on computational grids, there is generally a framework focusing on the interaction between the grid information server, the grid's resource broker, and the domain resource manager. In this section, the problem of this chapter is explained. The scheduling of jobs on the grids using fuzzy PSO is explained in the computational grid environment. For our proper understanding, some important terms and concepts are defined. They are as follows-

Scheduling Problem

The schedule is a function from jobs to the specific intervals of time of the grid node. A scheduling problem is defined as the jobs allocated to the machines with optimal criteria. In this chapter, the scheduling problem is defined as the allocation of jobs to specific computational grids with optimal criteria. Here optimal criteria is the maximum number of iterations allowed.

Grid Nodes

A grid node is a computational resource whose capacity is limited. It can be a computer lab, workstation, personal computer, or a collection of computers at a specific location. The computational capacity of the grid node depends on the amount of memory, number of Central processing units, basic storage space, and other types of specifications. Every Grid node has a processing speed of its own which is expressed as the number of cycles per unit time.

Makespan

In Job scheduling problems, makespan can be defined as the maximum of all the completion time. First jobs are assigned to grids, then we will compute the time taken by each grid to complete all the jobs that are assigned. Then we will take the maximum of all the completion

time of all the grids. Mathematically speaking if $d(i,j)$ is the completion time, In other words, $d(i,j)$ is the time taken by the grid node $G(i)$ to finish the job $J(j)$. The time taken by the grid node to execute all the jobs allocated to that grid node only is represented by $\sum d(i)$. Now max $\sum d(i)$ is called makespan.

Jobs

Job is a collection of operations or a single operation allocated to the computational grid. Now we are going to explain the concerned problem. Now $J(j)$ means Job on the machine j and $G(i)$ means Grid at the node i. Now let us consider jobs $J(j), j \in (1,2,\cdots,b)$ that are independent on Grid nodes $G(i), i \in (1,2,\cdots,a)$. This problem aims to minimize the time complexity and efficient use of grid nodes.

Now we define $d(i,j)$ as the completion time, In other words, the time taken by the grid node $G(i)$ to finish the job $J(j)$. The time taken by the grid node to execute all the jobs allocated to that grid node only is represented by $\sum \mathrm{d}(i)$. Now max $\{\sum \mathrm{d}(i)\}$ is called makespan. $\sum_{i=1}^{a}(\sum \mathrm{d}(i))$ is called the flowtime. These concepts are used while applying the fuzzy PSO algorithm. The objective of this chapter is to minimize the makespan value. We have to optimize a job scheduling that minimizes the makespan value. That is to minimize the maximum time all Grids take to complete all the assigned jobs. And then to see the effects of TFN and PFN on it.

2.4 Proposed Algorithm (Fuzzy PSO)

In this section, the fuzzy PSO algorithm is explained in detail. Here in the scheduling of jobs on the computational Grid environment using PSO, the position and velocities of particles are taken in the form of fuzzy matrices. In this section, it is explained how fuzzy PSO is used for solving problems in the scheduling of jobs on the computational grid nodes. Then their results are compared for fuzzy PSO with the TFN and PFN. To successfully apply PSO, one of the factors is to find the map between the problem solution and the PSO particle. The performance and feasibility are directly affected by it.

Suppose $G = \{G(1), G(2), \cdots, G(a)\}$, $J = \{J(1), J(2), \cdots, J(b)\}$ are the grid nodes and jobs respectively. The number of Grids and Jobs are a and b respectively. Let the position of the particle be defined as

$$Z = \begin{bmatrix} z(1,1) & \cdots & z(1,b) \\ \vdots & \ddots & \vdots \\ z(a,1) & \cdots & z(a,b) \end{bmatrix}$$

The elements of Z must satisfy the following criteria-

$z(i,j) \in [0,1]$, $\quad i \in \{1,2,\dots,a\}$ and $j \in \{1,2,\dots,b\}$

$\sum_{i=1}^{a} z(i,j) = 1 \quad i \in \{1,2,\dots,a\}$ and $j \in \{1,2,\dots,b\}$.

Similarly, the velocity of the particle is defined as

$$S = \begin{bmatrix} s(1,1) & \cdots & s(1,b) \\ \vdots & \ddots & \vdots \\ s(a,1) & \cdots & s(a,b) \end{bmatrix}$$

The normalization of the matrix Z is as follows-

$$(\| Z \|) = \begin{bmatrix} z(1,1) \Big/ \sum_{i=1}^{a} z(i,1) & \cdots & z(1,b) \Big/ \sum_{i=1}^{a} z(i,b) \\ \vdots & \ddots & \vdots \\ z(a,1) \Big/ \sum_{i=1}^{a} z(i,1) & \cdots & z(a,b) \Big/ \sum_{i=1}^{a} z(i,b) \end{bmatrix}$$

Before going into detail about the fuzzy PSO algorithm, let us see some of the notations required on the way, they are as follows-

α_1 =Collection of all the jobs to be processed.

α_2 = Collection of all the jobs that are being scheduled

α_3 = Collection of all the jobs after job allocation is already completed.

α_4 = Collection of all the available grid nodes.

α_5 = Collection of all the grid nodes that have already been allocated to the jobs.

α_6 = Collection of all the grid nodes that are available or free.

Pseudocode:

The Pseudocode of the fuzzy PSO algorithm is explained as follows:

Step 1

When the nodes are active and no new jobs are available, then we have to wait for the jobs that are new or update α_4 and α_1.

Step 2

At $t = 0$, If $\alpha_4 = 0$, wait for new grids to be available. If $\alpha_2 < \alpha_4$, then jobs are allocated on the principle called first come first serve basis. If $\alpha_1 > \alpha_4$, job allocation is given in Step 3.

Step 3

Now we have to initialize all the parameters of the particle swarm. The size of the particle swarm (N) depends on the experiment and its value is given before the start of the algorithm. The values of the parameters are initialized first.

3.1 Now, we have to initialize the position for each particle. So we have taken random matrices which will be treated as the position of the particles. Then the matrices are normalized.

3.2 $t = t + 1$ (Here we will start the iteration process from $t = 1$ to the maximum iteration)

3.2.1 Then the makespan value is calculated for each particle.

3.2.2 The latest best solution is calculated as follows-

$$y'' = argmin\,(f(y''(t-1)), f(y(1)(t)), f(y(2)(t)), \ldots, f(y(b)(t)))$$

3.2.3 For each particle, the personal best solution is computed as follows-

$$y'(t) = argmin\,(f(y'(t-1)), f(y(t)))$$

3.2.4 Take random velocity as a trapezoidal matrix for the first Case.

3.2.5 Take random velocity as a pentagonal matrix for the second Case.

3.2.6 Now update each particle using Eqns. (1.16) and (1.17).

3.2.7 Now for each particle, the position matrix is normalized.

3.3 The iteration process is continued until the optimality criteria are normalized.

Step 4

Repeat the process as long as the grid is active.

2.5 Experiments

Now, we have taken some parameters required to solve the problem. They have an Inertia weight of 0.8. Acceleration coefficients are as follows: 2 and 1.3, respectively. The two random numbers are generated automatically. Here, '1' represents the Job assigned to the Grid, and '0' means no Job assigned to the Grid. Total number of particles (N)=20.

2.5.1 Experiment 1

Here we are taking two Grid Nodes and the Number of Jobs are three. The speeds of the two grid nodes are 4 and 4.1 respectively. The time required for each jobs are as follows 1119, 1112, and 1811 respectively.

For the first case, we have taken a velocity matrix with each element as a TFN. Here we have observed that Job 1 is scheduled on Grid 2, Job 2 is scheduled on Grid 1, and Job 3 is scheduled on Grid 2. For the Second case, we have taken the velocity matrix with each element as a PFN. Here we have observed that Job 1 is scheduled on Grid 2, Job 2 is scheduled on Grid 2, and Job 3 is scheduled on Grid 1.

2.5.2 Experiment 2

Here we are taking three Grid Nodes and Number of Jobs are seven.

Optimal Schedule with TFN

Table 2.1. Optimal Schedule with TFN.

	$J(1)$	$J(2)$	$J(3)$	$J(4)$	$J(5)$	$J(6)$	$J(7)$
G (1)	0	0	0	0	0	1	1
G (2)	0	0	1	0	0	0	0
G (3)	1	1	0	1	1	0	0

Here the Grid speeds are as follows – 17, 34, 13 and the time required for each job is as follows: 45, 103, 80, 62, 91, 113, 88 respectively. Above Table 2.1 is the Optimal Schedule. Here Job 1 is scheduled on Grid 3, Job 2 is scheduled on Grid 3, Job 3 is scheduled on Grid 2, Job 4 is

scheduled on Grid 3, Job 5 is scheduled on Grid 3, Job 6 is scheduled on Grid 1, Job 7 is scheduled on Grid 1.

Optimal Schedule with PFN

Table 2.2. Optimal Schedule with PFN.

	$J(1)$	$J(2)$	$J(3)$	$J(4)$	$J(5)$	$J(6)$	$J(7)$
G (1)	0	1	0	1	0	0	1
G (2)	0	0	0	0	1	1	0
G (3)	1	0	1	0	0	0	0

Here the Grid speeds are as follows – 28, 9, 23 and the time required for each job is as following, 124, 71, 132, 99, 83, 78, 64 respectively. Above Table 2.2 is the Optimal Schedule. Here Job 1 is scheduled on Grid 3, Job 2 is scheduled on Grid 1, Job 3 is scheduled on Grid 3, Job 4 is scheduled on Grid 1, Job 5 is scheduled on Grid 2, Job 6 is scheduled on Grid 2, Job is scheduled on Grid 1.

2.5.3 Experiment 3

Here we are taking four Grid Nodes and Number of Jobs are twelve.

Optimal Schedule with TFN

Table 2.3. Optimal Schedule with TFN.

	$J(1)$	$J(2)$	$J(3)$	$J(4)$	$J(5)$	$J(6)$	$J(7)$	$J(8)$	$J(9)$	$J(10)$	$J(11)$	$J(12)$
G (1)	0	0	0	0	1	0	0	1	0	0	0	0
G (2)	0	0	0	0	0	1	0	0	0	0	1	0
G (3)	0	0	1	1	0	0	0	0	0	1	0	0
G (4)	1	1	0	0	0	0	1	0	1	0	0	1

Here the Grid speeds are as follows – 44, 3, 11, 23 and the time required for each Job is as follows: 144, 119, 68, 51, 9, 112, 77, 30, 65, 26, 113, 56 respectively. Above Table 2.3 is the Optimal Schedule. Here Job 1 is scheduled on Grid 4, Job 2 is scheduled on Grid 4, Job 3 is scheduled on Grid 3, Job 4 is scheduled on Grid 3, Job 5 is scheduled on Grid 1, Job 6 is scheduled on Grid 2, Job 7 is scheduled on Grid 4, Job 8 is scheduled on Grid 1, Job 9 is

scheduled on Grid 4, Job 10 is scheduled on Grid 3, Job 11 is scheduled on Grid 2, Job 12 is scheduled on Grid 4.

Optimal Schedule with PFN

Table 2.4. Optimal Schedule with PFN.

	$J(1)$	$J(2)$	$J(3)$	$J(4)$	$J(5)$	$J(6)$	$J(7)$	$J(8)$	$J(9)$	$J(10)$	$J(11)$	$J(12)$
$G\ (1)$	0	0	0	0	0	1	0	1	0	0	0	1
$G\ (2)$	0	0	0	0	0	0	0	0	1	0	1	0
$G\ (3)$	1	0	0	1	0	0	1	0	0	1	0	0
$G\ (4)$	0	1	1	0	1	0	0	0	0	0	0	0

Here the Grid speeds are as follows – 50, 21, 45, and 10. The time required for each job is as following-101, 74, 71, 58, 23, 123, 17, 123, 124, 41, 15, 84 respectively. Above Table 2.4 is the Optimal Schedule. Here Job 1 is scheduled on Grid 3, Job 2 is scheduled on Grid 4, Job 3 is scheduled on Grid 4, Job 4 is scheduled on Grid 3, Job 5 is scheduled on Grid 4, Job 6 is scheduled on Grid 1, Job 7 is scheduled on Grid 3, Job 8 is scheduled on Grid 1, Job 9 is scheduled on Grid 2, Job 10 is scheduled on Grid 3, Job 11 is scheduled on Grid 2, Job 12 is scheduled on Grid 1.

2.5.4 Experiment 4

Here we are taking five Grid Nodes and Number of Jobs are twenty.

Optimal Schedule with TFN

Here the Grid Speed are as follows – 4, 34, 47, 24, 19 and the time required for each job is as following-13, 82, 140, 105, 124, 129, 17, 106, 83, 79, 48, 25, 48, 101, 79, 92, 115, 25, 75, 115 respectively. Above Table 2.5 is the Optimal Schedule. Here Job 1, Job 2, Job 3, Job 4, Job 5, Job 6, Job 7, Job 8, Job 9, Job 10, Job 11, Job 12, Job 13, Job 14, Job 15, Job 16, Job 17, Job 18, Job 19 and Job 20 are scheduled on Grid 4, Grid 3, Grid 5, Grid 1, Grid 4, Grid 3, Grid 4, Grid 2, Grid 2, Grid 1, Grid 3, Grid 5, Grid 1, Grid 3, Grid 5, Grid 5, Grid 1, Grid 3, Grid 4 and Grid 1 respectively.

Table 2.5. Optimal Schedule with TFN.

	$J(1)$	$J(2)$	$J(3)$	$J(4)$	$J(5)$	$J(6)$	$J(7)$	$J(8)$	$J(9)$	$J(10)$	$J(11)$	$J(12)$	$J(13)$	$J(14)$	$J(15)$	$J(16)$	$J(17)$	$J(18)$	$J(19)$	$J(20)$
$G(1)$	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	1	0	0	1
$G(2)$	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
$G(3)$	0	1	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	1	0	0
$G(4)$	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0
$G(5)$	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0

Optimal Schedule with PFN

Here the Grid Speed are as follows, 19, 12, 8, 36, 15 and the time required for each job is as following, 86, 16, 142, 55, 96, 145, 81, 5, 51, 13, 119, 61, 20, 107, 100, 42, 75, 112, 71, 134 respectively. Above Table 2.6 is the Optimal Schedule. Here Job 1, Job 2, Job 3, Job 4, Job 5, Job 6, Job 7, Job 8, Job 9, Job 10, Job 11, Job 12, Job 13, Job 14, Job 15, Job 16, Job 17, Job 18, Job 19 and Job 20 are scheduled on Grid 4, Grid 1, Grid 2, Grid 5, Grid 3, Grid 5, Grid 3, Grid 3, Grid 4, Grid 2, Grid 4, Grid 1, Grid 5, Grid 3, Grid 3, Grid 5, Grid 4, Grid 5, Grid 1 and Grid 5 respectively.

Table 2.6. Optimal Schedule with PFN.

	$J(1)$	$J(2)$	$J(3)$	$J(4)$	$J(5)$	$J(6)$	$J(7)$	$J(8)$	$J(9)$	$J(10)$	$J(11)$	$J(12)$	$J(13)$	$J(14)$	$J(15)$	$J(16)$	$J(17)$	$J(18)$	$J(19)$	$J(20)$
$G(1)$	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0
$G(2)$	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
$G(3)$	0	0	0	0	1	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0
$G(4)$	1	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0	0
$G(5)$	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	1	0	1	0	1

Then we have taken more examples with more grid nodes and jobs, i.e., 10 Grid nodes and 50 Jobs, 40 Grid nodes and 100 Jobs. We are getting similar results. Here, we observe that with an increase in the number of iterations, the makespan value decreases, and after some iterations, it remains more or less constant. Here, the termination criteria are maximum iteration. Here, the optimal Criteria is the makespan value. We have to optimize a job scheduling that minimizes the makespan value. That is to minimize the maximum time all Grids

take to complete all the assigned jobs. Here, we can see the makespan value of fuzzy PSO using TFN and the makespan value of fuzzy PSO using PFN, which remains the same. We can also see that the global best position is the same for both approaches.

2.6 Conclusion

This chapter tackles the task of scheduling jobs on a computational grid through fuzzy PSO, utilizing both TFN and PFN. The algorithm's performance is analysed by first employing TFN and subsequently with PFN. The outcomes between the two approaches demonstrate a remarkable consistency. This opens opportunities for future exploration where diverse fuzzy numbers could be incorporated to recognize any impacts on results.

Chapter 3

Job Scheduling On Computational Grids Using Multi-Objective Fuzzy Particle Swarm Optimization

3.1 Introduction

This chapter provides a detailed exploration of utilizing fuzzy PSO for multi-objective job scheduling on the computational grid. This chapter builds upon the groundwork laid in the previous chapter, advancing from single-objective to MOO. The conflicting nature of makespan and flowtime as objective functions is addressed, necessitating a delicate balance between the two. So here we are comparing the results obtained using TFN with those obtained using PFN. The objective of this problem is to use the grid nodes efficiently. Here, we are using MOO, an area of multiple criteria decision-making. It is useful when multiple objective functions must be simultaneously optimized. The mapping between a PSO particle and a potential solution determines the outcome of a PSO problem. This method is an innovative approach, and to my knowledge, there is no such existing method.

The remaining portion of the chapter is structured as follows. Specific fundamental ideas necessary for understanding this chapter properly are explained in Section 3.2. The problem we have addressed in this chapter and our goal are described in Section 3.3. The proposed fuzzy PSO algorithm that was applied to solve the multi-objective job scheduling problem is described in Section 3.4. The proposed algorithm's numerical experiment is presented in Section 3.5. The proposed algorithm's conclusion is given in Section 3.6.

3.2 Background

A comprehensive understanding of the content in this chapter necessitates familiarity with key concepts such as fuzzy number, TFN, and PFN, explained in Section 1.3. The guidance of particles through the grids is drawn from the principles of PSO, which is also explained in Section 1.3. Here the concept of MOO is also required which is explained below.

MOO

MOO is used when more than one objective function has to be optimized. This process becomes very important when the objective functions are conflicting in nature. That is minimizing some objective functions that maximize some other objective functions[49,50].

3.3 Problem Formulation

This section illuminates the focal problem addressed in the chapter, specifically exploring the complexities of multi-objective job scheduling using fuzzy PSO within the computational grid environment. To facilitate a comprehensive grasp, essential terms and concepts are meticulously expounded.

Flowtime

In Job scheduling problems, flowtime can be defined as the sum of all the completion time. After jobs are assigned to grids, the time taken by each grid to complete all the jobs that are assigned is computed. Then we will take the sum of all the completion time of all the grids. Mathematically speaking if $d(i,j)$ is the completion time, In other words $d(i,j)$ is the time taken by the grid node $G(i)$ to finish the job (j) . The time taken by the grid node to execute all the jobs allocated to that grid node only is represented by $\sum d(i)$. Now is $\sum_{i=1}^{a}(\sum d(i))$ called flowtime, where 'a' is the number of grids in the problem.

Our formulated problem involves the simultaneous consideration of two conflicting objective functions: makespan and flowtime. Minimizing flowtime requires the swift completion of average jobs, compromising the duration of the longest job. On the contrary, minimizing makespan ensures that no job experiences excessive duration, although at the cost

of most jobs enduring extended periods. The maximization of makespan invariably leads to the minimization of flowtime, and vice versa. The objective of the proposed algorithm is to minimize the makespan value and flowtime. That is to minimize the maximum time taken by all Grids to complete all the jobs assigned to them according to the first objective function. And to minimize the sum of all time taken by all Grids to complete all the jobs assigned to them according to the second objective function. We have to optimize a job scheduling that minimizes the makespan value and flowtime. This inherent contradiction is effectively addressed through the application of the proposed fuzzy PSO algorithm.

Now we are going to explain the concerned problem. The basic meaning of $J(j)$ and $G(i)$ are already explained in the previous chapter in Section 2.3. Now let us consider jobs $J(j)$, $j \in (1,2,\cdots,b)$ that are independent on Grid nodes $G(i), i \in (1,2,\cdots,a)$. The objective of this chapter is to efficiently use the grid nodes by minimizing the makespan value and total flowtime values. Now we define $d(i,j)$ as the completion time. In other words time taken by Grid node $G(i)$ to finish the Job $J(j)$. The time taken by the grid node to execute all the jobs allocated to that grid node only is represented by $\sum d(i)$. Now max $\{\sum d(i)\}$ is called makespan. $\sum_{i=1}^{a}(\Sigma \mathrm{d}(i))$ is called as the flowtime. These concepts are used while applying the fuzzy PSO algorithm.

3.4 Proposed Algorithm

In this section, the proposed multi-objective fuzzy PSO algorithm is explained. Here in the scheduling of jobs on the computational Grid environment using PSO, the position and velocities of particles are taken in the form of fuzzy matrices. In this section, it is explained how fuzzy PSO is used for solving multi-objective job scheduling on the computational grid nodes. Then their results are compared for fuzzy PSO with TFN and fuzzy PSO with PFN. To successfully apply PSO, one of the key factors is to find the map between the problem solution and the PSO particle. The performance and feasibility are directly affected by it. Suppose $G = \{G(1), G(2), \cdots, G(a)\}$, $J = \{J(1), J(2), \cdots, J(b)\}$ are the Grid nodes and Jobs respectively. The number of Grids and Jobs are a and b respectively. Let the position and the velocity of the particle be defined as

$$Z = \begin{bmatrix} z(1,1) & \cdots & z(1,b) \\ \vdots & \ddots & \vdots \\ z(a,1) & \cdots & z(a,b) \end{bmatrix} \text{and } S = \begin{bmatrix} s(1,1) & \cdots & s(1,b) \\ \vdots & \ddots & \vdots \\ s(a,1) & \cdots & s(a,b) \end{bmatrix} \text{ respectively}$$

The normalization of the matrix Z is as follows-

$$(\| Y \|) = \begin{bmatrix} z(1,1) \Big/ \sum_{i=1}^{a} z(i,1) & \cdots & z(1,b) \Big/ \sum_{i=1}^{a} z(i,b) \\ \vdots & \ddots & \vdots \\ z(a,1) \Big/ \sum_{i=1}^{a} z(i,1) & \cdots & z(a,b) \Big/ \sum_{i=1}^{a} z(i,b) \end{bmatrix}$$

Before going into detail about the fuzzy PSO algorithm, The notational meaning of notation $\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5$ and α_6 are already discussed in previous chapter in Section 2.4.

Step 1

When the nodes are active and no new jobs are available, then we have to wait for the jobs that are new or update α_4 and α_1 .

Step 2

At $t = 0$, If $\alpha_4 = 0$, wait for new grids to be available. If $\alpha_2 < \alpha_4$, then jobs are allocated on the principle called first come first serve basis. If $\alpha_1 > \alpha_4$, job allocation as given in Step 3.

Step 3

Now we have to initialize all the parameters of the particle swarm. The size of the particle swarm (N) depends on the experiment and its value is given before the start of the algorithm. The values of the parameters are initialized.

3.1 Now we have to initialize the position for each particle. So from here a population Set has been initialized. So we have taken random matrices which will be treated as the position of the particles. Then the matrices are normalized.

3.2 Take random velocity as a trapezoidal matrix, i.e., every matrix element is a TFN for the first Case.

3.3 Take random velocity as a pentagonal matrix, i.e., every matrix element is a PFN for the second Case.

3.4 $t = t + 1$ (Here we will start the iteration process from t=1 to the maximum iteration, which can be changed in the programming code depending on the requirement of the coder)

3.4.1 A leader set is selected from the population Set.

3.4.2 Each particle's velocity and position are updated using Eqns. (1.16) and (1.17).

3.4.3 The makespan value and flowtime value of each particle are calculated.

3.4.4 Now update the personal best and Global Best of each particle, respectively.

3.4.5 Add Non-Dominated Particles to Non-dominating Front.

3.4.6 Determine Domination of New Non-dominating Front Members.

3.4.7 Keep only Non-Dominated Members in the Non-dominating Front.

3.4.8 For each particle, the position matrix is normalized.

3.4.9 The iteration process is continued until the Maximum iteration is achieved.

3.5 The iteration process is continued until the Maximum iteration is achieved.

Step 4

Repeat the process as long as the grid is active.

3.5 Experiments

Now, we have taken some parameters required to solve the problem. They are Inertia weight $(h) = 0.8$. Acceleration coefficients h_1 and h_2 are as follows 2 and 1.3 respectively. The two random numbers are generated automatically. Here, the total number of particles we have taken is 20. We have taken a velocity matrix for the first case with each element as a TFN. For the Second case, we have taken the velocity matrix with each element as a PFN. In the optimal schedule Tables 3.2, 3.4, 3.6, and 3.8, we use a technique in which grids and jobs are represented row-wise and column-wise, respectively, and then '1' represents the job assigned to the Grid, and '0' represents no job assigned to the Grid[51]. In Figs 3.1, 3.2, 3.3 and 3.4 Red coloured points are DMs and joining all the dominant points will form a Pareto-optimal curve. Other points present in the graph are dominated points.

3.5.1 Experiment 1

Here, we are taking three Grid Nodes, and the Number of Jobs is seven.

Optimal Schedule with TFN

Here the Grid Speed that is taken in this problem is as follows, 19.09, 27.017, 29.45, and the time required for each job is as follows, 69.73, 115.73, 19.34, 89.86, 128.99, 99.90, 82.96 respectively. In Table 3.1, all the DMs are shown with their makespan value and flowtime value. Here, in this experiment, there are 9 DMs out of which 8 DMs are shown in Fig 3.1.

Table 3.1 All DMs.

DMs	Makespan	Flowtime Value
1	10.638435	3.108031
2	8.455468	3.190418
3	4.774187	89.732866
4	13.546865	2.933361
5	19.043898	1.848561
6	13.737084	2.741747
7	38.040034	1.736883
8	8.715420	3.128136
9	6.932100	5.508100

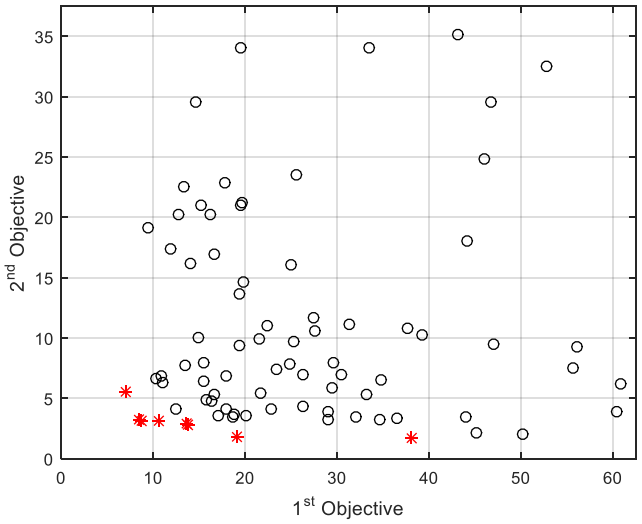

Fig. 3.1 Collection of all 8 DMs of Experiment 1 using TFN.

Table 3.2. Optimal Schedule of DM 2.

	$J(1)$	$J(2)$	$J(3)$	$J(4)$	$J(5)$	$J(6)$	$J(7)$
$G\ (1)$	1	1	0	0	0	1	0
$G\ (2)$	0	0	0	1	1	0	0
$G\ (3)$	0	0	1	0	0	0	1

Optimal Schedule with PFN

Here, the Grid Speed is as follows – 47.66, 8.08, 39.89, and the time required for each job is as follows- 28.13, 17.84, 64.98, 135.90, 92.86, 14.33, 24.04, respectively. Table 3.3 shows all the DMs with their makespan and flowtime values. Here, in this experiment, there are 7 DMs out of which 5 DMs are shown in Fig 3.2.

Table 3.3 All DMs

DMs	Makespan Value	Flowtime Value
1	5.853997	138.489278987881
2	8.581198	3.577729
3	8.120252	3.845658
4	10.182115	2.294112
5	7.289982	11.857017
6	22.652308	2.087883
7	115.157452	1.435459

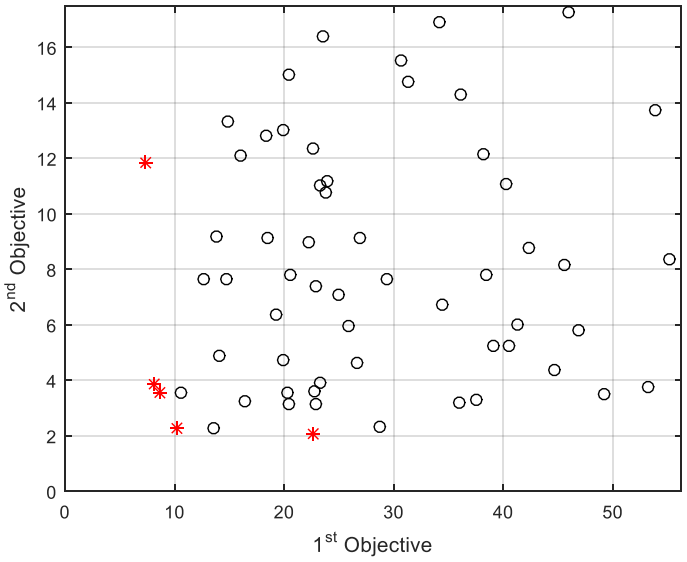

Fig. 3.2 Collection of all 5 DMs of Experiment 1 using PFN.

Table 3.4 Optimal Schedule of DM 4

	$J(1)$	$J(2)$	$J(3)$	$J(4)$	$J(5)$	$J(6)$	$J(7)$
$G\ (1)$	1	0	0	0	1	0	0
$G\ (2)$	0	1	0	1	0	1	0
$G\ (3)$	0	0	1	0	0	0	1

Above Table 3.4 is the Optimal Schedule. Here, Job 1 is assigned to Grid 1, Job 2 is assigned to Grid 2, Job 3 is assigned to Grid 3, Job 4 is assigned to Grid 2, Job 5 is assigned to Grid 1, Job 6 is assigned to Grid 2, Job 7 is assigned to Grid 3. With an increase in the number of iterations, we get particles whose makespan and flowtime values are minimized until we reach maximum iteration.

3.5.2 Experiment 2

Here, we are taking four Grid Nodes, and the Number of Jobs is nineteen.

Optimal Schedule with TFN

Here the Grid Speed are as follows – 23.82, 35.64, 28.83, 5.67, and the time required for each job is as follows, 119.61, 65.37, 81.28, 74.86, 36.39, 132.58, 141.25, 7.02, 108.82, 52.76, 17.62, 147.10, 13.067, 22.19, 73.99, 30.33, 104.16, 31.83 respectively. In Table 3.5, all the DMs are shown with their makespan value and flowtime value. In this experiment, there are 10 DMs which are shown in Fig 3.3.

Table 3.5 All DMs

DMs	Makespan Value	Flowtime Value
1	6.946936	16.739297
2	14.659486	2.563739
3	6.969654	7.014856
4	9.027537	5.088222
5	9.526646	3.943752
6	31.139973	2.008530
7	7.816608	5.145798
8	19.697327	2.253105
9	44.616472	1.929807
10	13.305884	3.293277

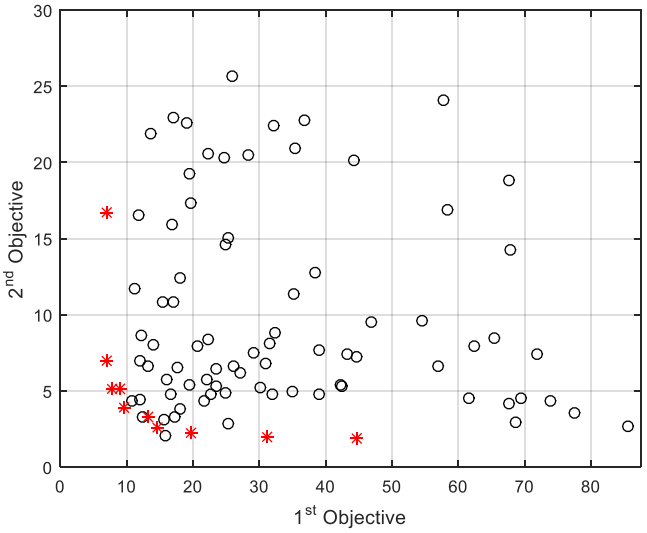

Fig. 3.3 Collection of all DMs of Experiment 2 using TFN.

Table 3.6 DM 5

	$J(1)$	$J(2)$	$J(3)$	$J(4)$	$J(5)$	$J(6)$	$J(7)$	$J(8)$	$J(9)$	$J(10)$	$J(11)$	$J(12)$	$J(13)$	$J(14)$	$J(15)$	$J(16)$	$J(17)$	$J(18)$	$J(19)$	$J(20)$
$G(1)$	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	1	0	0	0	0
$G(2)$	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	1	0
$G(3)$	0	1	1	0	1	0	0	0	1	1	0	0	0	0	1	0	0	0	0	0
$G(4)$	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1	1	0	1

Above Table 3.6 is the Optimal Schedule. Here, Job 1 is assigned to Grid 4, Job 2 is assigned to Grid 3, Job 3 is scheduled to Grid 2, Job 4 is scheduled to Grid 4, Job 5 is scheduled to Grid 3, job 6 is assigned to Grid 1, job 7 is assigned to Grid 1, job 8 is assigned to Grid 4, job 9 is assigned to Grid 3, job 10 is assigned to Grid 3, job 11 is assigned to Grid 2, job 12 is assigned to Grid 2, job 13 is assigned to Grid 2, job 14 is assigned to Grid 1, job 15 is assigned to Grid 3, job 16 is assigned to Grid 1, job 17 is assigned to Grid 4, job 18 is assigned to Grid 4, job 19 is assigned to Grid 2. With an increase in the number of iterations, we get particles whose makespan and flowtime values are minimized until we reach maximum iteration.

Optimal Schedule with PFN

Here the Grid Speed is as follows – 40.11, 35.88, 47.73, 47.88, and the time required for each job is as follows- 121.7649, 50.02, 19.94, 125.62, 55.22, 130.92, 54.20, 95.65, 57.75, 26.74,

32.62, 134.42, 74.42, 37.24, 42.13, 35.81, 11.42, 66.08 respectively. Table 3.7 shows all the DMs with their makespan and flowtime values. Here, in this experiment, there are 8 DMs which are shown in Fig 3.4.

Table 3.7 All DMs

DMs	Makespan Value	Flowtime Value
1	16.261141	1.975759
2	43.342028	1.613511
3	57.514148	1.479592
4	6.563975	10.077199
5	7.711173	2.009736
6	21.738393	1.736575
7	25.865067	1.633693
8	7.411354	4.348766

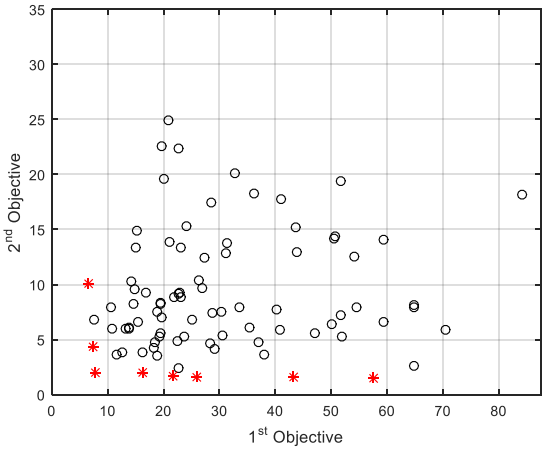

Fig. 3.4 Collection of all DMs of Experiment 2 using PFN.

Table 3.8 DM 5

	$J(1)$	$J(2)$	$J(3)$	$J(4)$	$J(5)$	$J(6)$	$J(7)$	$J(8)$	$J(9)$	$J(10)$	$J(11)$	$J(12)$	$J(13)$	$J(14)$	$J(15)$	$J(16)$	$J(17)$	$J(18)$	$J(19)$
$G(1)$	0	0	0	0	1	0	1	0	0	1	0	0	0	1	1	0	0	0	0
$G(2)$	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0
$G(3)$	0	1	0	0	0	1	0	0	1	0	0	1	1	0	0	0	1	1	1
$G(4)$	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Above Table 3.8 is the Optimal Schedule. Here job 1 is assigned to Grid 4, job 2 is assigned to Grid 3, job 3 is assigned to Grid 2, job 4 is assigned to Grid 2, job 5 is assigned to Grid 1, job 6 is assigned to Grid 3, job 7 is assigned to Grid 1, job 8 is assigned to Grid 2, job 9 is is scheduled on Grid 3, job 10 is assigned to Grid 1, job 11 is assigned to Grid 4, job 12 is assigned to Grid 3, job 13 is assigned to Grid 3, job 14 is assigned to Grid 1, job 15 is assigned to Grid 1, job 16 is assigned to Grid 2, job 17 is assigned to Grid 3, job 18 is assigned to Grid 3, job 19 is assigned to Grid 3. With an increase in the number of iterations, we get particles whose makespan and flowtime values are minimized until we reach maximum iteration. Then we have taken more examples with more grid nodes and jobs, i.e., 10 Grid nodes and 50 jobs, 40 Grid nodes and 100 jobs. We are getting similar results.

3.6 Conclusion

This chapter navigates the complicated multi-objective job scheduling on a computational grid by applying fuzzy PSO with trapezoidal and PFN. The optimal criteria for optimization are the makespan and flowtime values. Here, we have taken a particle among the set of DMs that simultaneously minimizes both makespan and flowtime values for both cases. The consistency in results is validated through experimentation, including 10 Grid Nodes and 50 Jobs and 40 Grid Nodes with 100 Jobs. The work reveals that the objective values exhibit remarkable similarity despite distinct scheduling arrangements. Consequently, the objective values of fuzzy PSO utilizing trapezoidal and PFN are thoroughly calculated and compared, demonstrating the equivalence in results. We can take other fuzzy numbers in this process and compare the results for future work.

Part – III

Application of the developed hybrid metaheuristic algorithms on Hybrid Feature Selection Problems.

(Chapter 4, Chapter 5, Chapter 6, Chapter 7, and Chapter 8)

Chapter 4

Hybrid Particle Swarm Optimization for a Feature Selection Problem with Stability Analysis

4.1 Introduction

Classification problems often contain a large NF. Feature selection (FS) is the process of choosing features that will be the subset of the relevant features, and it will increase the classification accuracy (CA) and decrease the NF. However, not all the features are helpful for classification-type problems. The features that are reductant and Irrelevant may reduce the CA. Selecting all the features increases the time complexity of the problem. Also, choosing all the features will increase the dimensionality of the problem. The selection of features plays a critical role in the classification type problems. So, the primary objective of FS problems is to decrease the NF and increase the problem's accuracy. It broadly has two categories: wrapper and filter. The wrapper method uses a classification algorithm on the features, while the filter is independent of any classification algorithm. Wrapper approaches have a better classification performance when compared with filter methods. However, the wrapper methods are computationally expensive compared to filter methods, as they are cheap. So, both methods have their advantages and disadvantages. Hence, if we combine both methods, we might get a better result. Much work has been done in this direction. Few works[52] in this direction have been done by combining both filter and wrapper approaches, but their computational cost is higher. There are many applications of FS problems, such as Image processing and computer vision, Text mining, Industrial applications, Bioinformatics, etc. Many metaheuristic methods have been applied to various FS problems[53]. Many Evolutionary computational techniques like GP[54], GA[55], and PSO[56] are used on the FS problems because of their global search ability and effectiveness. Many Hybrid meta-heuristic methods like[57,58] are developed and applied on different engineering problems or benchmark optimization problems.

The rest of the chapter is structured as follows. Section 4.2, explains some basic concepts required for a proper understanding of this chapter. Section 4.3, explains about the problem we have tackled in this chapter. In Section 4.4, the proposed HPSO approach has been explained. The convergence work and proof of stability analysis of the proposed HPSO algorithm have been explained in Section 4.5. In Section 4.6, the proposed HPSO algorithm is compared with other four metaheuristic algorithms using seven datasets. Then their statistical significance is checked. Also, the time completion of all the methods is computed in this section. Finally, in Section 4.7, the conclusion of the present work has been given.

4.2 Background

To fully grasp the content in this chapter, it is imperative to be familiar with fundamental concepts like PSO and GA, as explained in Section 1.3. Additionally, a basic understanding of FS problems, explained in Section 1.5, is essential.

4.3 Problem Formulation

In this section, the problem of this chapter is explained. For our proper understanding, some important terms and concepts are defined. They are as follows-

Filter Evaluations (FE)

The Filter approach technique is used to speed the fitness evaluation process. Hence computationally cheap measure, mutual information, is employed here to form the FE[59]. The FE is used to maximize and minimize the relevance and redundancy of the selected features respectively. Eq. (4.1) is the FE, where D and R are the relevancy and redundancy of the selected features respectively.

$$F_1(X) = D - R \tag{4.1}$$

Where, $D = \sum_{x \epsilon X} I(x; c)$ and $R = \sum_{x \epsilon X, y \epsilon Y} I(x; y)$

Wrapper Evaluations (WE)

The WE is to maximise the CA of the selected feature subset, which is calculated by Eq. (4.2). For WE we need a classification method and there are many classification methods. Here, we have used KNN classification for classification method, with *K*=1 as it is widely used and simple. The number of correctly classified instances divided by the total number of instances gives the accuracy.

$$F_2(X) = accuracy \tag{4.2}$$

Pseudo-code for finding the accuracy

```
Model= fitness (x, y, 'NumNeighbors', k);
prediction = predict (Model, xvalid);
total NF = length(y_valid);
correct   = 0;
for i = 1:total NF
  if isequal(yvalid(i), prediction(i))
    correct = correct + 1;
  end
end
Accuracy   = correct / total NF;
```

NF

In all FS problems, many datasets are involved. Considering all the features and solving the problems is impossible, as it increases computational time complexity. So most FS problems prefer to take less NF.

$$F_3(X) = Number\ of\ Features \tag{4.3}$$

Pseudo-code for finding the Number of features

```
For Position=1:dimension
S1= Position((particle>parameter)==1)
Selected_feature=feat(:,S1)
Number of Features =length(Selected_feature)
end
```

The objective is to increase the CA and to decrease the NF. For the FS problem, a combination of filter and wrapper method is taken and then the proposed HPSO algorithm is applied. Theoretical explanation of stability and convergence of the proposed HPSO algorithm has been explained with proof. Then the proposed algorithm is compared with other metaheuristic algorithms. Then Friedman's test and Mann Whitney U test (MWUT)[60] are applied to check the statistical significance. In this chapter, every row of the position matrix is the position of the particle. Here 0 means that the feature has not been selected and 1 means that the feature has been selected. The quality of the representation plays a significant role for the effectiveness of the proposed method.

4.4 Proposed HPSO Algorithm

1 First, all the parameters of PSO are initialized. The particle swarm (N) is given before the start of the experiment as it depends on the experiment

2 The position for each particle is initialized. So random matrices have been taken, where each row represents the position of the particle. For the effectiveness of the method, proper one-to-one correspondence between the possible solution and position matrix is very critical.

3 The velocity and position of the particles for the first iteration are taken as random matrices.

4 *For t= 1: Max_iteration*, Here the main iteration loop starts.

5 Compute the filter evaluation of each particle which will be treated as fitness value for filter evaluation (F_1).

6 Compute the wrapper evaluation of each particle which will be treated as fitness value for wrapper evaluation (F_2).

7 So, here the objective function is $\big((F_1 + F_2) - F_3\big)$.

8 Initialize the personal best and compute the filter evaluation and wrapper evaluation for that position.

9 Compute the filter evaluation. Then if we get a desirable result, then we will compute the wrapper evaluation.

10 Update the personal best of each particle. Then we will update the filter and wrapper evaluations.

11 Update the global best.

12 Update the velocity and position of each particle using the Eqns. (1.16) and (1.17).

13 Apply the cross-over operator and mutation operator. Then one of them will be treated as a global best solution. If termination criteria are not achieved, then the process keeps repeating. So, steps 5 to 12 will be repeated. After achieving termination criteria, select the features using global best, and its CA is reported.

4.5 Mathematical Analysis of Proposed Algorithm

The stability analysis of the proposed HPSO algorithm has been discussed in Sections 4.5.1.

4.5.1 Stability Analysis

The stability of the proposed algorithm is explained using the von Neumann stability criterion for finite difference scheme (FDS) and the Fourier series concept.

Theorem

The Proposed HPSO algorithm with the following two conditions

a) Uses a random three-point crossover concept from GA. Here, the two parent solutions are the global best solution (z^g) and a random personal best solution (z^p)

b) Uses mutation concept on global best solution (z^g).

is stable iff h, h_1 and h_2 satisfies $0 \leq (h_1 + h_2) \leq 2(1 + h)$

Proof: Considering Eqns. (1.16) and (1.17) of the PSO method, we get

$$z(i,t+1) = z(i,t) + h \times s(i,t) + (h_1 \times c_6) \times \left(z^p(i,t) - z(i,t)\right) + (h_2 \times c_7) \times \left(z^g(i,t) - z(i,t)\right) \quad (4.4)$$

Where $z(i,t)$ is the position of particle i at iteration t. Now conditions (a) and (b) are applied to the global best solution (z^g). If the value of (z^g) is better than (z). The value of (z^g) will be assigned to (z), otherwise not. For better understanding, we simply use system by taking z^p and z^g as constants. Let $b_1 = h_1 \times c_6$, $b_2 = h_2 \times c_7$, $z^p(i,t) = z'^p$ and $z_g(i,t) = z'^g$.

Now,

$$z(i,t+1) = z(i,t) + h \times s(i,t) + (b_1) \times \left(z'^{p} - z(i,t)\right) + (b_2) \times \left(z'^{g} - z(i,t)\right)$$

$$z(i,t+1) = z(i,t) \times (1 - b_1 - b_2) + (b_1 \times z'^{p}) + (b_2 \times z'^{g}) + \left(h \times s(i,t)\right)$$

$$z(i,t+1) = z(i,t) \times (1 - b_1 - b_2) + (b_1 \times z'^{p}) + (b_2 \times z'^{p}) + \left(h \times \left(z(i,t) - z(i,t-1)\right)\right)$$

$$z(i,t+1) = \left(z(i,t) \times (1 + h - b_1 - b_2)\right) - \left(h \times z(i,t-1)\right) + (b_1 \times z'^{p}) + (b_2 \times z'^{g})$$

$$z(i,t+1) - \left(z(i,t) \times (1 + h - b_1 - b_2)\right) + \left(h \times z(i,t-1)\right) = (b_1 \times z'^{p}) + (b_2 \times z'^{g})$$

or

$$z_{i,t+1} - \left(z_{i,t} \times (1 + h - b_1 - b_2)\right) + \left(h \times z_{i,t-1}\right) = (b_1 \times z'^{p}) + (b_2 \times z'^{g}) \quad (4.5)$$

Let $t = t + 1$, for simplicity purpose

$$z_{i,t+2} - \left(z_{i,t+1} \times (1 + h - b_1 - b_2)\right) + \left(h \times z_{i,t}\right) = (b_1 \times z'^{p}) + (b_2 \times z'^{g}) \quad (4.6)$$

Generalized FDS is given by –

$$\sum_{q=-z`_l}^{z`_s} Z'_q \, X'_{m,j+q} = \sum_{q=-c_l}^{c_s} c_q \, X'_{m+1,j+q} + D \quad (4.7)$$

Here z'_l, $z`_s$, $c_l$and c_s are non-negative integers. On Comparing Eqns. (4.6) and (4.7)

$$Z'_{-1} = q, Z'_0 = -(1 + h - b_1 - b_2), Z_1 = 1, X'_{-1} = X'_0 = X'_1 = 0$$

$$\text{and } D = (b_1 \times z'^{p}) + (b_2 \times z'^{g})$$

Since Eq. (4.6) is a non-homogeneous. We will consider an associated homogeneous differential scheme of Eq. (4.6) to analyze the stability condition. Now,

$$z_{i,t+2} - \left(z_{i,t+1} \times (1 + h - b_1 - b_2)\right) + \left(h \times z_{i,t}\right) = 0$$

$$z_{i,t+2} - \left(z_{i,t+1} \times \Lambda\right) + \left(h \times z_{i,t}\right) = 0 \quad (4.8)$$

where $\Lambda = 1 + h - b_1 - b_2$

Let $z = x(i,t)$ be the exact solution of the $i - t$ computational domain. Then $z\left(i_h, t_j\right)$ is the approximate solution at the nodes of the grids. For stability analysis of HPSO, the FDS Eq. (4.8) is considered instead of Eq. (4.5). The von Neumann stability criteria for FDS are used

to analyze the stability of the HPSO method. Let the *m*th component of the Fourier series solution of the above equation be given by:

$$z(i_h, t_j) = C_m e^{l(\sigma_m i_h - \beta_m t_j)}$$

$$z_{i_h,t_j} = C_m e^{l(\sigma_m h\Delta i - \beta_m j\Delta t)} \tag{4.9}$$

where $l = \sqrt{-1}$, $i_h = h\Delta i$ $t_j = j\Delta d$, C_m represents the amplitude of the m^{th} component, β_m is the angular frequency, σ_m is the wave number of m^{th} component. Now consider Eq. (4.8) in terms of the grid point (i_h, t_j)

$$z_{i_h,t_j+2} - z_{i_h,t_j+1} \times (\Lambda) + h \times z_{i_h,t_j} = 0 \tag{4.10}$$

Now, putting the value of Eq. (4.9) to Eq. (4.10)

$$C_m e^{l(\sigma_m h\Delta i - \beta_m j\Delta t)}(e^{-l\beta_m 2\Delta t} - \Lambda e^{-l\beta_m \Delta t} + h) = 0 \tag{4.11}$$

Since $C_m \neq 0$, $(e^{-l\beta_m 2\Delta t} - \Lambda e^{-l\beta_m \Delta t} + h) = 0$

$$Z'^2 - \Lambda Z' + h = 0 \tag{4.12}$$

where $Z' = e^{-l\beta_m \Delta t}$, Z' is the amplification factor. Here, a deterministic approach is used to solve the quadratic Eq. (4.12). Now,

$$Z' = \frac{\Lambda \pm \sqrt{(\Lambda^2 - 4q)}}{2}$$

According to von Neumann's stability criteria, the FDS is stable iff |(amplification factor)| ≤ 1. FDS is given by Eq. (4.8) is considered, so the FDS given by (4.4) is considered. Hence HPSO is stable iff $|Z`| \leq 1$

$$\left|\frac{\Lambda \pm \sqrt{\Lambda^2 - 4h}}{2}\right| \leq 1$$

Hence, $-1 \leq \frac{\Lambda \pm \sqrt{\Lambda^2 - 4h}}{2} \leq 1$. Now we have two cases.

Case A: $\frac{\Lambda \pm \sqrt{\Lambda^2 - 4h}}{2} \leq 1$.

$$\pm\sqrt{\Lambda^2 - 4h} \leq 2 - \Lambda \rightarrow \left|\sqrt{\Lambda^2 - 4h}\right| \leq 2 - \Lambda$$

Now, Squaring both sides of the above equation, we get

$$(\Lambda^2 - 4h) \leq (2 - \Lambda)^2$$

Now replace the value of Λ by $1 + h - b_1 - b_2$ in the above inequality. Then, solving the inequality, we get

$$0 \leq h_1 + h_2 \tag{4.13}$$

Case B: $-1 \leq \frac{\Lambda \pm \sqrt{\Lambda^2 - 4h}}{2}$

$$-(2 + \Lambda) \leq \pm\sqrt{\Lambda^2 - 4h}$$

$$\rightarrow (2 + \Lambda) \geq \mp\sqrt{\Lambda^2 - 4h}$$

$$\rightarrow (2 + \Lambda) \geq \left|\sqrt{\Lambda^2 - 4h}\right|$$

On squaring both sides of the above equation, we get $(2 + \Lambda)^2 \geq (\Lambda^2 - 4h)$

Now replacing the value of Λ by $1 + h - b_1 - b_2$ in the above inequality. Then solving the inequality, we get

$$h_1 + h_2 \leq 2(1 + h) \tag{4.14}$$

From Eqns. (4.14) and (4.13). We get the stability condition-

$$0 \leq h_1 + h_2 \leq 2(1 + h) \tag{4.15}$$

Hence the proposed HPSO is stable iff $0 \leq h_1 + h_2 \leq 2(1 + h)$

4.6 Results and Discussions

Then, the proposed HPSO method is compared to four different methods (Binary Harris Hawks Optimization (BHHO)[61], GA, Salp Swarm Algorithm (SSA)[15]) on seven datasets[39] (Ionosphere, Wine, Breast Cancer Wisconsin, Sonar, Libras Movement, Hill Valley, Musk 1) using UCI Machine Learning respiratory as given in Table 1.3. On each dataset, we have conducted five experiments. In each experiment, we have run the Matlab code six times. Then, we recorded their CA, feature, and computational time for each experiment and each dataset. We have also checked whether the difference is statistically significant or not by applying Friedman's test and the MWUT. For more detailed information on the work, the result of the Musk 1 dataset has been explained in Section 4.6.1**.** In Section 4.6.2, we have discussed the effect of all the methods on all seven datasets.

4.6.1 Musk 1[39]

The graphs of experiments 1, 2, 3, 4, and 5 of all the methods are obtained. In Figs 4.1 and 4.2, the first row gives the CA, where PSO, HPSO, GA, BHHO, and SSA are represented by green, red, yellow, blue, and black colours, respectively. In the second and third rows of the Figs 4.1 and 4.2, we are getting the NF and the time complexity of all the methods. Here, PSO, HPSO, GA, BHHO, and SSA are represented by bar 1, bar 2, bar 3, bar 4, and bar 5, respectively. The graphs of experiments 1 and 2 are in Figs 4.1 and 4.2.

In Table 4.1, the CAs of the experiments 1, 2, 3, 4, and 5 are given in a detailed manner. Since we have taken six runs in each experiment, every run is represented by a column in Table 4.1. Hence, each column of Table 4.1 represents a run.

Table 4.1. Results of PSO, HPSO, GA, BHHO, and SSA obtained in five Experiments concerning CA

Experiment 1	**RUN 1**	**RUN 2**	**RUN 3**	**RUN 4**	**RUN 5**	**RUN 6**
PSO	0.936842	0.957895	0.936842	0.947368	0.957895	0.968421
HPSO	**0.968421**	**0.957895**	**0.957895**	**0.978947**	**0.978947**	**0.978947**
GA	0.947368	0.936842	0.957895	0.926316	0.936842	0.957895
BHHO	0.947368	0.936842	0.936842	0.947368	0.957895	0.926316
SSA	0.947368	0.926316	0.957895	0.947368	0.926316	0.936842
Experiment 2						
PSO	0.947368	0.926316	0.957895	0.926316	0.915789	0.978947
HPSO	**0.978947**	**0.968421**	**0.978947**	**0.947368**	**0.957895**	**0.978947**
GA	0.968421	0.915789	0.957895	0.968421	0.926316	0.957895
BHHO	0.926316	0.915789	0.915789	0.936842	0.915789	0.947368
SSA	0.915789	0.936842	0.926316	0.936842	0.936842	0.947368
Experiment 3						
PSO	0.947368	0.957895	0.978947	0.957895	0.978947	0.947368
HPSO	**0.978947**	**0.978947**	**0.968421**	**0.968421**	**0.978947**	**0.968421**
GA	0.968421	0.947368	0.957895	0.915789	0.957895	0.926316
BHHO	0.936842	0.947368	0.947368	0.947368	0.936842	0.905263
SSA	0.947368	0.936842	0.936842	0.936842	0.947368	0.894737
Experiment 4						
PSO	0.915789	0.926316	0.947368	0.915789	0.915789	0.957895
HPSO	**0.926316**	**0.947368**	**0.968421**	**0.978947**	**0.936842**	**0.968421**
GA	0.915789	0.905263	0.915789	0.936842	0.894737	0.978947

BHHO	0.905263	0.905263	0.894737	0.905263	0.894737	0.936842
SSA	0.915789	0.894737	0.894737	0.947368	0.894737	0.947368
Experiment 5						
PSO	0.957895	0.936842	0.947368	0.968421	0.968421	0.915789
HPSO	**0.968421**	**0.968421**	**0.989474**	**0.989474**	**0.989474**	**0.989474**
GA	0.978947	0.978947	0.947368	0.957895	0.905263	0.905263
BHHO	0.936842	0.936842	0.936842	0.947368	0.936842	0.947368
SSA	0.947368	0.926316	0.947368	0.936842	0.936842	0.957895

Note: Bold values indicate the best value

Experiment 1

Here, we observe from Fig 4.1 that 4 times, HPSO is giving optimal results, 2 times PSO is giving optimal results and 1 time GA and SSA are showing optimal results concerning CA. Again, 3 times, GA takes the least NF, 2 times, HPSO takes the least NF, and 1 time, SSA takes the least NF. The detailed results in Fig 4.1 are numerically written in Table 4.1.

Experiment 2

Here, we observe from Fig 4.2 that 5 times, HPSO is giving optimal results, and 1 time, GA is giving optimal results concerning CA. Again, 2 times BHHO is taking the least NF, and the rest are taking the least NF 1 time. The detailed results in Fig 4.2 are numerically written in Table 4.1.

Experiments 3, 4, 5

From Experiment 3, 5 times HPSO gives optimal results and 2 times PSO gives optimal results concerning CA. Again, 3 times, HPSO takes the least NF, 2 times, SSA takes the least NF, and GA takes the least NF 1 time. From Experiment 4, 6 times, HPSO gives optimal CA results. Again, 6 times, BHHO is taking the least NF. From Experiment 5, 4 times HPSO gives optimal results, and 2 times GA gives optimal results concerning CA. Again, 2 times HPSO is taking the least NF, and the rest are taking the least NF 1 time.

4.5.2 Discussions

The experiment was conducted five times for each dataset. We have recorded their accuracy, NF, and time complexity. Figs 4.3 to 4.9 are the convergence curve, which compares all five methods on the seven datasets of a particular run of experiment 1. In Tables 4.2 and 4.3, the observations of the experiment concerning CA and NF are given, respectively. Table 4.2 gives the mean CA by all five methods on all seven datasets and the Friedman average rank (FAR). Table 4.3 gives the mean number of features selected by all five methods on all seven datasets, along with the average Friedman ranking. Here, the values in bold are best, and those in *italics* are second best.

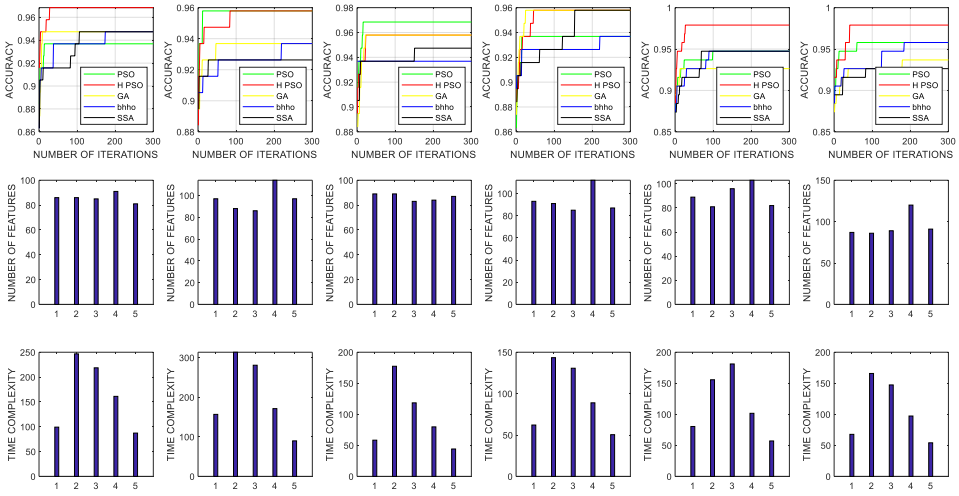

Fig. 4.1. Graph of Experiment 1

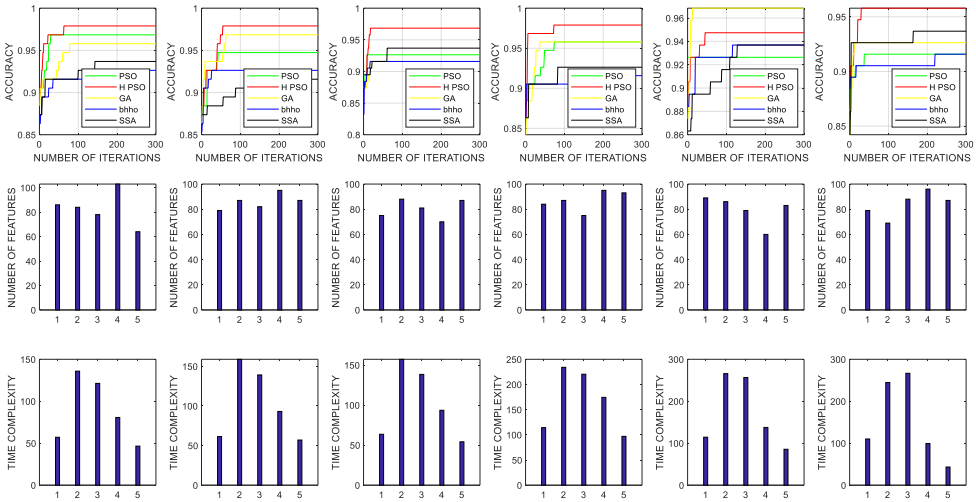

Fig. 4.2. Graph of Experiment 2

Table 4.2. Experimental results of Mean CA using PSO, HPSO, GA, BHHO, and SSA methods on the seven datasets along with Friedman ranking

	Musk 1	Hill Valley	Libras Moment	Sonar	Ionosphere	WDBC	Wine	FAR
PSO	*0.94702*	0.610193	*0.82361*	0.92264	0.930476	0.943953	0.968571	3.43
HPSO	**0.969474**	**0.639669**	**0.843056**	*0.939453*	*0.93857*	0.943363	0.977143	2.14
GA	*0.94105*	*0.62204*	0.821759	0.922403	0.93619	0.943953	*0.98*	2.86
BHHO	0.930526	0.590634	0.813426	*0.92392*	**0.945912**	**0.957817**	**0.995238**	2.57
SSA	0.932982	0.590083	0.812963	0.916486	0.930952	*0.94867*	0.978095	4

Note: Bold values indicate the best value, and italic values indicate the second best value

Table 4.3. Experimental results of Mean of NF selected using PSO, HPSO, GA, BHHO, and SSA methods on the seven datasets along with Friedman ranking

	Musk 1	Hill Valley	Libras Moment	Sonar	Ionosphere	WDBC	Wine	FAR
PSO	**75**	47.2	44.23333	33.2667	14.8	16.5	5.43333	3
HPSO	77.133	**45.9667**	**42.03333**	**31.8333**	*14.5667*	16.0333	6.06667	2.29
GA	*76.067*	47.3666	43.83333	32.5333	15.4	15.3	*6.36667*	3.43
BHHO	91.8	*46.1*	44.76667	*32.6667*	**22.56667**	**13.033**	**2.36667**	3.43
SSA	83.4	47.8666	*42.9333*	31.1333	13.86667	*14.866*	6.83333	2.86

Note: Bold values indicate the best value, and italic values indicate the second best value

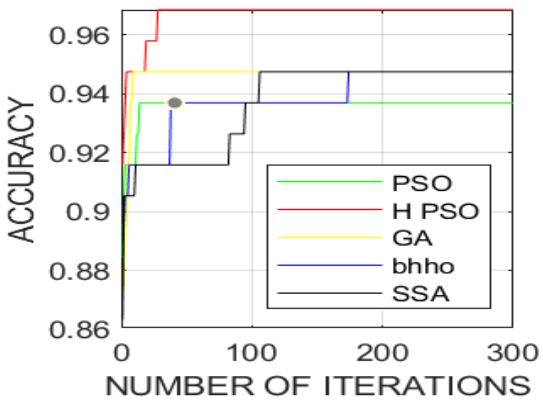

Fig. 4.3. CA of all the methods on the Musk 1 dataset.

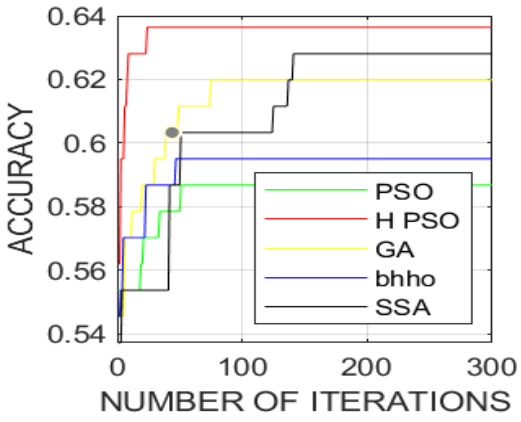

Fig. 4.4. CA of all the methods on the Hill Valley dataset.

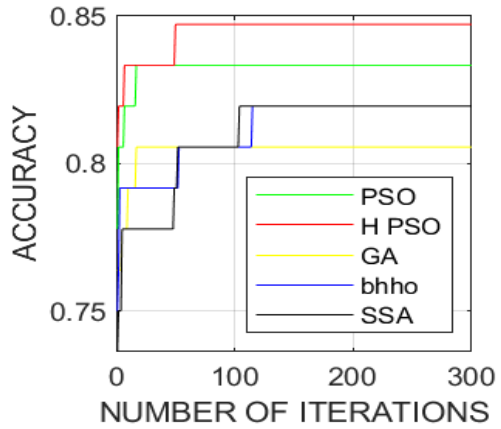

Fig. 4.5. CA of all the methods on the Libras Moment dataset.

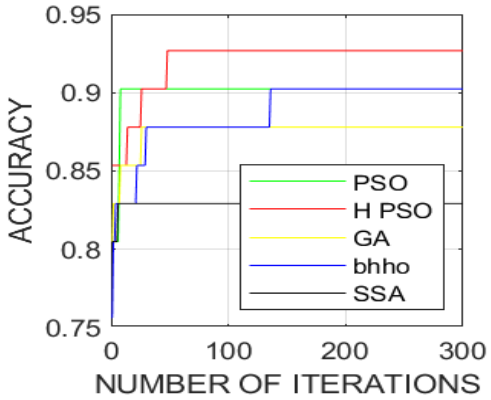

Fig. 4.6. CA of all the methods on the Sonar dataset.

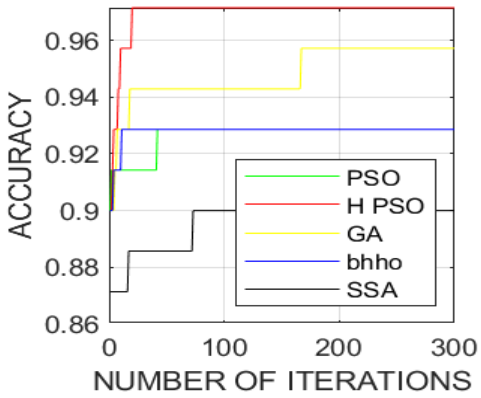

Fig. 4.7. CA of all the methods on the Ionosphere dataset.

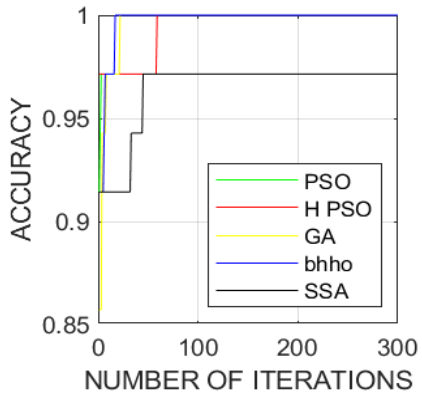

Fig. 4.8. CA of all the methods on the WDBC dataset.

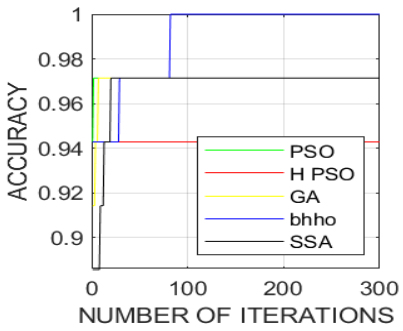

Fig.4.9. CA of all the methods on the Wine dataset.

4.7 Conclusion

From Table 4.2 and the U values of HPSO against other methods, we observe that HPSO has the best solution, followed by BHHO, then again by GA, then by PSO, and then by SSA. Hence, we can conclude that HPSO has the best CA with a statistically significant difference. From Table 4.3 and the U values of HPSO against other methods, we observe that HPSO has the best solution, followed by SSA, then again followed by PSO, then by GA and BHHO. Hence, we can conclude that HPSO takes the least NF with a statistically significant difference. Thus, we get a statistically significant difference after applying the Friedman's test and MWUT.

Hence, based on statistical measures and convergence rates, the proposed HPSO method gives better results on the high-dimensional and medium-dimensional datasets than the other meta-heuristic methods considered in this paper. However, on low-dimensional datasets, the results are satisfactory. Since the practical applications of FS problems involve large datasets, the proposed HPSO is more application-oriented and useful. This satisfies our objective to increase CA and decrease the NF.

Chapter 5

Comparative study between Hybrid Particle Swarm Optimization and Particle Swarm Optimization on a Multi-Objective Feature Selection Problem

5.1 Introduction

Classification is a common application of machine learning, which finds the class for a given instance based on a set of similarities[62]. However, due to the abundance of noisy, pointless, and duplicated features on some datasets, CA declines significantly as the number of features rises and training time accelerates[63]. Three crucial dimensionality reduction technologies—feature extraction, feature building, and feature selection—are suggested to eliminate these characteristics[64]. The fundamental difficulty feature selection algorithms face as the number of features rises a broad search space[65]. Consequently, the search method is a crucial component of feature selection algorithms. Sequential forward selection (SFS) and sequential background selection (SBS) are examples of traditional search algorithms that can identify the best answers. Still, they need much computing time and are prone to local optimization, especially on high-dimensional datasets[66]. PSO, Grey Wolf Optimization (GWO), teaching-learning-based optimization (TLBO), bacterial colony optimization (BCO), GA, ABC, and differential evolution (DE) are examples of metaheuristic algorithms that have a better global search capability than conventional search techniques. Hence, these algorithms are frequently used in feature selection algorithms. As a result, much research focuses on feature selection algorithms built using metaheuristic algorithms.

It is challenging for a single swarm on high-dimensional datasets to tackle the feature selection challenges[67]. As a result, variable-length PSO representation was proposed by decomposing the entire collection of features into feature subsets with various characteristics[68]. Combining classification performance and the number of features into a single fitness function using a single assessment criterion is technically challenging, even though many researchers delete some features using information theory or select feature subsets with good classification accuracy. As a result, feature selection methods based on metaheuristic algorithms have been given the MOO treatment[69]. Since maximizing classification performance and reducing the number of features are two competing goals, feature selection is a multi-objective issue[56]. Generally speaking, the Pareto dominance mechanism on the archive and diversity enhancement strategies is primarily used in feature selection algorithms based on multi-objective metaheuristic algorithms. The TMABC-FS[70] algorithm incorporates a multi-objective ABC algorithm with a two-archive mechanism based on the Pareto mechanism to increase diversity. The two search algorithms must maintain a fair balance of convergence and diversity, and the upkeep of the two archives requires a significant amount of computation time[71]. To preserve population diversity, a multi-objective immune method using an elite selection strategy based on reference vectors was proposed[72]. Recently, MOFS-BDE[73] proposed a one-bit purifying search operator for the self-learning capability, non-dominated sorting with crowding distance on the archive, and a binary mutation operator based on possibility differences to identify viable solutions.

This is how the rest of the chapter is organised. Certain fundamental ideas necessary for comprehending this chapter properly are explained in Section 5.2. Section 5.3 describes the issue that we have addressed in this chapter. The suggested HPSO technique for the multi-objective FS problem has been covered in Section 5.4 of the chapter. Section 5.5 explains the theoretical analysis of the suggested HPSO algorithm. Using seven datasets, the proposed HPSO algorithm is contrasted with four different metaheuristic algorithms in Section 5.6. They are next examined for statistical significance. This section also computes the time completion of all the techniques. Lastly, the conclusion of the present work is provided in Section 5.7.

5.2 Background

A comprehensive understanding of this chapter requires familiarity with core concepts like PSO and GA, detailed in Section 1.3. It is equally critical to have a basic understanding of MOO for proper comprehension, as explained in Chapter 3, Section 3.2. Additionally, an awareness of FS problems, discussed in Section 1.5, is vital.

5.3 Problem Formulation

For our proper understanding, some important terms and concepts are defined. Most literature combines the two objectives, forming a single objective optimization problem. But in this chapter, we are dealing with them separately, forming a MOO problem. This gives rise to the set of Pareto-optimal solutions. Hence, a hybrid metaheuristic algorithm for the multi-objective FS problem is developed and explained in this chapter.

The objective functions of the MOO problem are explained here. We have two objective CA and the NF. The article aims to maximize the selected features CA and minimize the NF. If we increase the CA, the NF will decrease and vice versa. Since both objective functions are contradictory, we will get a set of Pareto-optimal solutions.

Here, we have used the concept of wrapper evaluation for computing CA, which is the first objective function $F_1(X)$ of the FS problem. The wrapper fitness function maximizes the CA of the selected feature subset and is calculated using Eq. (4.2). Here, the number of correctly classified instances is divided by the total number of cases computed. Here, we have used the KNN classification, with K=1 for its simplicity. The pseudo-code and formula for finding the CA are given in the previous chapter in Section 4.3. The NF is calculated as given in Eq. (4.3) and its pseudo-code is given in the previous chapter in Section 4.3. Here , the first objective function is $F_1(X) + F_2(X)$ and the second objective function is $F_3(X)$.

5.4 Proposed HPSO Algorithm

The proposed algorithm uses the concept of a cross-over operator to develop a hybrid PSO. The pseudo-code of the proposed algorithm is given below:

***INPUT*:** Initialization of the position of each particle and all the parameters are done.

***OUTPUT*:** Pareto-optimal solutions, with each solution giving accuracy and NF.

(1) Each particle is initialized with the help of random matrices. Each row of the matrix represents the particle's position. Therefore, proper one-to-one correspondence between the possible solution and position matrix is critical for the algorithm's effectiveness.

(2) The velocity and position of the particles for the first iteration are taken as random matrices.

(3) Compute the wrapper evaluation of each particle, which will be treated as accuracy for Wrapper evaluation $F_1(X) + F_2(X)$ from Eqns. (4.1) and (4.2).

(4) Compute the NF for each particle $F_3(X)$ from Eq. (4.3).

(5) $t = t + 1$; from here, the iteration process starts till maximum iteration is reached. (We have started from $t=1$).

   a. From the population, a leader set is selected.
   b. The position and velocity are updated using Eq. (1.16) and Eq. (1.17).
   c. Then the boundary conditions are applied.
   d. The accuracy of each particle and the NF taken by each particle are computed.
   e. The personal best for each particle is updated.
   f. Apply the crossover operator between the personal best solution and the global best solution.
   g. Chang the Non-Dominating Front by adding Non-Dominated Particles.
   h. The domination of all the new Non-Dominating Front Members is determined.

(6) The iteration process continues until the Maximum iteration is achieved.

(7) The algorithm will give Pareto-optimal solutions after achieving the termination criteria. Then, each solution will give us accuracy and the NF it takes.

5.5 Mathematical Analysis of Proposed Algorithm

The stability analysis of the proposed HPSO algorithm can be discussed with the help of the groundwork laid out in the previous chapter.

5.6 Results and Discussions

Here, the results obtained using the Proposed HPSO algorithm are compared with the PSO algorithm on seven datasets[39] to know which produces better results. The detailed information on all the datasets is explained in Table 1.3. Since we are dealing with the MOO problem, we will get a set of Pareto optimal solutions. Each Pareto optimal solution gives accuracy and NF. Five experiments on all seven datasets have been conducted. The results of three experiments obtained by using the Ionosphere dataset, the Musk 1 dataset, and two experiments on Hill Valley Datasets have been explained in detail in this section. Both HPSO and PSO algorithms are compared using Figs 5.1- 5.7 and Tables 5.1- 5.16, where DMs, accuracy, and NF are given. In Figs 5.1- 5.7, the PSO algorithm is indicated on the Left side, the HPSO algorithm is marked on the Right side, and the Red colour denotes the DMs.

5.6.1 Ionosphere Dataset

The experimental results on the Ionosphere Dataset have been explained in this section. Here, only three out of the five experiments are explained in detail. The DMs of Experiments 1, 2, and 3 using the Ionosphere dataset are given in Figs 5.1, 5.2, and 5.3.

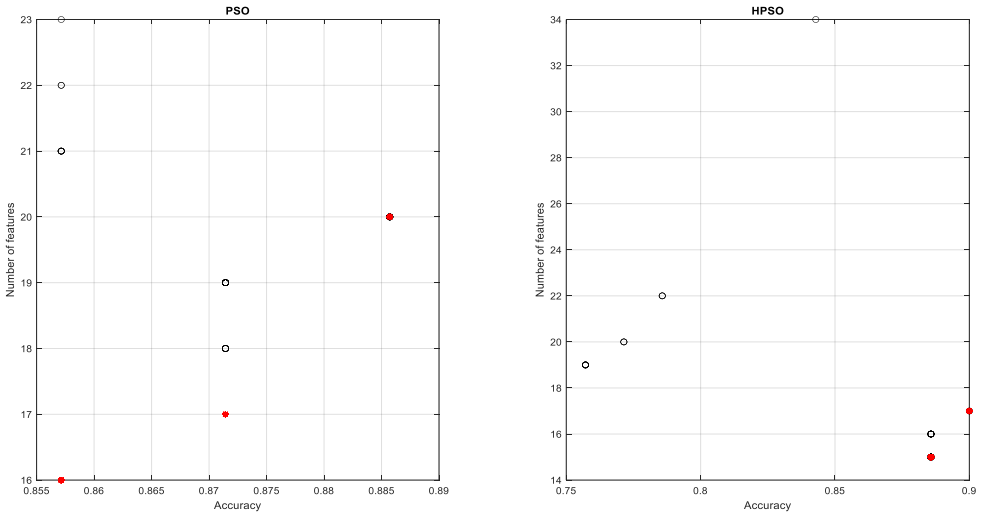

Fig. 5.1. Graph of Experiment 1

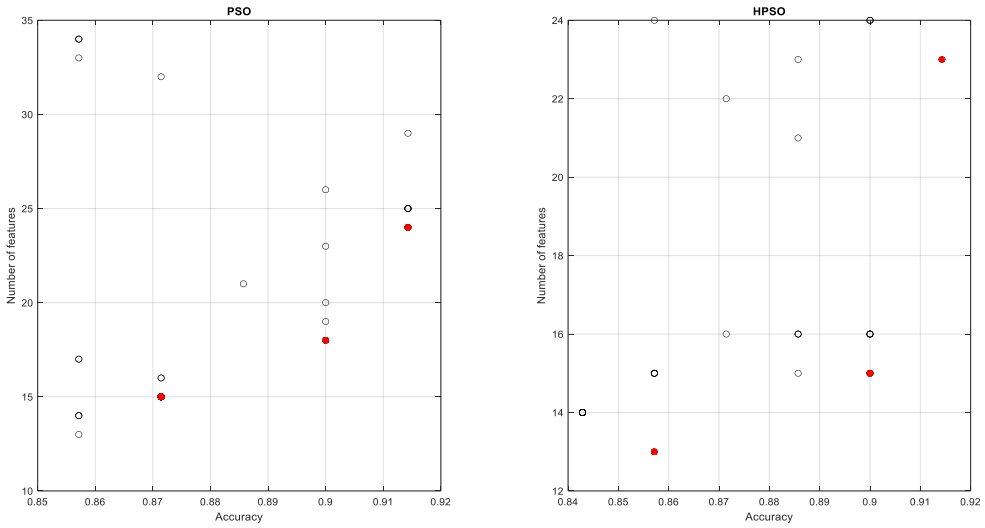

Fig. 5.2. Graph of Experiment 2

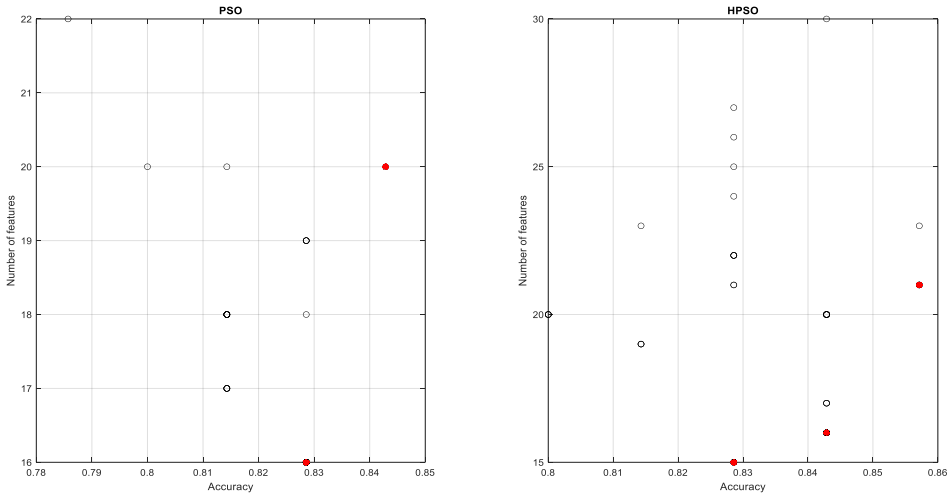

Fig. 5.3. Graph of Experiment 3

Result of Experiment 1

In experiment 1, the DMs in PSO and HPSO algorithms are given in Tables 5.1 and 5.2, respectively, with their accuracy and NF. In this Experiment, the average accuracy of PSO and HPSO algorithms is 0.8714 and 0.8929, respectively. On the other hand, the NF taken by PSO and HPSO algorithms are 18 and 16, respectively. Here, we observe that the HPSO algorithm has higher accuracy and less average NF than the PSO algorithm. So, we are getting a better result when we apply the HPSO algorithm.

Table 5.1. Results for PSO of Experiment 1

DMs	ACCURACY	NF
1	0.871428	17
2	0.857142	16
3	0.885714	20

Table 5.2. Results for HPSO of Experiment 1

DMs	ACCURACY	NF
1	0.900000	17
2	0.885714	15

Result of Experiment 2

In experiment 2, the DMs in PSO and HPSO algorithms are given in Tables 5.3 and 5.4, respectively, with their accuracy and NF. In this Experiment, the average accuracy of PSO and HPSO algorithms is 0.8952 and 0.8905, respectively. On the other hand, the NF taken by PSO and HPSO algorithms are 19 and 17, respectively. Hence, we observe that the average accuracy is approximately the same in both PSO and HPSO algorithms. However, the proposed HPSO algorithm takes less average NF when compared with the PSO algorithm. So, we are getting a better result when we apply the HPSO algorithm.

Table 5.3. Results for PSO of Experiment 2

DMs	**Accuracy**	**NF**
1	0.8714285	15
2	0.9142857	24
3	0.9000000	18

Table 5.4. Results for HPSO of Experiment 2

DMs	**Accuracy**	**NF**
1	0.9000000	15
2	0.8571428	13
3	0.9142857	23

Result of Experiment 3

In experiment 3, the DMs in PSO and HPSO algorithms are given in Tables 5.5 and 5.6, respectively, with their accuracy and NF. In this Experiment, the average accuracy of PSO and

HPSO algorithms is 0.8952 and 0.8905, respectively. On the other hand, the NF taken by PSO and HPSO algorithms are 19 and 17, respectively. Hence, we observe that the average accuracy is approximately the same in both PSO and HPSO algorithms. However, the proposed HPSO algorithm takes less average NF when compared with the PSO algorithm. So, we are getting a better result when applying the HPSO algorithm.

Table 5.5. Results for PSO of Experiment 3

DMs	Accuracy	NF
1	0.828571	16
2	0.842857	20

Table 5.6. Results for HPSO of Experiment 3

DMs	Accuracy	NF
1	0.828571	15
2	0.842857	16
3	0.857142	21

5.6.2 Musk 1 Dataset

The experimental results on the Musk 1 Dataset have been explained in this section. Here, only three out of the five experiments are explained in detail. The DMs of Experiments 1, 2, and 3 are given in Figs 5.4, 5.5, and 5.6.

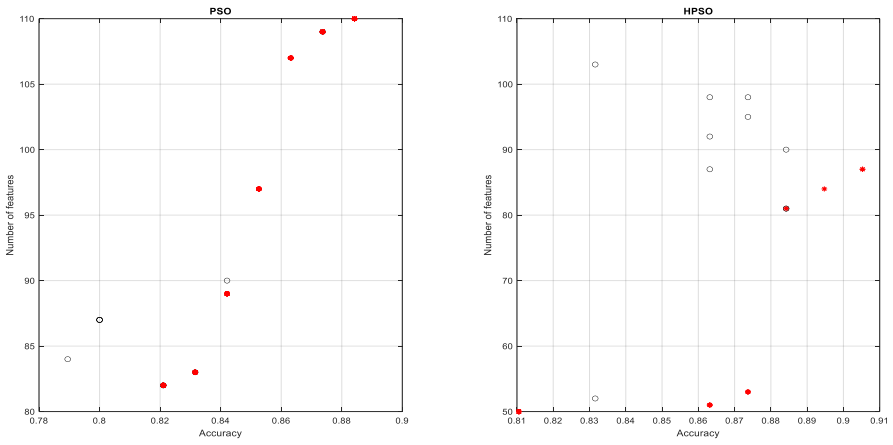

Fig. 5.4. Graph of Experiment 1

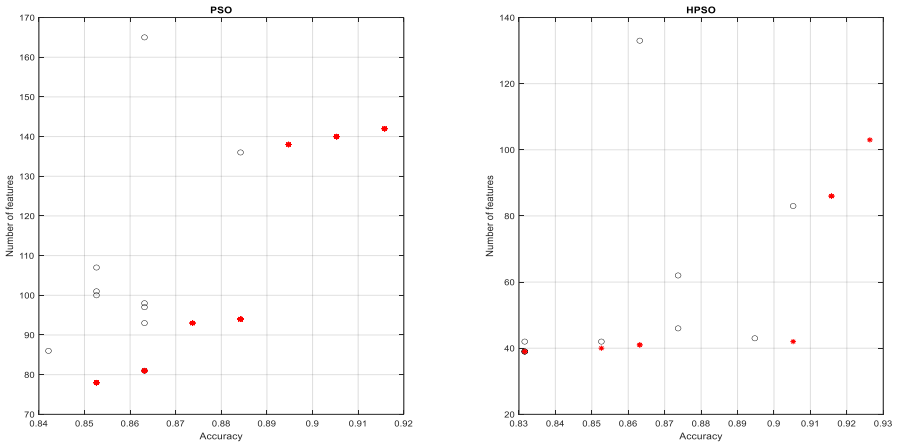

Fig. 5.5. Graph of Experiment 2

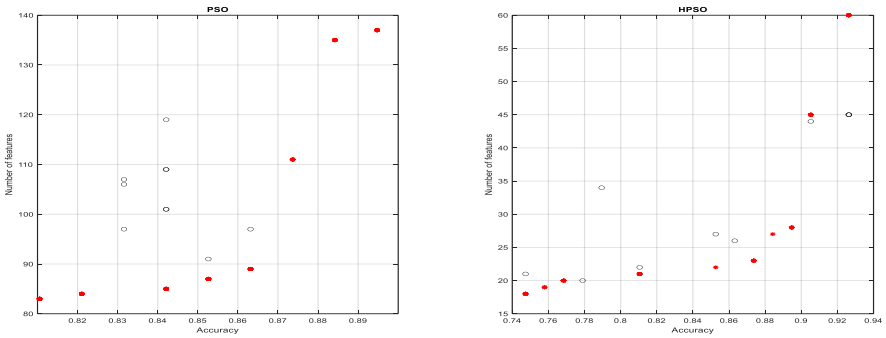

Fig. 5.6. Graph of Experiment 3

Result of Experiment 1

In experiment 1, the DMs in PSO and HPSO algorithms are given in Tables 5.7 and 5.8, respectively, with their accuracy and NF. In this Experiment, the average accuracy of PSO and HPSO algorithms is 0.8526 and 0.8719, respectively. On the other hand, the NF taken by PSO and HPSO algorithms is 96.7143 and 67.6667, respectively. Here, we observe that the HPSO algorithm has higher accuracy and less average NF than the PSO algorithm. So, we are getting a better result when we apply the HPSO algorithm.

Table 5.7. Results for PSO of Experiment 1

DMs	Accuracy	NF
1	0.852631	97
2	0.842105	89
3	0.831578	83
4	0.884210	110
5	0.873684	109
6	0.863157	107
7	0.821052	82

Table 5.8. Results for HPSO of Experiment 1

DMs	Accuracy	NF
1	0.810526	50
2	0.873684	53
3	0.863157	51
4	0.884210	81
5	0.894736	84
6	0.905263	87

Result of Experiment 2

In experiment 2, the DMs in PSO and HPSO algorithms are given in Tables 5.9 and 5.10, respectively, with their accuracy and NF. In this Experiment, the average accuracy of PSO and HPSO algorithms is 0.8842 and 0.8825, respectively. On the other hand, the NF taken by PSO and Hybrid HPSO algorithm is 109.4286 and 58.5, respectively. Here, we observe that the HPSO algorithm has higher accuracy and less average NF than the PSO algorithm. So, we are getting a better result when we apply the HPSO algorithm.

Table 5.9. Results for PSO of Experiment 2

DMs	Accuracy	NF
1	0.852631	78
2	0.863157	81
3	0.873684	93
4	0.884210	94
5	0.894736	138
6	0.905263	140
7	0.915789	142

Table 5.10. Results for HPSO of Experiment 2

DMs	Accuracy	NF
1	0.831578	39
2	0.852631	40
3	0.863157	41
4	0.905263	42
5	0.915789	86
6	0.926315	103

Result of Experiment 3

In experiment 3, the DMs in PSO and HPSO algorithms are given in Tables 5.11 and 5.12, respectively, with their accuracy and NF. In this Experiment, the average accuracy of PSO and HPSO algorithms is 0.8553 and 0.8421, respectively. On the other hand, the NF taken by PSO and HPSO algorithms are 101.3750 and 28.3, respectively. Hence, we observe that the average accuracy of PSO is better than that of the HPSO algorithm. However, the proposed HPSO algorithm takes less average NF when compared with the PSO algorithm. So, we are getting a better result when we apply the HPSO algorithm.

Table 5.11. Results for PSO of Experiment 3

DMs	Accuracy	NF
1	0.821052	84
2	0.842105	85
3	0.852631	87
4	0.863157	89
5	0.873684	111
6	0.884210	135
7	0.894736	137
8	0.810526	83

Table 5.12. Results for HPSO of Experiment 3

DMs	Accuracy	NF
1	0.926315	60
2	0.905263	45
3	0.894736	28
4	0.810526	21
5	0.768421	20

6	0.747368	18
7	0.873684	23
8	0.757894	19
9	0.852631	22
10	0.884210	27

5.6.3 Hill Valley Dataset

The experimental results on the Hill Valley Dataset have been explained in this section. Here, only three out of the five experiments are explained in detail. The DMs of Experiment 1 are given in Fig 5.7.

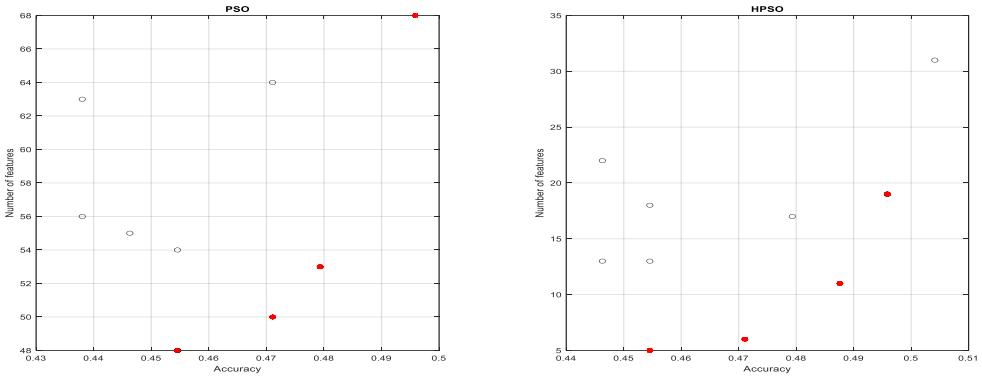

Fig. 5.7. Graph of Experiment 1

Result of Experiment 1

In experiment 1, the DMs in the PSO and HPSO algorithms are given in Tables 5.14 and 5.15, respectively, with their accuracy and NF. In this Experiment, the average accuracy of the PSO and HPSO algorithms attained is 0.4752 and 0.4773, respectively. On the other hand, the NF taken by the PSO and HPSO algorithms is 54.75 and 10.25, respectively. Here, we observe that the HPSO algorithm has higher accuracy and less average NF than the PSO algorithm. So, we are getting a better result when we apply the HPSO algorithm.

Table 5.13. Results for PSO of Experiment 1

DMs	Accuracy	NF
1	0.495867	68
2	0.471074	50
3	0.479339	53
4	0.454545	48

Table 5.14. Results for HPSO of Experiment 1

DMs	Accuracy	NF
1	0.454545	5
2	0.471074	6
3	0.487603	11
4	0.495868	19

Result of Experiment 2

In experiment 2, the DMs in the PSO and HPSO algorithms are given in Tables 5.16 and 5.17, respectively, with their accuracy and NF. In this Experiment, the average accuracy of PSO and HPSO algorithms is 0.5240 and 0.5438, respectively. On the other hand, the NF taken by the PSO and HPSO algorithms is 52.6 and 5.2, respectively. Here, we observe that the HPSO algorithm has higher accuracy and less average NF than the PSO algorithm. So, we are getting a better result when we apply the HPSO algorithm.

Table 5.15. Results for PSO of Experiment 2

DMs	Accuracy	NF
1	0.512396	51
2	0.520661	52
3	0.545454	56
4	0.537190	55
5	0.504132	49

The mean accuracy and NF of both algorithms on all the datasets are given in Tables 5.17 and 5.18, respectively. First, the WRST is applied to know their statistical significance. Then, the p-values of accuracy and NF are collected and given in Table 5.19. Whenever the p-values are less than 0.05, the null hypothesis is rejected. Hence, there is a significant difference. On the other hand, whenever the p-values are more than 0.05, the null hypothesis cannot be rejected. Hence, there is no significant difference.

Table 5.16. Results for HPSO of Experiment 2

DMs	Accuracy	NF
1	0.487603	1
2	0.545454	2
3	0.553719	3
4	0.561983	8
5	0.570247	12

Table 5.17. Mean Accuracy of both algorithms on all the seven datasets

Datasets	PSO Algorithm	HPSO Algorithm
Wine	0.863736	0.852718
WDBC	0.932421	0.91642
Ionosphere	0.871428	0.865934
Sonar	0.786116	0.789268
Libras Moment	0.749074	0.694951
Hill Valley	0.506562	0.538429
Musk 1	0.866947	0.868571

Table 5.18. Mean NF of both the algorithms on all seven datasets

Datasets	PSO Algorithm	HPSO Algorithm
Wine	6.461538	2.444444
WDBC	15.818181	2.777777
Ionosphere	17.727272	15.384615
Sonar	34	9.04
Libras Moment	42.533333	9.137931
Hill Valley	50.882353	10.2
Musk 1	102.92	52.285714

Table 5.19. P-values of WRST of accuracy and features on all the seven datasets

Datasets	P-Values of accuracy	P-Values of NF
Wine	0.459	0
WDBC	0.178	0
Ionosphere	0.62414	0.0455
Sonar	0.34	0
Libras Moment	0.414	0
Hill Valley	0.018	0
Musk 1	0.519	0

From Tables 5.17, 5.18, and 5.19, we get the following observations-

(1) The accuracy of both algorithms on the wine dataset is comparable. But we are getting less NF with a statistical significance when compared with the PSO.

(2) On the WDBC dataset, the accuracy of both algorithms is comparable. But we are getting less NF with a statistical significance when compared with the PSO.

(3) On the Ionosphere dataset, the accuracy of both algorithms is comparable. But we are getting less NF with a statistical significance when compared with the PSO.

(4) On the Sonar dataset, the accuracy of HPSO is higher when compared with PSO, but the difference is not statistically significant. We are getting less NF with a statistical significance when compared with the PSO.

(5) The accuracy of both algorithms is comparable to the Libras Moment dataset. But we are getting less NF with a statistical significance when compared with the PSO.

(6) On the Hill Valley dataset, the accuracy of HPSO is higher with statistical significance when compared with PSO. Also, we are getting less NF with a statistical significance when compared with the PSO.

On the Musk 1 dataset, the accuracy of HPSO is higher with statistical significance when compared with PSO. Also, we are getting less NF with a statistical significance when compared with the PSO.

5.7 Conclusion

This work proposes a hybrid metaheuristic algorithm, HPSO, using a cross-over operator and PSO. Here, a pseudo-code for finding the accuracy and NF, which is used as an objective function, has been developed. The proposed HPSO algorithm was applied to MOFS problems, and the performance of HPSO was compared against PSO using seven UCI datasets.

The experimental results demonstrate that the HPSO algorithm outperforms PSO on a MOFS problem. Hence, the proposed algorithm gives better accuracy on higher dimensional problems. But on low dimensional problems, the result is comparable. The proposed algorithm takes less NF on both higher and lower dimensional problems. Furthermore, statistical tests were conducted to support this conclusion. Therefore, the proposed HPSO algorithm has the potential to serve as an excellent MOO problem and to tackle FS problems.

In the future, we will expand its application to real-life problems such as machine learning, medical applications, financial fields, and engineering optimization tasks. Furthermore, we will also integrate the fuzzy concept and novel algorithms with other strategies to build a better optimizer.

Chapter 6

Innovative Hybrid Metaheuristic Algorithms: Exponential Mutation and Dual-Swarm Strategy for Hybrid Feature Selection Problem

6.1 Introduction

Metaheuristic algorithms are preferred over deterministic optimization algorithms due to their simplicity and ease of implementation in real-life scenarios. Various metaheuristic algorithms, such as BA[12], GA[10], Quantized Salp Swarm Algorithm[74] and others have been applied to various real-world problems[35].

The exploration and exploitation phases are common features in all metaheuristic algorithms[75]. During the exploration phase, the algorithm explores different regions of the solution space to avoid premature convergence or stagnation. The success of metaheuristic algorithms depends on maintaining a proper balance between exploration and exploitation, which is achieved by selecting appropriate parameters.

FS involves selecting a subset of original features that can achieve high accuracy in a classification problem. The primary goals of the FS problems are to improve accuracy[76] and reduce the NF. Several fields, including text mining, image processing, computer vision, industrial applications, bioinformatics, and others, use FS problems in various ways[77]. As per the NFL theorem[9] exploring novel and innovative metaheuristic algorithm is very important.

The chapter is organised as follows for the remainder of it. Some fundamental ideas needed for a thorough understanding of this chapter are explained in Section 6.2. The problem we have addressed in this chapter is explained in Section 6.3. Section 6.4 discusses the proposed PSOHHO and PSOHHO-V algorithms for the FS problem. Section 6.5 provides an explanation of the proposed algorithm's theoretical analysis. Several datasets are used in Section 6.6 to evaluate the proposed PSOHHO and PSOHHO-V algorithms with additional metaheuristic algorithms. Next, the statistical significance of them is examined. The conclusion of the current work has been provided in Section 6.7.

6.2 Background

To fully comprehend this chapter, it's essential to be familiar with fundamental concepts such as HHO, PSO, and GA, explained in Section 1.3. Moreover, a crucial awareness of FS problems, explained in Section 1.5, is critical.

6.3 Problem Formulation

This section provides a brief overview of the challenge addressed in this chapter. As per the NFL theorem, no optimization algorithm can be the most efficient for every optimization problem. There is a need for development of innovative and efficient metaheuristic algorithms that will provide researchers and experts with broader options for solving complex optimization problems.

In this chapter, two novel hybrid variants of metaheuristic algorithms are developed and discussed. In the novel variants an innovative EMO is introduced, to enhance the exploration capability. It includes an exponential function that determines mutation probability per particle based on its history. Also, the concept of dual-swarm strategy is introduced into this algorithm to foster diversity throughout the optimization process.

There is a considerable research gap as theoretical and mathematical analysis of most of the metaheuristic algorithms has yet to be proved or even discussed. To address this issue the Signature of the proposed algorithm has been developed, this helps in the theoretical analysis of the proposed algorithm.

Initially, the efficacy of the developed algorithms PSOHHO and PSOHHO-V are assessed on Benchmark Functions (BF) to validate their applicability to real-world problems. Subsequently, their application to FS problems aims to enhance CA and reduce the NF. For the FS problem, a combination of FE, WE, and NF are considered, and the proposed algorithms, PSOHHO and PSOHHO-V, are employed. Comparative analyses with other metaheuristic algorithms are conducted, and statistical significance is verified using Friedman's test and Wilcoxon Rank Sum Test (WRST). In this chapter, each row of the position matrix corresponds to a particle's position, where '0' signifies a feature not selected and '1' indicates feature selection. The quality of representation critically influences the effectiveness of the proposed method.

6.4 Proposed Hybrid Approaches (PSOHHO and PSOHHO-V)

Exploration and exploitation stand out as the primary techniques employed in the search for solutions within the solution space. Achieving an optimal answer requires a careful equilibrium between these two approaches to thoroughly navigate the solution domain. Traditional PSO algorithms and their modifications encounter challenges in maintaining this balance, resulting in limitations in generating effective solutions. The proposed algorithms help to address these issues. This algorithm employs two strategies to produce effective search results: 1) A dual-swarm technique is implemented and 2) PSO is subjected to an EMO.

When particles get stuck in local minima, they experience the mutation operator, which consists of two crucial components. First, in order to improve performance, the algorithm's overall mutation probability, represented by the letter mp_1^t, gradually decreases over time. Secondly, particles whose z^p has stayed stationary in recent iterations see an enhanced chance of mutation, and is accomplished by introducing mp_2^t. After that, Eq. (6.1) is utilized to determine mu_j^t. Throughout the iterations, the parameter mp_1^t decreases by a factor of λ_d. It is determined by applying Eq. (6.2), which is explained in[78] and Eq. (6.3) is used to compute mp_2^t.

$$mu_j^t = mp_1^t \times mp_2^t \tag{6.1}$$

$$mp_1^t = mp_1^{t-1} \times \lambda_d \tag{6.2}$$

$$mp_2^t = e^{\left(\frac{(NSI_j - NI)}{\lambda_m}\right)} \tag{6.3}$$

Here the number of iterations of particle p_j, during which it's z^p remains unchanged, is represented as NSI_j. Parameter NI indicates how many times the particle has been iterated to take benefit of its neighbourhood.

The quest for an optimal exploration-exploitation balance leads to the design of a dual-swarm strategy, wherein the total population N is divided into two halves. The first half of the particles team (Say Team A), the particles are guided using HHO and in the second half (Say Team B), particles are guided using PSO, for every iteration. Then at each iteration, the best values of teams A and B are compared. Then at the end of the termination criteria, the best value is selected as the global best solution. The global best solution gives us features with higher accuracy and less NF. In the second algorithm, a variant of PSOHHO (PSOHHO-V) is introduced, where the concept of crossover is applied. The flowchart of PSOHHO is given in Fig. 6.1 and the detailed explanation of the proposed algorithms are given below -

***Algorithm 1*:** Pseudo-code of PSOHHO and PSOHHO-V (with step 22).
INPUT: Initialization of the position of each particle and all the parameters are done.

***OUTPUT*:** Global best solution.

Pseudo-code

1) First, all the parameters of the PSO and HHO are initialized. The particle swarm (*N*) is given before the start of the experiment as it depends on experiment $h = 1$, $h_1 = 2$, and $h_2 = 2$ are the values of the parameter taken.
2) The particle's velocity and position are initialized by assigning random matrices.
3) In a corresponding solution matrix, the position of every particle is represented by the row of the corresponding position matrix.
4) The main iterative loop starts in this step.
5) Compute the FE (F_1), WE (F_2) and NF (F_3) of each particle using Eq. (4.1), Eq. (4.2) and Eq. (4.3) respectively
6) The objective function $\left((F_1 + F_2) - F_3\right)$ is treated as the fitness function of each particle.

7) Now first Team A and then Team B will come into play respectively.
8) $For\ i = 1: N/2$ (Say Team A)
9) Now, compute $|E\_E|$ from Eq. (1.3) to check whether to stay in exploration or exploitation mode.
10) If $|E\_E| > 1$, update the particle according to Eq. (1.1). This is the exploration phase.
11) If $|E\_E|$ and c are greater than or equal to 0.5.
then the particles are guided by Eq. (1.4)
12) Else if $|E\_E|$ is less than 0.5 and c is greater than or equal to 0.5.
then the particles are guided by Eq. (1.7)
13) Else if $|E\_E|$ is greater than or equal to 0.5 and c is less than 0.5.
then the particles are guided by Eq. (1.12)
14) Else if $|E\_E|$ and c are less than 0.5.
then the particles are guided by Eq. (1.15)
End
15) End
16) Eq. (6.1) is used to apply the EMO.
17) ***End (steps 8 to 16)***
18) Then we take the best value from the above steps (steps 8 to 16). (Say (Fit H)).
19) $For\ i = N/2: N$ (Say Team B)
20) Update the velocity and position of each particle using Eq. (1.16) and Eq. (1.17)
21) The personal best of each particle is updated and Eq. (6.1) is used to apply the EMO. Then the global best is updated.
End
22) Then the crossover operator is introduced (ONLY FOR SECOND ALGORITHM, KNOWN AS PSOHHO-V).
23) ***End (steps 18 to 23)***
24) Then we take the best value from the above steps (steps 18 to 23). (Say (Fit P)).
25) Now, Fit H and Fit P are compared. The higher fitness value will be treated as the global best.
26) End (***steps 4 to 25***). Here the main iteration loop stops.
27) If termination criteria is not achieved then the process keeps repeating. So steps 8 to 17 and steps 19 to 23, will be repeated again and again till the criteria is not satisfied.

28) After termination criteria is achieved, select the features using global best and its CA is reported.

Note: If Step 22 is included, then it is the PSOHHO-V algorithm, else it is the PSOHHO algorithm.

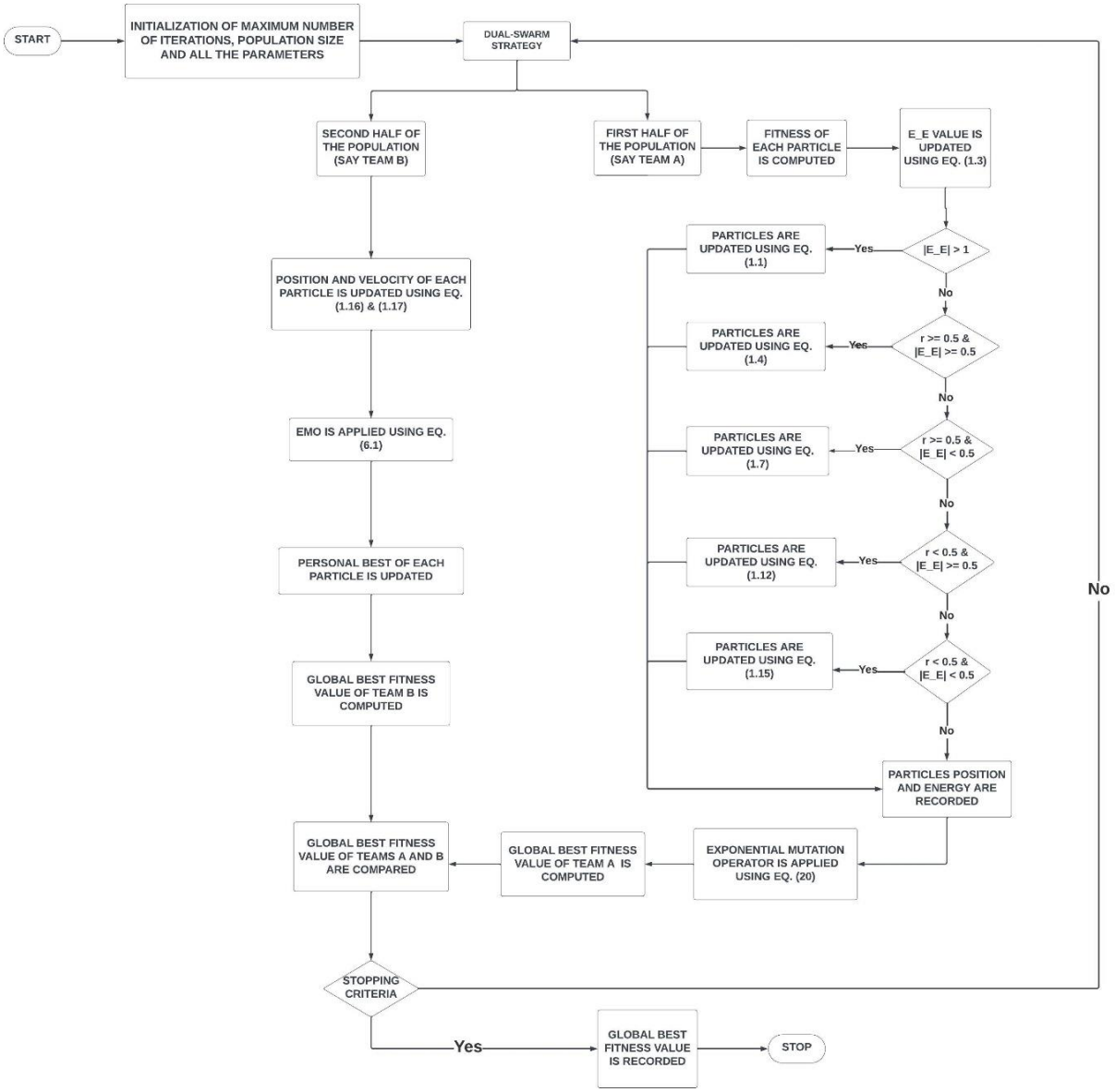

Fig. 6.1. Flowchart of PSOHHO

6.5 Theoretical Analysis of PSOHHO

With the aid of the Signature stated in Section 6.5.1 and the convergence discussed in Section 6.5.2, the theoretical analysis of the proposed algorithm is covered in this part.

6.5.1 Signature of the PSOHHO

An unbiased stochastic optimization technique generates a collection of positions that are statistically identical to those generated by an arbitrary search since every point in the search space has the same importance. Hence, it is ideal to identify the inherent bias of an optimization technique before calculating its efficiency. We must be vigilant for central bias, edge bias, none, or both. Let $f(x) = 1$ to observe the bias of optimization. The underlying bias of the algorithms is shared by many optimization techniques. Many optimization algorithms like the Marine Predators Algorithm, WOA, and GWO can be rejected because of their biases[79]. Fig. 6.2 displays the HHO Signature. The HHO algorithm is disallowed due to its center bias. To assess the bias of the suggested optimization algorithm (PSOHHO), we used $f(x) = 1$. We can see the PSOHHO signature in Fig. 6.3. The PSOHHO Signature is not skewed toward the center, axis, or region, as seen in Fig. 6.3. We can observe that the Signature of PSOHHO must be acknowledged and that it is far superior to the Signature of HHO.

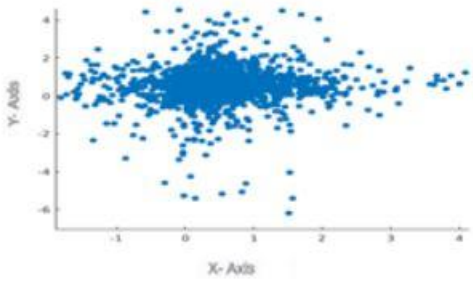

Fig. 6.2. Signature of HHO

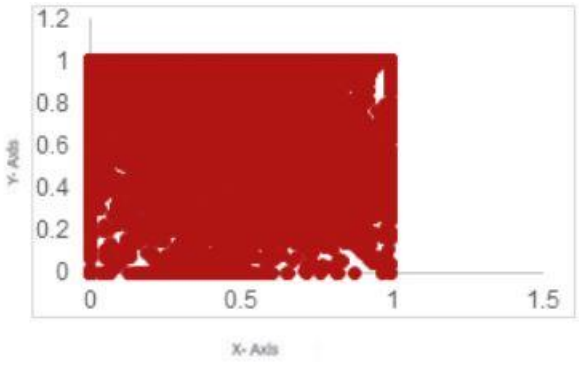

Fig. 6.3. Signature of PSOHHO

6.5.2 Convergence analysis of the PSOHHO

Convergent analysis of standard PSO has been carried out using the MC[18].

Convergent criteria

The convergence of an algorithm depends on the framework of the theoretical analysis of the algorithm and certain criteria; here, we have used the conditions given by Solis and Wets[80]. If a stochastic optimization T iterates for t iterations, then we can obtain the new solution $z(t+1)$ using the equation defined below-

$$z(t+1) = T(z(t), \acute{\varepsilon})$$

Where $\acute{\varepsilon}$ is the solution set.

To obtain global optimality, two conditions are necessary to achieve during the iterative process: -

Condition 1. [80]: The sequence $f\big(z(t)\big)$ should decrease when applying algorithm S. If I is a feasible solution space, there exists

$$C = \{c = (a_1, a_2, \dots, a_{\mathrm{n}}) |\ a_i \epsilon\ \mathrm{A}\}$$

$\acute{\varepsilon}\ \epsilon\ I$ such that

$$f\big(T(z, \acute{\varepsilon})\big) \leq f(\acute{\varepsilon})$$

Condition 2. [80] For all subsets $C\ \epsilon$ I with $v(C) > 0$, probability measure $u_{\mathrm{t}}(C)$ for iteration 't,' we have

$$Product \prod_{t=0}^{\infty} \big(1 - u_{\mathrm{t}}(C)\big) = 0$$

Criteria 1. If 'f' is a measurable and 'I' is a measurable subset, the algorithm will converge with probability one when 't' is sufficiently large if conditions 1 and 2 are satisfied. In other words, the algorithm can almost imply global convergence.

We consider the parameters h_1 and h_2 as constant for simplicity.

Now we will define the concepts required to prove the convergence of PSOHHO.

Definition 1. Particle's position z, Intensity E_I, particle's best position z^p and particle's velocity s forms a state, ad it is denoted as (z, s, E_I, z^p). All the possible states of all the individuals form state space and are denoted by $A = \{\mathrm{a} = (z, s, E\_I, z^p) |\ z, z^p\ \epsilon\ I\}$

Definition 2. Now we define the group status space as

$$C = \{c = (a_1, a_2, \dots, a_{\mathrm{n}}) |\ a_i \epsilon\ \mathrm{A}\}$$

We can observe from the above definitions that the group status space also contains the best positions. For all a_1 = (z_1, s_1, E_E_1, z^{p_1}) ∈ A and a_2 = (z_2 s_2, E_E_2, z^{p_2}) ∈ A, the state transition can be denoted by $F_a(a_1) = a_2$. Here $F_a$is the transition function from a_1 to a_2. Similarly, for all c_i =$\left(z_{i,1}, s_{i,2},\ E\_E_{i,3}, z_{i,4}^{p}\right)$ ∈ H, $\mathrm{F_c}(c_i) = c_j$.

Theorem 1. In the proposed algorithm, state a_1 is shifted to state a_2 in one step, and the transition probability (TP) is

$$\mathrm{P(F_A(a_1) = a_2) = P(z_1 \rightarrow z_2)P(s_1 \rightarrow s_2)P(E\_E_1 \rightarrow E\_E_2)P(z^{p_1} \rightarrow z^{p_2})}$$

Proof. The state of the particle is transferred from a_1 = (z_1, s_1, E_E_1, z^{p_1}) to a_2 = (z_2 s_2, E_E_2, z^{p_2}). Hence $z_1 \rightarrow z_2$, $s_1 \rightarrow s_2$, $E\_E_1E\_E_2, z^{p_1} \rightarrow z^{p_2}$ are transferred simultaneously. The probability of $P(F_A(a_1) = a_2)$ is

$$P(F_A(a_1) = a_2) = P(z_1 \rightarrow z_2)P(s_1 \rightarrow s_2)P(E\_E_1 \rightarrow E\_E_2)P(z^{p_1} \rightarrow z^{p_2})$$

Theorem 2. The PSOHHO's group state sequence is a finite homogeneous MC.

Proof. The number of iterations is finite, the population size is finite, and the search space is finite. So, z, s, E_E, z^p is also finite, which implies that each state a = (z, s, E_E, z^p) is also finite. Hence we can conclude that the state space is also finite.

The particle's position updates in every iteration of the algorithm. So, it cannot be an MC. However, if we can group the position, Escaping Intensity, personal best history, and velocity as one state $C(t)$. Then the other state, $C(t+1)$, is only linked with the state $C(t)$. Then the sequence $C(t)$ has proper MC properties.

The Transition Probability (TP) $P\left(\mathrm{F_c}\big(C(t-1)\big) = C(t)\right)$ from state $C(t-1)$ to $C(t)$ is computed by the TP of all the individuals in the group. From theorem 1, the TP is calculated by the joint probability of $P\big(z(t-1) \rightarrow z(t)\big)$, $P\big(s(t-1) \rightarrow s(t)\big)\, P\big(E\_E(t-1) \rightarrow E\_E(t)\big)P\big(z^p(t-1) \rightarrow z^p(t)\big)$. Also, $\big(z(t-1) \rightarrow z(t)\big)$, $P\big(s(t-1) \rightarrow s(t)\big)\, P\big(E\_E(t-1) \rightarrow E\_E(t)\big)P\big(z^p(t-1) \rightarrow z^p(t)\big)$ are only related to *z, s, E_E, z^p* at time *t.* Hence $P\left(\mathrm{F_c}\big(C(t-1)\big) = C(t)\right)$ is linked only to the state $a_i(t-1)$, $1 \le i \le N$. Hence the MC is finite. Again $P\left(\mathrm{F_A}\big(a(t-1)\big) = a(t)\right)$ is independent of time $t-1$ according to Theorem 1. Similarly, $P\left(\mathrm{F_c}\big(C(t-1)\big) = C(t)\right)$ is independent of time $t-1$. Hence these finite MCs are homogeneous.

Global Convergence Analysis of PSOHHO Using MC

Let us state the following three theorems:

Theorem 3.[81] If $V \subset C$, then there does not exist any closed set J other than C satisfying $J \cap V = \emptyset$.

Theorem 4.[81] Suppose there is a non-empty set z of. an MC with no closed set E satisfying $Y \cap \mathbf{E} = \emptyset$, then $\lim_{t\to\infty} P(z^k = j) = \pi_j$ only if $j \in Y$ and $\lim_{t\to\infty} P(x^k = j) = 0$ only if j does not belong to Y.

Theorem 5.[81] When the number of iterations increases and becomes sufficiently large, the group state sequence converges to the optimal state set.

Using the above theorems, we will prove that PSOHHO converges globally.

Theorem 6. PSOHHO, with the MC model defined above, converges to the global point.

Proof. A stochastic optimization algorithm will converge to the global optimality if it meets both Conditions 1 and 2, according to Criteria 1. In essence, the first condition (Condition 1) can ensure that the stochastic optimization algorithm's fitness value is declining. Furthermore, the prior theorems also establish that the group state sequence will converge to the optimal set after sufficient repetitions, i.e., the probability of failing to find the globally optimal solution is asymptotically zero. By extension, this proves that the second convergence criterion is likewise met. Therefore, we can conclude that PSOHHO will converge to the optimal point with probability one.

6.6 Results and Discussions

The efficiency of the proposed algorithms is explained in Section 6.6.1, by comparing them with other four metaheuristic algorithms on ten BF. Then its application to the FS problem is explained in Section 6.6.2. The maximum number of iterations, swarm size, and dimension we have taken are 200, 10, and 30 respectively. The codes are run on Windows 10 (64 bit), RAM 8 GB.

6.6.1 BF

All six algorithms including the proposed algorithms (PSOHHO and PSOHHO-V) are converted into binary form after using a well-defined transformation function and then applied on ten BF[2]. The BF consists of unimodal (UP), multimodal (MP), and fixed dimensional multimodal problem (FDMP) and are given in Table 6.1. The mean fitness value of all the algorithms on the BF are given below in Table 6.2 along with Friedman ranking. Then to check the statistical significance of the proposed algorithms Friedman test and Wilcoxon Rank Sum Test (WRST) are applied. Then the P-values of WRST are given in brackets in Table 6.2, when PSOHHO-V is compared with other algorithms on all the BF. The efficiency of the algorithm is affected by the dimension of optimization problems. The scalability analysis is done to observe the efficiency of the proposed algorithms. So, we have tested the algorithms on the BF with dimensions 30, 100, 500, 1000. Here best fitness values of BF are written in bold in Table 6.2 and we are getting the following observations-

1. When PSOHHO-V is compared with PSOHHO, four times PSOHHO-V gives significantly better (i.e. on BF_1, BF_3, BF_6, BF_7) fitness value than PSOHHO. PSOHHO-V is giving better fitness value than PSOHHO twice but the difference is not significant (i.e. on BF_2 and BF_8), zero times significantly worse and remaining four times same (i.e. on $BF_4, BF_5, BF_9, BF_{10}$).
2. When PSOHHO-V is compared with HHO, four times PSOHHO-V is giving significantly better (i.e. on BF_2, BF_3, BF_6, BF_7) fitness value than HHO. PSOHHO-V is giving better fitness value than PSOHHO twice but the difference is not significance (i.e. on BF_1 and BF_8), zero times significantly worse and remaining four times same (i.e. on $BF_4, BF_5, BF_9, BF_{10}$).
3. When PSOHHO-V is compared with GA, six times PSOHHO-V is giving significantly better (i.e. on $BF_1, BF_2, BF_4, BF_6, BF_7, BF_8$) fitness value than GA. Once better but the difference was not significance (i.e. on BF_3), zero times significantly worse, and remaining three times same (i.e. on BF_5, BF_9, BF_{10}).
4. When PSOHHO-V is compared with PSO, seven times PSOHHO-V is giving significantly better (i.e. on $BF_1, BF_2, BF_4, BF_6, BF_7, BF_8$) fitness value than PSO, zero times significantly worse and remaining three times same (i.e. on BF_5, BF_9, BF_{10}).

5. When PSOHHO-V is compared with SSA, seven times PSOHHO-V is giving significantly better (i.e. on BF_1, BF_2, BF_4, BF_6, BF_7, BF_8) fitness value than SSA, zero times significantly worse and the remaining three times same (i.e. on BF_5, BF_9, BF_{10}).

Table 6.1. Collection of BF [2]

BF	Type
$BF_1(x) = \sum_{i=1}^{n} x_i^2$	UP
$BF_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	UP
$BF_3(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	UP
$BF_4(x) = \sum_{i=1}^{n} (x_i + 0.5)^2$	UP
$BF_5(x) = 418.9829 * n - \sum_{i=1}^{n} (x_i)(sin\ sin\ \sqrt{\lvert x_i \rvert})$	MP
$BF_6(x) = \sum_{i=1}^{n} [x_i^2 - 10\ cos\ cos\ 2\pi x_i\ + 10]$	MP
$BF_7(x) = -20e^{\left(-0.2\sqrt{\left(\frac{1}{n}\right)\sum_{i=1}^{n} x_i^2}\right)} - e^{\left(\left(\frac{1}{n}\right)\sum_{i=1}^{n} cos cos\ 2\pi x_i\right)} + 20 + e$	MP
$BF_8(x) = \left(\frac{1}{4000}\right)\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \left(\frac{x_i}{\sqrt{i}}\right) + 1$	MP
$BF_9(x) = \left(\left(\frac{1}{500}\right) + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2} (x_i - a_{ij})^6}\right)^{-1}$	FDMP
$BF_{10}(x) = \left(x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6\right)^2 + 10\left(1 - \frac{8}{\pi}\right) cos\ x_1 + 10$	FDMP

Table 6.2: Mean fitness value of 50 runs on the BF when dimension is 30 along with P-values in bracket.

BF	PSOHHO	PSOHHO-V	HHO	GA	PSO	SSA
BF_1	29.1 (0.046)	**29.36**	29.2 (0.029)	27.24 (0)	23.46 (0)	20.39 (0)
BF_2	29.68 (0.0356)	**29.88**	29.38 (0.025)	27.36 (0)	25.56 (0)	23.47 (0)
BF_3	2360 (0)	**2540**	2370 (0)	2520 (0.26)	2070 (0)	1730 (0)
BF_4	**1040** (0.07)	**1040**	**1040** (0.23)	1020 (0)	996 (0)	974 (0)
BF_5	**12600** (0)	**12600**	**12600** (0)	**12600** (0)	**12600** (0)	**12600** (0)
BF_6	524 (0)	**599**	29.18 (0)	27.46 (0)	402 (0)	581 (0)
BF_7	4.33 (0)	**4.63**	3.59 (0)	3.47 (0)	4.49 (0)	4.56 (0)

BF_8	0.89 (0.056)	**0.89**	0.89 (0.18)	0.88 (0)	0.84 (0)	0.83 (0)
BF_9	**14.56** (1)	**14.56**	**14.56** (1)	**14.56** (1)	**14.56** (1)	**14.56** (1)
BF_{10}	**55.60** (1)	**55.60**	**55.60** (1)	**55.60** (1)	**55.60** (1)	**55.60** (1)
RANK	2	1	3	4	6	5

Note: Bold values indicate the best value, and P-values are given in the corresponding bracket

6.6.2 FS Application

To implement the proposed algorithms, the given algorithms are converted into binary form using transformation. Then the proposed algorithms PSOHHO and PSOHHO-V are compared to four different algorithms (HHO, GA[82], SSA[15], PSO) on seven datasets[39] (Ionosphere, Wine, Breast Cancer Wisconsin, Sonar, Libras Movement, Hill Valley, Musk 1) using UCI Machine Learning repository. On each dataset, we have run the MATLAB code fifty times. Then we have recorded their CA and NF. We have also checked whether the difference is statistically significant or not by applying Friedman's test and WRST. Here Friedman test is used to express the FAR of all compared methods more clearly for further statistical evaluation. Here have chosen the common KNN classifier. Here the K-fold cv is 1.

Here have chosen the common KNN Classifier. Here the K-fold cv is 1. The graphs of the CA of the algorithms on Musk 1, Hill Valley, Libras Movement, Sonar, and Ionosphere datasets are shown in Figs 6.4, 6.6, 6.8, 6.10, and 6.12. In these Figs, PSO, PSOHHO, PSOHHO-V, GA, HHO, and SSA are represented by green colour, red colour, cyanide colour, yellow colour, blue colour, and black colour respectively. The graphs of the NF of the algorithms on Musk 1, Hill Valley, Libras Movement, Sonar, and Ionosphere datasets are shown in Figs 6.5, 6.7, 6.9, 6.11, and 6.13. In the second row of these Figs, PSO, PSOHHO, GA, HHO, SSA, and PSOHHO-V are represented by bars (1, 2, 3, 4, 5, and 6) respectively.

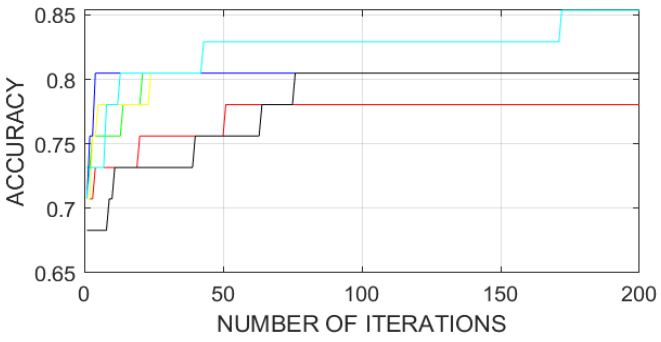

Fig. 6.4. CA on MUSK 1 dataset.

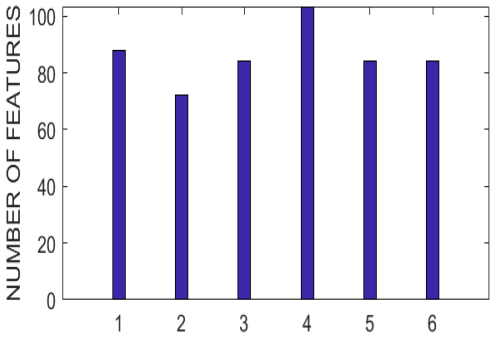

Fig. 6.5. NF on MUSK 1 dataset.

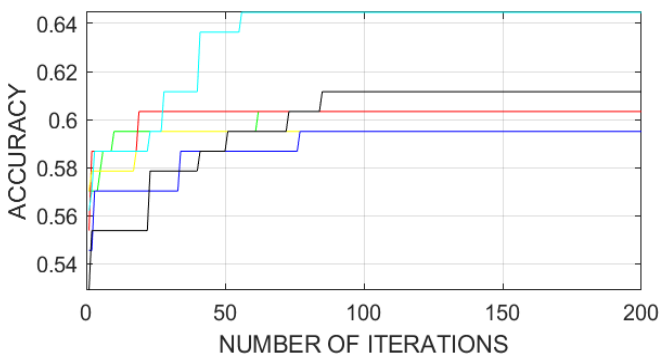

Fig. 6.6. CA on Hill Valley dataset.

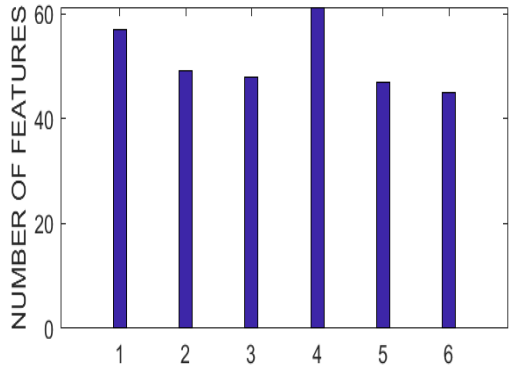

Fig. 6.7. NF on Hill Valley dataset.

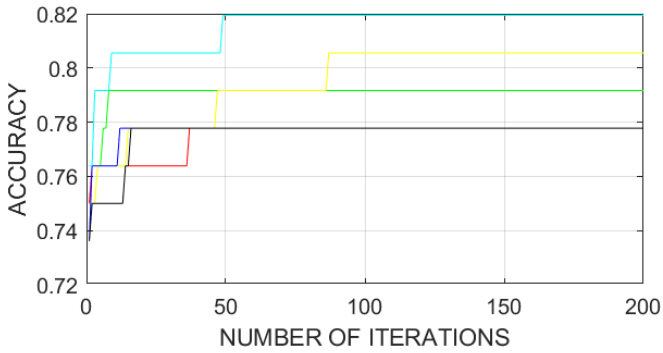

Fig. 6.8. CA on Libras Movement dataset.

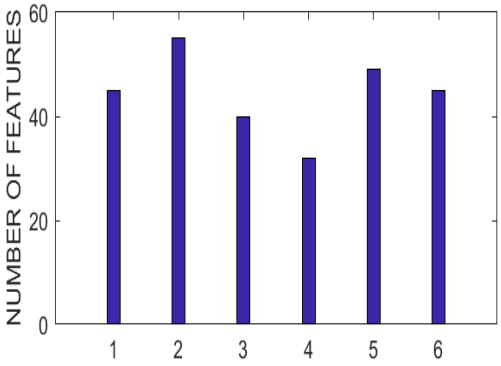

Fig.6. 9. NF on Libras Movement dataset.

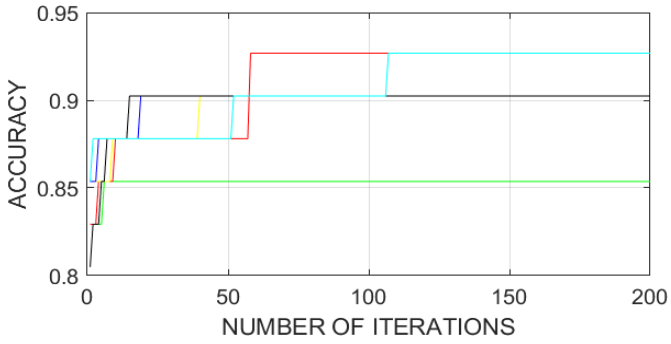

Fig. 6.10. CA on Sonar dataset.

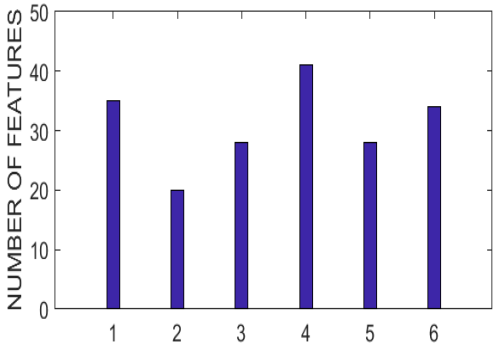

Fig. 6.11. NF on Sonar dataset.

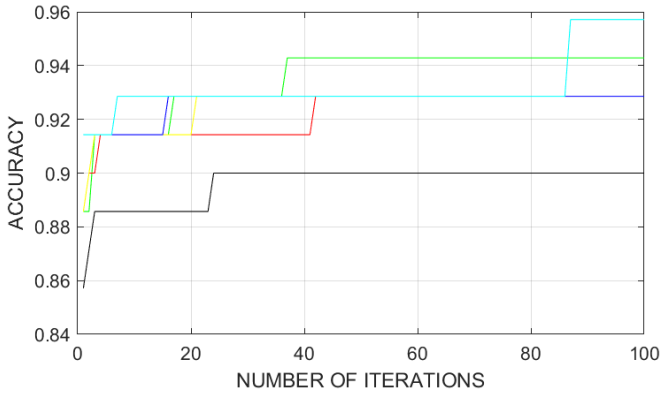

Fig. 6.12. CA on Ionosphere dataset.

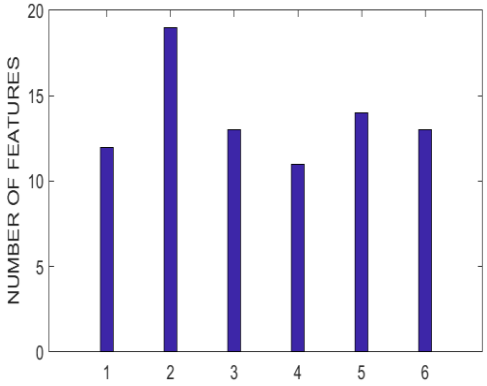

Fig. 6.13. NF on Ionosphere dataset.

All six algorithms are applied to the FS problem fifty times on all seven datasets. The mean CA and NF of all six algorithms on all the datasets are given in Tables 6.3 and 6.4 respectively. Here best values are written in bold. The Friedman test and WRST are applied to test the statistical significance. Then the P-values of WRST when PSOHHO-V is compared with other algorithms on all the seven datasets with respect to CA and NF are given in brackets of Tables 6.3 and 6.4 respectively.

Table 6.3: Mean CA of the all the six algorithms on all the seven datasets.

Dataset	PSOHHO	PSOHHO-V	HHO	GA	PSO	SSA
Wine	**1 (1)**	**1**	**1** (1)	**1** (0)	0.989 (0.01)	0.999 (0.863)
Wdbc	0.962 (0)	**0.984**	0.962 (0)	0.966 (0)	0.953 (0)	0.958 (0)
Ionosphere	0.94 (0.412)	**0.982**	0.941 (0.211)	0.949 (0.799)	0.943 (0)	0.933 (0.001)
Sonar	0.942 (0)	**0.967**	0.944 (0)	0.968 (0.488)	0.929 (0)	0.93 (0)
Libras	0.816 (0)	**0.837**	0.816 (0)	0.829(0.381)	0.818 (0)	0.816 (0)
Hill Valley	0.581 (0)	**0.94**	0.584 (0)	0.595 (0)	0.589 (0)	0.582 (0)
Musk 1	0.940 (0)	0.954	0.941 (0)	**0.955** (0.359)	0.944 (0.002)	0.94 (0)
FAR	4.214	1.5	3.786	1.786	4.286	5.429
Rank	4	1	3	2	5	6

Note: Bold values indicate the best value, and P-values are given in the corresponding bracket

Table 6.4: Mean NF of the all the six algorithms on all the seven datasets.

Dataset	PSOHHO	PSOHHO-V	HHO	GA	PSO	SSA
Wine	4.14 (0.004)	5.24	**4.12 (0.001)**	5.98 (0.004)	6.36 (0)	5.76 (0.043)
Wdbc	13 (0)	**12.26**	13.14 (0)	12.58 (0)	15.36 (0)	14.38 (0)
Ionosphere	14.46 (0.168)	15.58	14.78 (0.608)	**13.28 (0.001)**	15.68 (0.82)	15.08 (0.354)
Sonar	29.12 (0.743)	28.98	30.6 (0.36)	**28.08 (0.211)**	30.04 (0.193)	28.36 (0.469)
Libras	43.32 (0.877)	43.42	45.58 (0.038)	**42.8 (0.694)**	43.92 (0.608)	43.96 (0.715)
Hill Valley	46.2 (0.679)	47.12	**43.54 (0.121)**	46.64 (0.669)	47.4 (0.662)	47.26 (0.672)
Musk 1	94.98(0.0001)	87.72	91.58 (0.012)	**82.48 (0.001)**	84.64 (0.005)	82.54 (0.001)
FAR	3	3.286	3.714	2	5.143	3.857
Rank	3	2	4	1	6	5

Note: Bold values indicate the best value, and P-values are given in the corresponding bracket

Here we are observing from Table 6.3, PSOHHO-V is giving best CA. Here we are observing from Table 6.4, PSOHHO and PSOHHO-V are not taking less NF. Whenever the p-values are less than 0.05, null hypothesis is rejected. Hence the difference is significant.

Whenever the p-values are more than 0.05, null hypothesis cannot be rejected. Hence the difference is not significant.

From Table 6.3, we are getting the following observations with respect to the CA-

(1) When PSOHHO-V is compared with PSOHHO, five times PSOHHO-V is significantly better (WDBC, Sonar, Libras Moment, Hill Valley, Musk 1) mean CA than PSOHHO, once better (Ionosphere) without any significance difference. And once same CA (WINE).

(2) When PSOHHO-V is compared with HHO, five times PSOHHO-V is significantly better (WDBC, Sonar, Libras Moment, Hill Valley, Musk 1) mean CA than HHO, once better (Ionosphere) without any significance difference. And once same CA (WINE).

(3) When PSOHHO-V is compared with GA, twice PSOHHO-V is significantly better (WDBC, Hill Valley) mean CA than GA, three times better (Ionosphere, Sonar, Libras Moment) without any significance difference. And once same CA (Wine).

(4) When PSOHHO-V is compared with PSO, seven times PSOHHO-V is significantly better (Wine, WDBC, Ionosphere, Sonar, Libras Moment, Hill Valley, Musk 1) mean CA than PSO.

(5) When PSOHHO-V is compared with SSA, six times PSOHHO-V is significantly better (Ionosphere, WDBC, Sonar, Libras Moment, Hill Valley, Musk 1) mean CA than SSA, once better (Wine) without any significance difference.

From Table 6.4, we are getting the following observations with respect to the NF-

(1) When PSOHHO-V is compared with PSOHHO, two times PSOHHO-V is significantly less (WDBC, Musk1) NF than PSOHHO, once less NF (Sonar) without any significance difference, once more NF (Wine) with significance difference and three times more NF (Ionosphere, Libras Moment, Hill Valley) without any significance difference.

(2) When PSOHHO-V is compared with HHO, two times PSOHHO-V is significantly less (WDBC, Libras Moment) NF than PSOHHO, twice less NF (Sonar, Musk1) without any significance difference, once more NF (WINE) with significance difference and two times more NF (Ionosphere, Hill Valley) without any significance difference.

(3) When PSOHHO-V is compared with GA, two times PSOHHO-V is significantly less

(Wine, WDBC) NF than PSOHHO, once more NF (Ionosphere) with significance difference and four times more NF (Sonar, Libras Moment, Hill Valley, Musk 1) without any significance difference.

(4) When PSOHHO-V is compared with PSO, two times PSOHHO-V is significantly less (Wine, WDBC) NF than PSOHHO, four times less NF (Ionosphere, Sonar, Libras Moment, Hill Valley) without any significance difference and once more NF (MUSK 1) without any significance difference.

(5) When PSOHHO-V is compared with SSA, two times PSOHHO-V is significantly less (Wine, WDBC) NF than PSOHHO, two times less NF (Libras Moment, Hill Valley) without any significant difference and three times more NF (Ionosphere, Sonar, Musk 1) without any significance difference.

6.7 Conclusion

The proposed algorithms PSOHHO and PSOHHO-V were evaluated based on statistical measures and convergence rate, and their performance was tested on ten BF. The complete experimental results indicate that the established algorithms outperform other optimizers regarding searchability and convergent speed when solving global optimization problems. Furthermore, statistical tests were conducted to support this conclusion. The proposed algorithms PSOHHO and PSOHHO-V were also applied to feature selection problems, and their performance was compared against other algorithms using seven UCI datasets. Again, the experimental results demonstrate that PSOHHO and PSOHHO-V are giving better results than the other metaheuristic algorithms on the high-dimensional and medium-dimensional datasets. But on low dimensional datasets, the results are not much useful. Since the proposed algorithms are giving better results on high and low-dimensional datasets and practical applications of FS problems involve large datasets, therefore, the proposed algorithms have the potential to tackle FS problems.

We will investigate the use of this algorithm in MOO challenges in the future and extend its applicability to real-world issues like financial, medicinal, and engineering optimization challenges. To create a stronger optimization technique, we will also combine the innovative algorithm with other techniques.

Chapter 7

A Hybrid Swarm Optimization with Trapezoidal Fuzzy Number and Pentagonal Fuzzy Number using Benchmark functions

7.1 Introduction

There is a growing trend of Hybridising two metaheuristic algorithms. Hybridization of metaheuristic algorithms is performed to increase the performance of metaheuristic algorithms on real-world problems as many real-world optimization problems are highly non-linear and highly dimensional. Metaheuristic algorithms are developed and applied to real-life problems in place of deterministic optimization methods, due to their simplicity and easy implementation. So, they are most helpful when we want to find the approximate results in a given time frame. It helps to eliminate the drawbacks of convergence and stagnation. Also, they are sensitive to parameter tuning.

Theoretical and mathematical examinations of numerous metaheuristic algorithms are notably lacking. There exists a notable gap in the investigation of the convergence patterns of these algorithms. Considering issues such as premature convergence and susceptibility to local optima in many metaheuristic algorithms, a thorough mathematical analysis is essential. For proving the convergence of metaheuristic algorithms[83], different methods have been proposed, such as spectral radius for PSO[84] and MC for GA[85,86]. MC is a random process with a strong capability for probabilistic analysis and convergence analysis of randomized algorithms. It has been successfully implemented on the ABC algorithm, the PSO[87], the ACO, and the SA[88]. Hence, in our study, the convergence of the PSOMHHO using the MC property is proved.

Our work aims to develop a hybrid metaheuristic algorithm PSOMHHO with TFN and PFN as parameters. The concept of the Signature of an algorithm is also introduced to check the intrinsic bias of an algorithm. With so many varieties of heuristic and metaheuristic algorithms getting discovered yearly, the Signature of an algorithm plays a critical role. The Signature of an algorithm indicates whether every point in the search space is given equal importance. Here PSOMHHO is applied to nine standard BF. The significance of the statistical difference in their fitness value is checked using MWUT and Friedman's test.

The chapter is organised as follows for the remainder of it. Some fundamental ideas needed for a thorough understanding this chapter are explained in Section 7.2. The problem we have addressed in this chapter is explained in Section 7.3. Section 7.4 discusses the proposed PSOMHHO algorithm. Section 7.5 explains the proposed algorithm's theoretical analysis. Several BF are used in Section 7.6 to evaluate the proposed PSOMHHO algorithm with additional metaheuristic algorithms. Next, the statistical significance of them is examined. The conclusion of the current work has been provided in Section 7.7.

7.2 Background

To fully comprehend this chapter, it's essential to be familiar with fundamental concepts such as HHO, PSO, and GA, explained in Section 1.3. Moreover, a proper understanding of fuzzy number, TFN and PFN is also critical which is explained in Section 1.3.

7.3 Problem Formulation

This section provides a brief overview of the challenge addressed in this chapter. As per the NFL theorem, no optimization algorithm can be the most efficient for every optimization problem. There is a need for development of innovative and efficient metaheuristic algorithms that will provide researchers and experts with broader options for solving complex optimization problems.

In this chapter innovative hybrid variant of metaheuristic algorithms is developed and discussed. In this variant, the concepts of TFN and PFN are introduced. There is a considerable research gap as theoretical and mathematical analysis of most of the metaheuristic algorithms has yet to be proved or even discussed. To address this issue the convergence and signature of the proposed algorithm have been proved, this helps in the theoretical analysis of the proposed algorithm. The effectiveness of the developed algorithms PSOMHHO is assessed on BF on Windows 10 (64 bit), RAM 8 GB to validate their applicability to real-world problems.

The BF consisting of unimodal, fixed dimensional multimodal BF is taken from the literature[89,90]. The detailed mathematical description of the BF is explained in the previous chapter in Table 6.1. Unimodal problem (UP) is helpful for testing convergence and exploitive strength. In addition, testing the explorative strength and the ability to not fall into the trap of local optimal point, Multimodal problem (MP) is helpful. In every experiment applied on the BF, every algorithm is applied 50 times with different dimensions after converting into a binary form using a well-defined transformation function. The maximum number of iterations and the swarm size are 200 and 10, respectively. Then the statistical performance measure means is used to access the optimization ability of PSOMHHO. Then to check the statistical significance of the PSOMHHO Friedman test and MWUT are applied. The level of significance for the MWUT is set to 0.05.

7.4. Proposed Algorithm (PSOMHHO)

In the proposed PSOMHHO algorithm, the parameters h_1 and h_2 of PSO are PFN, and the parameter h is TFN. The fuzzy parameters are then defuzzfied. When the termination criteria are achieved, the best value is selected as the global best solution. Pseudocode is given in Section 7.4.1. The flowchart of PSOMHHO is given in Fig. 7.1.

7.4.1 Pseudo code

1) Initialization of the position of each particle and all the parameters of HHO and PSO are done.

2) The parameter h is TFN, and the parameters h_1 and h_2 are PFN. Then the fuzzy parameters are defuzzfied.
3) The particle's velocity and position are initialized by assigning random matrices.
4) In a corresponding solution matrix, the position of every particle is represented by the row of the corresponding position matrix.
5) The main iterative loop starts in this step.
6) Each particle's fitness value is calculated using the BF.
7) Here nine BF are used, and each will be treated as the objective function.
8) Now, compute $|E\_E|$ from Eq. (1.3) to check whether to stay in exploration or exploitation mode.
9) If $|E\_E| > 1$, then update the particle according to Eq. (1.1). This is the exploration phase.
10) If $|E\_E|$ and c are greater than or equal to 0.5.
    Then the particles are guided by Eq. (1.4)
11) If $|E\_E|$ is less than 0.5 and c is greater than or equal to 0.5.
    Then the particles are guided by Eq. (1.7)
12) If $|E\_E|$ is greater than or equal to 0.5 and c is less than 0.5.
    Then the particles are guided by Eq. (1.12)
13) If $|E\_E|$ and c are less than 0.5.
    Then the particles are guided by Eq. (1.15)
14) The particle's velocity and particle's position are updated using the Eqns. (1.16) and (1.17).
15) Each particle's personal best is updated.
16) Then the mutation operators are introduced.
17) Then the global best solution is updated.
18) The highest fitness value will be the global best solution.
19) End (***steps 5 to 18***). Here the main iteration loop stops.
20) When the termination criteria are achieved, the iterative process stops, and the global best solution is reported.

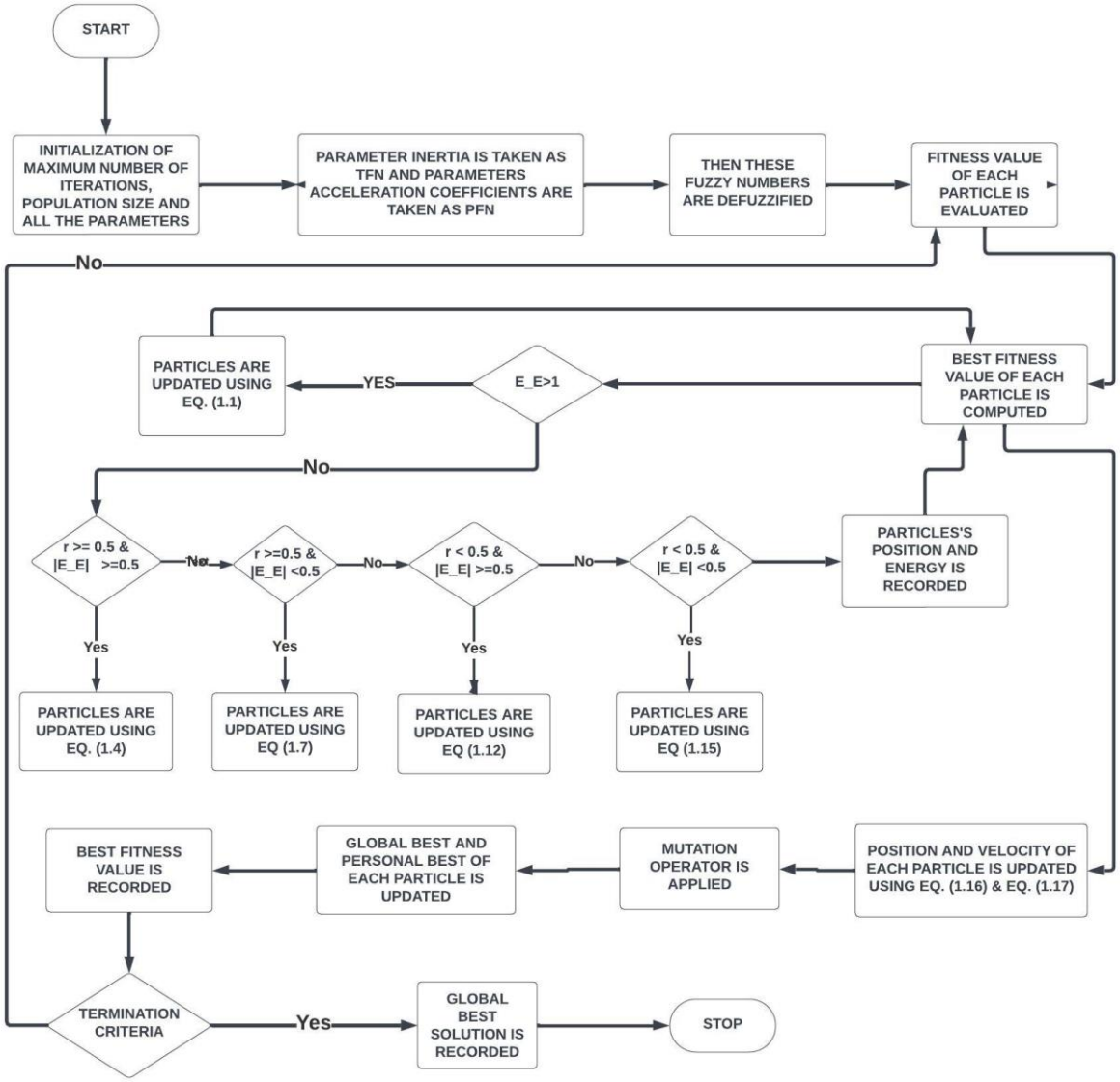

Fig. 7.1. Flowchart of PSOMHHO

7.5 Theoretical Analysis of PSOMHHO

In this section, the theoretical analysis of the purposed algorithm is discussed with the help of Signature discussed in Section 7.5.1 and convergence in Section 7.5.2.

7.5.1 Signature of the PSOMHHO

For an unbiased stochastic optimization algorithm, the equivalent significance accorded to each point in the search space can result in positions similar to those formed by a arbitrary search. Hence, understanding the inherent bias of an optimization algorithm becomes

paramount before evaluating its performance. For a thorough exploration of the concepts, the preceding chapter provides a detailed discussion in Section 6.5.1.

The Signature of HHO is given in Fig 6.2. The algorithm HHO is rejected because it is center-biased. We have taken $f(x) = 1$ to check the bias of the proposed optimization algorithm (PSOMHHO). The Signature of PSOMHHO is given in Fig 7.2. We can observe from Fig 7.2 that PSOMHHO's Signature is not biased toward the center, axis, or region. We can observe that the Signature of PSOMHHO has to be accepted, and the Signature of PSOMHHO is significantly better than HHO. Here we have taken the values of h_1 and h_2 as 1.5.

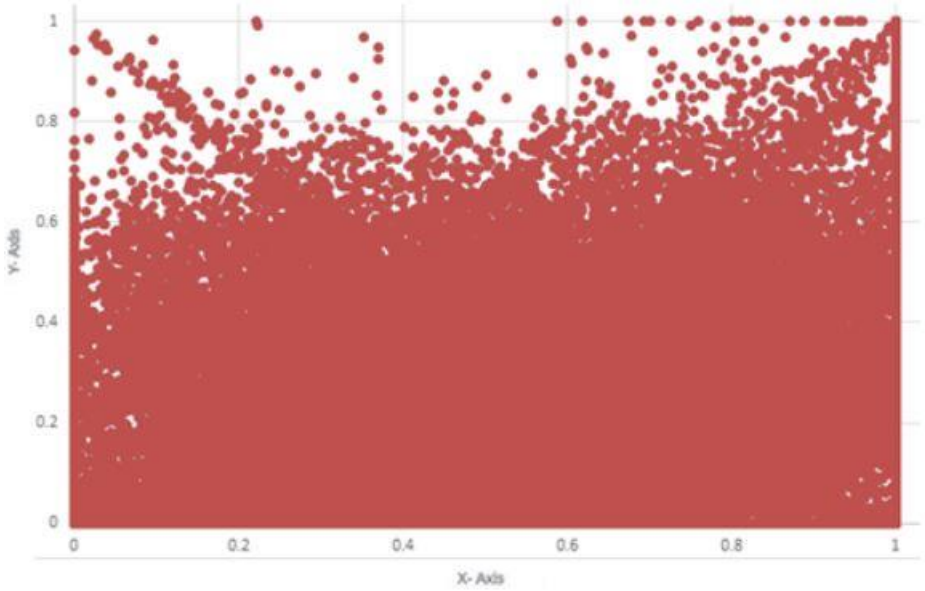

Fig. 7.2. Signature of PSOMHHO

7.5.2 Convergence of PSOMHHO

The concepts and methods used to prove convergence of PSOHHO in the previous chapter in Section 6.5.2 can also be extended to prove convergence of PSOMHHO.

7.6 Results and Discussions

The efficiency of PSOMHHO is checked in this section by comparing it with four other metaheuristic algorithms. Here PSO, PSOMHHO, GA, HHO, and Salp Swarm Algorithm

(SSA) are represented by green colour, red colour, cyanide colour, yellow colour, blue colour, and black colour, respectively. Furthermore, the convergence curves of BF $BF_1, BF_2, BF_3, BF_4, BF_5, BF_6, BF_7, BF_8$ and BF_9 are denoted by Figs 7.3, 7.4, 7.5, 7.6, 7.7, 7.8, 7.9, 7.10, and 7.11 respectively.

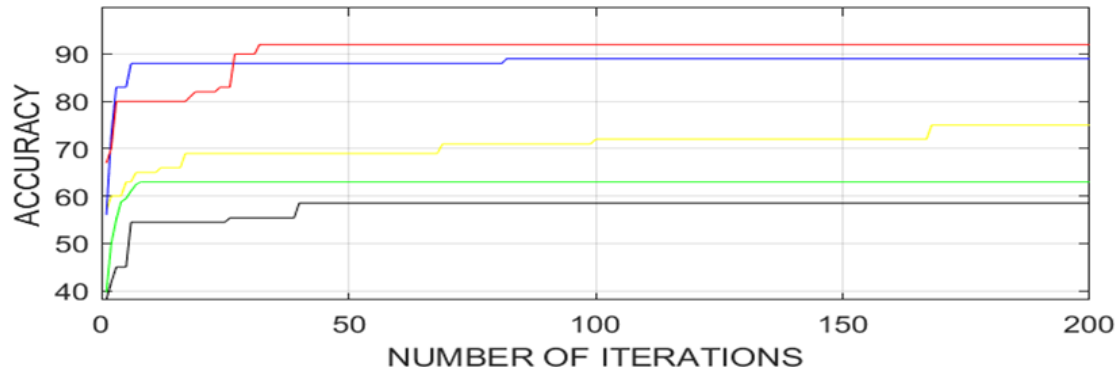

Fig. 7.3. Graph of BF_1

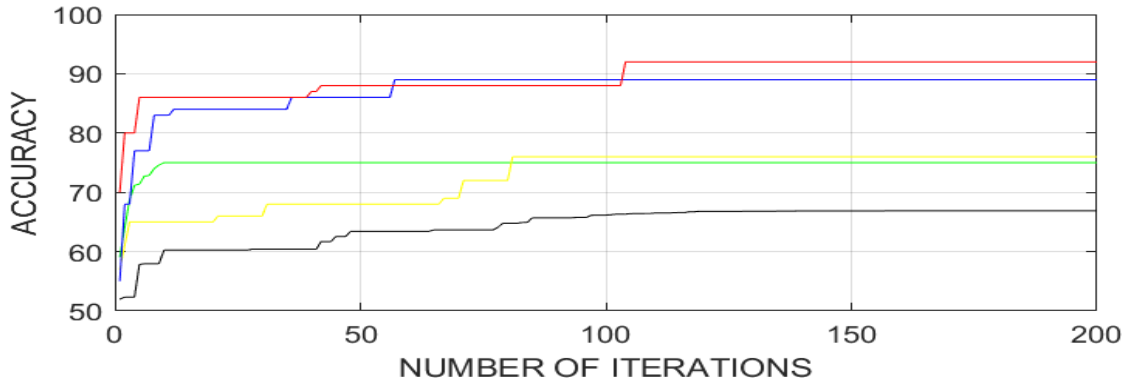

Fig. 7.4. Graph of BF_2

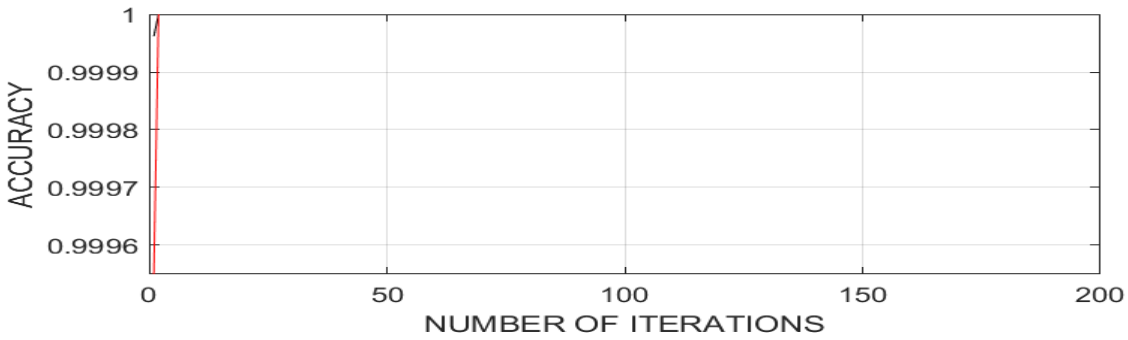

Fig. 7.5. Graph of BF_3

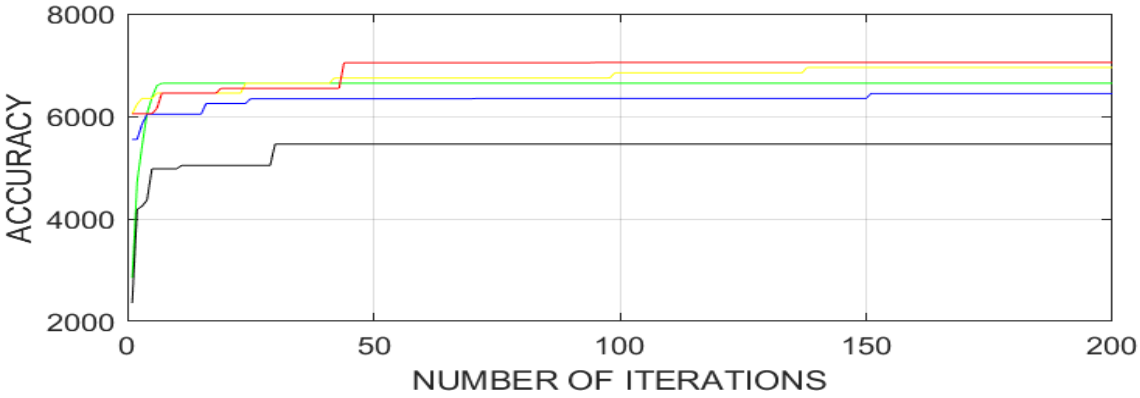

Fig. 7.6. Graph of BF_4

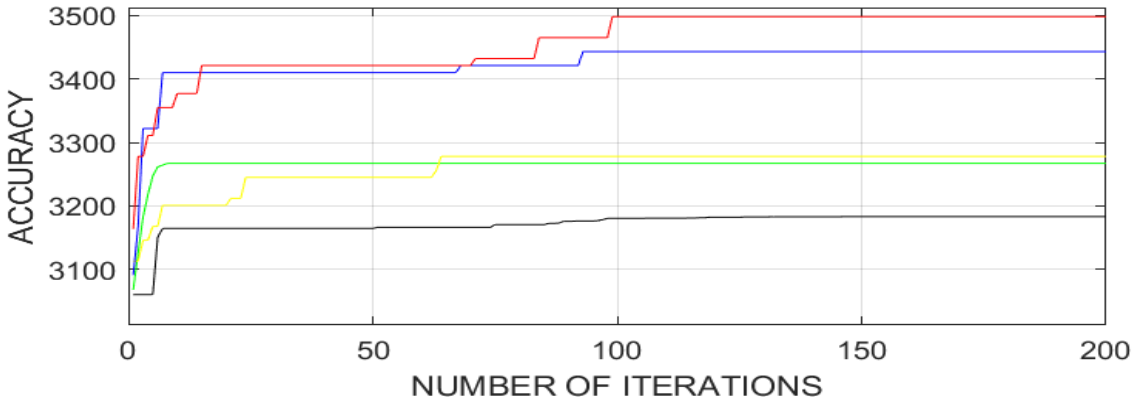

Fig. 7.7. Graph of BF_5

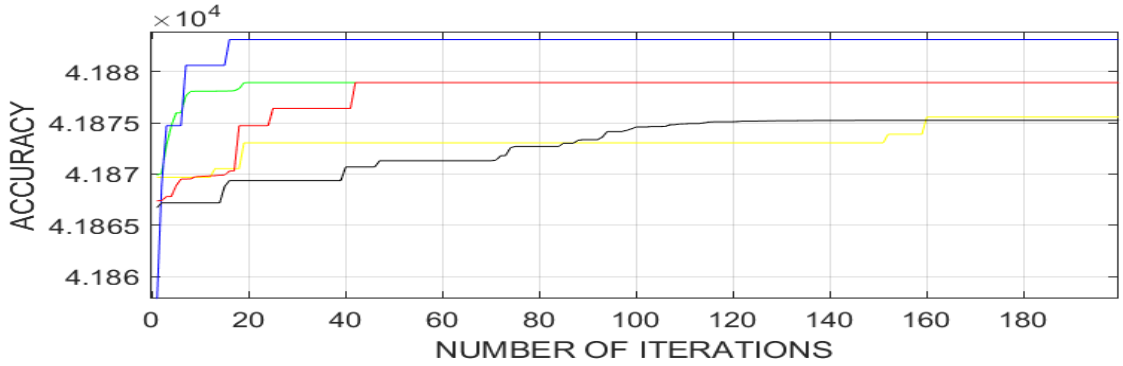

Fig. 7.8. Graph of BF_6

Fig. 7.9. Graph of BF_7

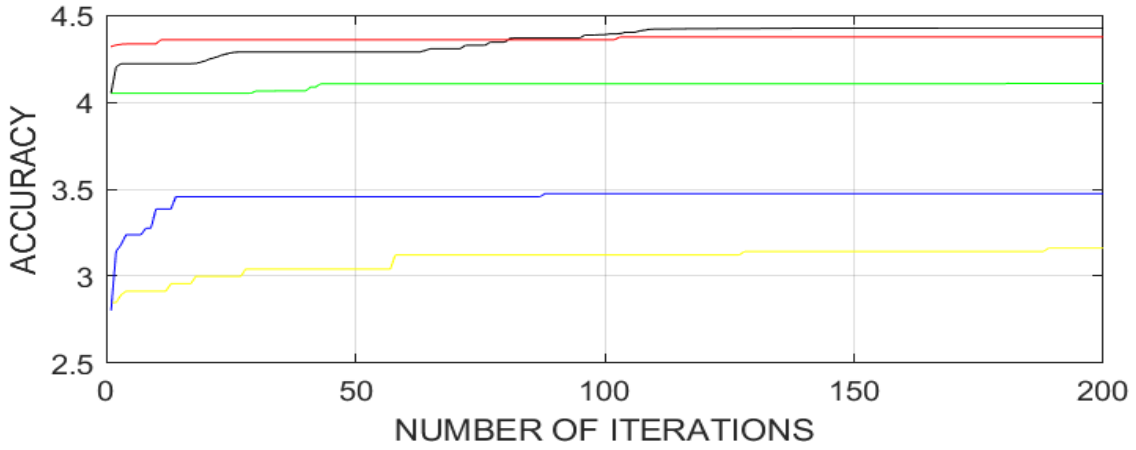

Fig. 7.10. Graph of BF_8

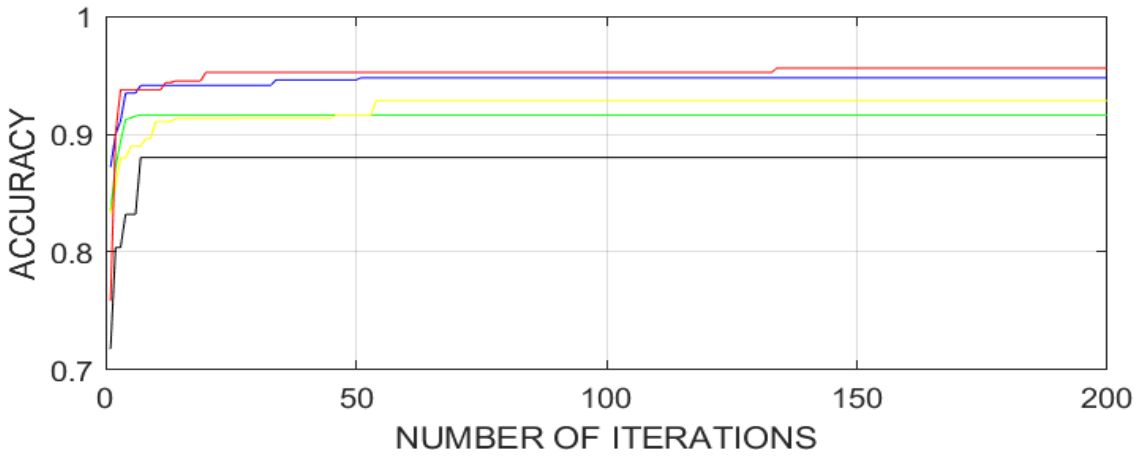

Fig. 7.11. Graph of BF_9

Table 7.1. The mean fitness value of 50 runs when the dimension is 100.

BF	PSOMHHO	HHO	GA	PSO	SSA
BF_1	**91.06**	*90.54*	74.24	65.52	56.98
BF_2	**91.36**	*89.96*	74.06	71.46	66.00
BF_3	**1**	**1**	***1***	**1**	**1**
BF_4	**7.00e+03**	6.595e+03	*6.976e+03*	6.252e+03	5.154e+03
BF_5	**3.468e+03**	*3.459e+03*	3.28e+03	3.245e+03	3.183e+03
BF_6	**4.1887e+04**	**4.188e+04**	**4.187e+04**	**4.187e+04**	**4.187e+04**
BF_7	**1.998e+03**	89.94	74.18	1.167e+03	*1.877e+03*
BF_8	*4.373*	3.457	3.157	4.159	**4.381**
BF_9	**55.60211**	**55.60211**	**55.60211**	**55.60211**	**55.60211**
Rank	**1**	**2**	**3**	**4**	**5**

Note: Bold values indicate the best value and italic indicates second best value

We can make the following observations from Table 7.1-

(1) PSOMHHO gives the best solution on eight BF and once the second best among the nine BF.
(2) HHO is giving the best solution on three BF and three times second best among the nine BF.
(3) GA gives the best solution on three BF and once second best among the nine BF.
(4) PSO gives the best solution on three BF and no second best among the nine BF.
(5) SSA gives the best solution on four BF and once the second best among the nine BF.

From Table 7.1, Friedman test, and MWUT, we observe that PSOMHHO is giving the most superior result, followed by HHO, GA, PSO, and SSA.

7.6.2 Scalability analysis of PSOMHHO

The efficiency of the algorithm is affected by the dimension of optimization problems. So, we have tested the algorithms on the BF with dimensions 100, 500, and 1000. Hence scalability analysis is done to observe the efficiency of PSOMHHO. In every size we are observing, PSOMHHO is giving better results.

7.7 Conclusion

This experiment is tested with different dimensions. Here we can observe that the Signature of PSOMHHO is significantly better than HHO. Also, convergence proof helps establish the mathematical strength of PSOMHHO.

We have taken BF with different properties like UP, MP, and Fixed-dimension multimodal BF. Here we are getting that the mean value of the fitness value of the PSOMHHO algorithm with dimensions 100, 500, and 1000 is more when compared with other algorithms like HHO, GA, PSO, and SSA. Then statistical significance tests like MWUT and Friedman test are applied. Then their results are recorded. Based on statistical measures, statistical significance tests like MWUT and Friedman test, and the convergence behaviour, we observe that PSOMHHO is giving better fitness value when compared with other algorithms like HHO, GA,

PSO, and SSA on the different types of BF. So, this algorithm can be applied to other real-world problems.

In the future, we can take different varieties of fuzzy numbers and check the effect on our proposed algorithm PSOMHHO. We can also focus on the stability analysis of PSOMHHO. Other variants of PSOMHHO can be developed using different cross-over operators, and then compared with the original algorithm (PSOMHHO). Here the fuzzy parameters play a very critical role. Also, we can take other different varieties of fuzzy numbers.

Chapter 8

Bluefin Trevally Optimizer (BTO): A Metaheuristic Algorithm Using Fuzzy Logic Controller for Feature Selection Problem

8.1 Introduction

The preference for metaheuristic algorithms over deterministic optimization algorithms for their uncomplicatedness and ease of execution in practical applications. Various metaheuristic algorithms, such as the GA, BA[12], SMA[13], WOA[2], SSA[15], and others have been applied to various real-world problems[91,92]. The most common feature of all the metaheuristic algorithms is exploration and exploitation[75]. The purpose of the exploration phase is to explore various regions of the solution space. So, the optimizer should have random nature in the exploration phase to randomly generate solutions to different areas of the problem topography during the early steps of the search process[93]. Hence proper use of randomized operators is advisable. Normally, the exploitation stage comes after the exploration stage and focuses on the neighbourhood of better quality solutions. It exploits the better solutions obtained in the exploration phase. Also, there is a possibility of the solutions getting trapped in local optima (premature convergence). Hence, the most important feature of all metaheuristic algorithms is maintaining a proper balance between exploration and exploitation.

Fuzzy logic, rooted in the FYS theory introduced by Zadeh[94], is a valuable tool for representing information using fuzzy if-then rules. It is particularly effective in handling linguistic information and improves numerical computation by employing linguistic labels assigned by membership functions[20,95]. FYSs have been widely used in various fields, including pattern recognition[96–98].

The structure of the chapter for its remaining sections is outlined as follows. Section 8.2 presents essential concepts crucial for a comprehensive grasp of this chapter. The problem under consideration is elucidated in Section 8.3, while Section 8.4 digs into the discussion of the proposed Fuzzy BTO algorithms. The theoretical underpinnings of the proposed algorithm are expounded upon in Section 8.5. To assess the performance of the proposed Fuzzy BTO algorithms against other metaheuristic algorithms, various BF are employed in Section 8.6. Subsequently, a meticulous examination of their statistical significance is conducted. Finally, Section 8.7 encapsulates the conclusions drawn from the present work.

8.2 Background

Understanding this chapter in its entirety requires a robust knowledge of fundamental concepts like fuzzy numbers, HHO, PSO, and GA, all of which are explained in Section 1.3. Additionally, an elementary understanding of FS problems, as explained in Section 1.5, is critical.

8.3 Problem Formulation

In this section, we get a brief introduction to the problem addressed in this chapter. The NFL theorem establishes that no single optimization algorithm can universally outperform others across all problems. Hence, there arises a necessity for the creation of innovative and effective metaheuristic algorithms. Such innovations would offer researchers and experts a diverse array of tools to tackle intricate optimization challenges.

In this chapter four innovative hybrid variants of metaheuristic algorithms are developed and discussed. In the innovative variants, the parameters of metaheuristic algorithms are dynamically changed using Fuzzy Logic Controller (FLC). A notable research gap exists, with many metaheuristic algorithms lacking thorough theoretical and mathematical analysis. To bridge this gap, this chapter explores the convergence and signature aspects of the proposed algorithms, offering a foundation for the theoretical analysis of the newly introduced algorithmic variants.

Initially, the effectiveness of the developed Fuzzy BTO algorithms and their variations are evaluated on BF to confirm their utility in addressing real-world challenges. Subsequently, their application to FS problems is pursued with the objective of improving CA and diminishing the NF. Considering a combination of FE, WE, and NF, the proposed algorithms are applied and subjected to comparative evaluations against other metaheuristic algorithms. Statistical significance is rigorously verified through Friedman's test and the Kruskal-Wallis Test (KWT).

8.4 Proposed Approach (Fuzzy BTO)

The suggested Fuzzy BTO algorithm uses a fuzzy approach to adapt the algorithmic parameters to changing population conditions. Since the parameters $h_1$and h_2 account for the movement of the particles, they are selected to be adjusted dynamically using FLC.

Algorithm performance measures such as diversity of the swarm, iteration, and average error at one point in the execution of the algorithm need to be considered to evaluate the algorithm. In our work, all the above are considered for the fuzzy systems to modify the parameters dynamically at each iteration of the algorithm. For measuring the iteration of the algorithm, it will be used as a percentage, and it can be represented as Eq. (8.1). The iteration will be considered “low” when the algorithm starts. It will be regarded as “high” when the algorithm is about to be completed.

$$\text{Iteration} = \left(\frac{current\ iteration}{maximum\ number\ of\ iteration}\right) \tag{8.1}$$

The diversity measures the degree of dispersion of the particle; when the particles are closer, there is less diversity, and when the particles are separated, the diversity is high. The diversity equation can be considered as the average Euclidean distance between each particle and the best particle as given below:

$$\text{Diversity} = \frac{\sum_{k=1}^{n_s}\sqrt{\sum_{l=1}^{n_x}(z_{kl}(t)-z_l(t))^2}}{n_s} \tag{8.2}$$

While applying fuzzy BTO to the FS problem we will also take fitness and NF as inputs for the fuzzy system. The inputs (iteration, diversity, accuracy, and NF) are granulated into three triangular membership functions. Each output is granulated in five triangular membership functions of the output variables h_1 and h_2 respectively.

Hence to design a fuzzy system that will dynamically adjust the parameters, we will construct two models for BF and four models for the FS problem. The four fuzzy models that follow Mamdani fuzzy system have two inputs and two outputs ($h_1$and h_2). The rules of each fuzzy system must be designed in such a way that the early iteration of the fuzzy BTO algorithms must be explored and then eventually exploited.

The first fuzzy system for BTOF1 has iteration and diversity as inputs, whose membership functions are triangular fuzzy numbers. The second fuzzy system for BTOF2 has accuracy and iteration as inputs, but the membership functions are TFN. The rules of both the fuzzy systems (BTOF1 and BTOF2) are given below:

1. If both iteration and diversity are low, then h_1 is high and $h_2$is low.
2. If iteration and diversity are medium and low, then $h_1$is medium-high and $h_2$is medium-low.
3. If iteration and diversity are high and low, then $h_1$is medium and $h_2$is high.
4. If iteration and diversity are low and medium, then $h_1$is medium-high and $h_2$is medium.
5. If both iteration and diversity are medium then $h_1$is medium and $h_2$is medium.
6. If iteration and diversity are high and medium, respectively, then $h_1$is medium-low and h_2 is medium-high.
7. If iteration and diversity are low and high, respectively, then $h_1$is medium-high and $h_2$is medium-low.
8. If iteration and diversity are medium and high, then $h_1$is medium-low and $h_2$is medium-high.
9. If both iteration and diversity are high then $h_1$is low and $h_2$is high.

The third fuzzy system for BTOF3 has iteration and accuracy as inputs, whose membership functions are triangular fuzzy numbers. It applies only to the FS problem, and its rules are given below:

1. If accuracy and iteration are high and low, respectively, then $h_1$is low and $h_2$is medium.
2. If accuracy and iteration are high and medium, respectively, then $h_1$is medium-low and $h_2$is medium-high.
3. If both accuracy and iteration are high, then $h_1$is low ,and $h_2$is high.
4. If accuracy and iteration are medium and low, respectively, then $h_1$is medium-low and $h_2$is medium-high.
5. If both accuracy and iteration are medium, then $h_1$is medium and $h_2$is medium.
6. If accuracy and iteration are medium and high, then $h_1$is medium and $h_2$is high.
7. If both accuracy and iteration are low, then $h_1$is high , and $h_2$is medium-low.
8. If accuracy and iteration are low and medium, respectively, then $h_1$is medium-high and $h_2$is medium.
9. If accuracy and iteration are low and high, then $h_1$is high and $h_2$is low.

The fourth fuzzy system for BTOF4 has iteration and NF as inputs, whose membership functions are triangular fuzzy numbers. It applies only to the FS problem, and its rules are given below:

1. If both iteration and NF are low, then h_1 is high and $h_2$is low.
2. If iteration and NF are medium and low, respectively, then $h_1$is medium-high and $h_2$is medium-low.
3. If iteration and NF are high and low, then $h_1$is medium and $h_2$is high.
4. If iteration and NF are low and medium, then $h_1$is medium-high and $h_2$is medium.
5. If both iteration and NF are medium then $h_1$is medium and $h_2$is medium.
6. If iteration and NF are high and medium, respectively, then $h_1$is medium-low and $h_2$is medium-high.
7. If iteration and NF are low and high, then $h_1$is medium-high and $h_2$is medium-low.
8. If iteration and NF are medium and high, respectively, then $h_1$is medium-low and $h_2$is medium-high.
9. If both iteration and NF are high, then $h_1$is low and $h_2$is high.

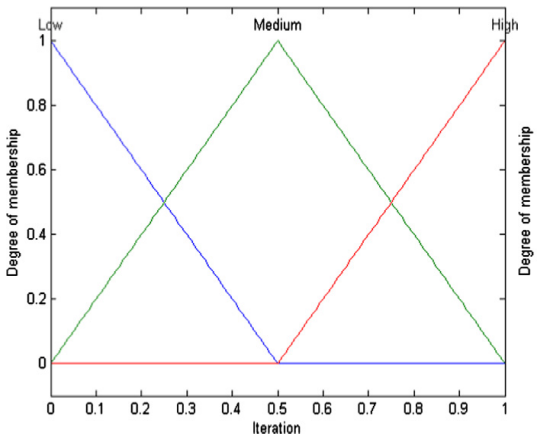

Fig. 8.1 Input 1: Diversity

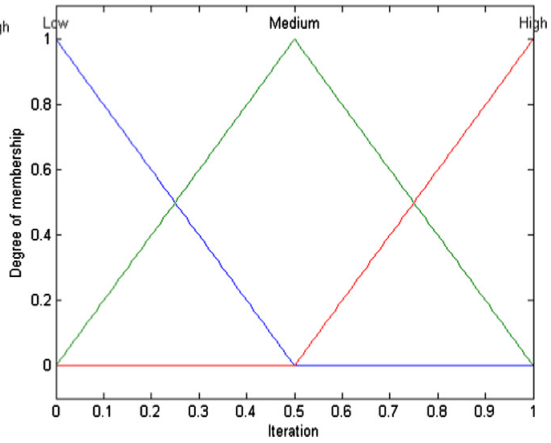

Fig. 8.2 Input 2: Iteration

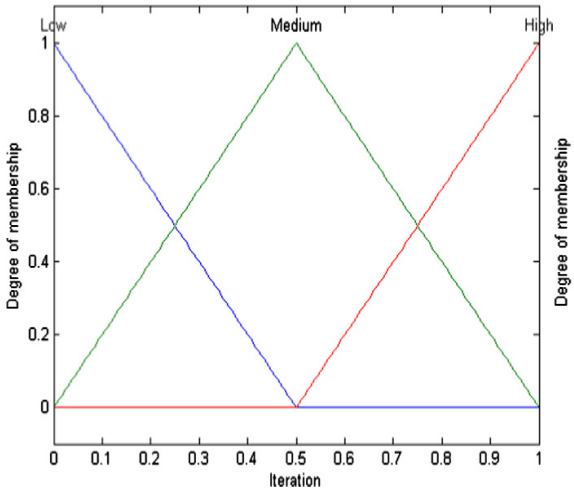

Fig. 8.3 Input 3: Accuracy

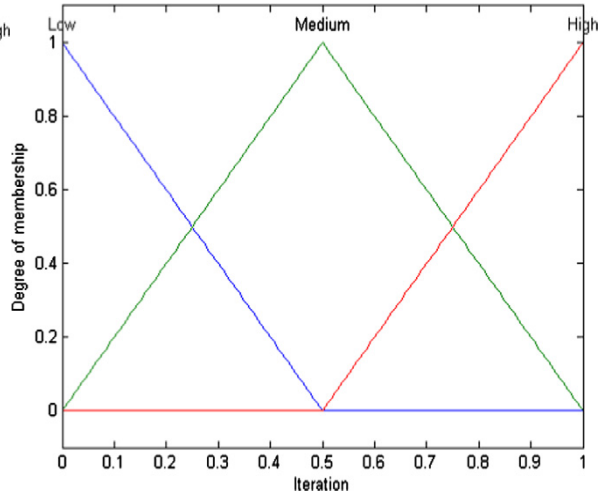

Fig. 8.4 Input 4: NF

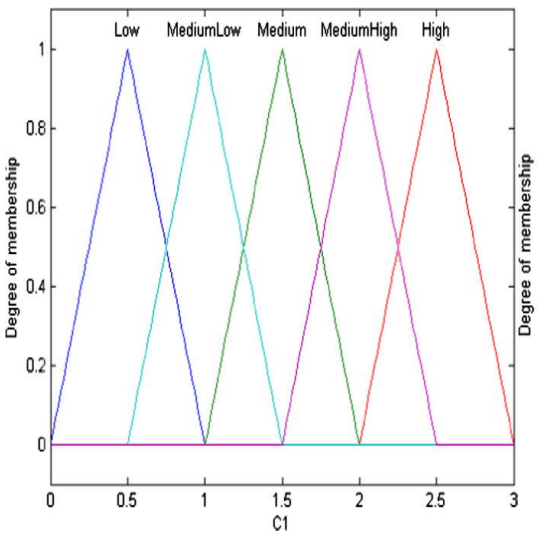

Fig. 8.5: Output 1: $\mathbf{h_1}$

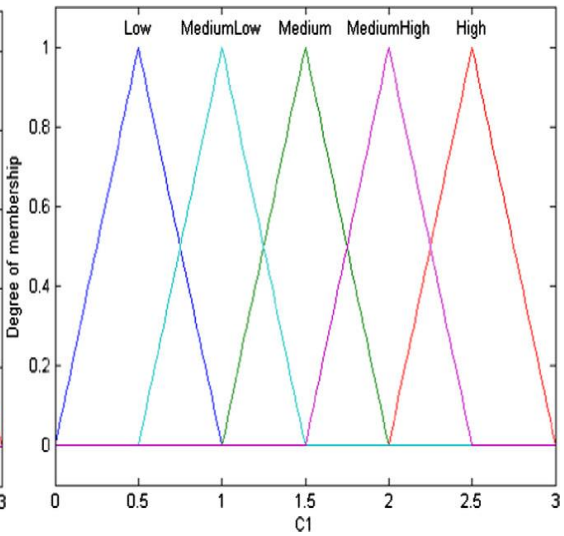

Fig. 8.6 Output 2: $\mathbf{h_2}$

Pseudo-Code

In this section, the pseudo-code of the proposed fuzzy BTO is given:

INPUT: Initialization of the position of each particle and all the parameters are done.

OUTPUT: Global best solution

1) The particle's velocity and position are initialized by assigning random matrices.
2) In a corresponding solution matrix, the position of every particle is represented by the row of the corresponding position matrix.
3) The main iterative loop starts in this step.
4) Each particle's fitness value is calculated using the BF.
5) Here, nine BF are used, each treated as the objective function.
6) Now, compute $|E\_E|$ from Eq. (1.3) to check whether to stay in exploration or exploitation mode.
7) If $|E\_E| > 1$, update the particle according to Eq. (1.1). This is the exploration phase.
8) If $|E\_E|$ and c are greater than or equal to 0.5.
   Then the particles are guided by Eq. (1.4)
9) Else if $|E\_E|$ is less than 0.5 and c is greater than or equal to 0.5.
   Then the particles are guided by Eq. (1.7)
10) Else if $|E\_E|$ is greater than or equal to 0.5 and c is less than 0.5.
    Then the particles are guided by Eq. (1.12)
11) Else if $|E\_E|$ and c are less than 0.5.
    Then the particles are guided by Eq. (1.15)

    End

End

12) The particle's velocity and particle's position are updated using Eq. (1.16) and (1.17).
13) The parameters h_1 and h_2 are dynamically adapted using FLC. The parameters follow the rules given in Section 3.2. Hence different fuzzy models are developed.
14) Each particle's personal best is updated.
15) Then, the mutation operators are introduced.
16) Then, the global best solution is updated.
17) The highest fitness value will be the best global solution.
18) End (***Steps 3 to 16***). Here the main iteration loop stops.

19) When the termination criteria are achieved, the iterative process stops, and the global best solution is reported.

First, the basic BTO algorithm has been applied. Then at step 15, the parameters h_1 and h_2 are dynamically changed using FLC (as mentioned in Section 3.2). Finally, we develop fuzzy BTO algorithms (BTOF1, BTOF2, BTOF3, and BTOF4) depending on different inputs and rules.

8.5 Mathematical Analysis of Fuzzy BTO

The proposed algorithm can be analysed using the concept of signature. All the points in the search space are equally important, so an unbiased stochastic optimization algorithm generates a collection of statistically identical positions to a random search. So before evaluating the performance of an optimization algorithm, it is ideal to obtain an idea of the intrinsic bias of the optimization algorithm. The detailed concepts are discussed in Chapter 6 in Section 6.5.1.

Many optimization algorithms like MPA, WOA, SMO, and GWO can be rejected because of their biases [79]. We have to check the bias of the proposed optimization algorithm Fuzzy BTO (BTOF1). The Signature of BTOF1 is given in Fig 8.7. We can observe from Fig 8.7 that BTOF1's Signature is not biased toward the center, axis, or region. Therefore, we can observe that the Signature of BTOF1 has to be accepted.

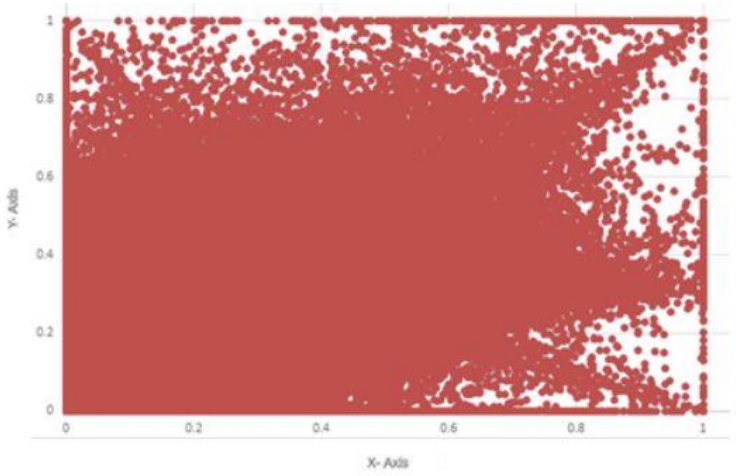

Fig. 8.7. Signature of BTOF1

8.6 Results and Discussions

BTOF1 and BTOF2 were applied to a set of nine BF, including both unimodal and fixed dimensional multimodal BF taken from the literature[90], as given in Chapter 6 in Table 6.1. The maximum number of iterations and swarm size were 200 and 10, respectively. The experiments were conducted on a Windows 10 (64-bit) system with 8GB of RAM. First, statistical performance measures were used to assess the optimization ability of BTOF1 and BTOF2, the KWT was applied to check the statistical significance.

To assess the practicality of Fuzzy BTO, we apply it to the feature selection problem. The proposed algorithms BTOF1, BTOF2, BTOF3, and BTOF4 are compared to three different algorithms (HHO, SSA, PSO) on seven datasets[39] (Ionosphere, Wine, Breast Cancer Wisconsin, Sonar, Libras Movement, Hill Valley, Musk 1) using UCI Machine Learning repository. For classification, we used the commonly employed KNN classifier with K-fold CV set to 1.

8.6.1 Results and Discussions on BF

Each algorithm was evaluated 20 times on every BF with different dimensions after converting them into the binary form using a well-defined transformation function. The effectiveness of BTOF1 and BTOF2 is evaluated by comparing their performance with four other metaheuristic algorithms, namely PSO, GA, HHO, and Salp Swarm Algorithm (SSA), which are represented by green, yellow, blue, and black colours, respectively. The proposed algorithms BTOF1 and BTOF2 are represented by red and cyanide colours. The convergence curves of the BF are presented in Figs 8.8 to 8.16.

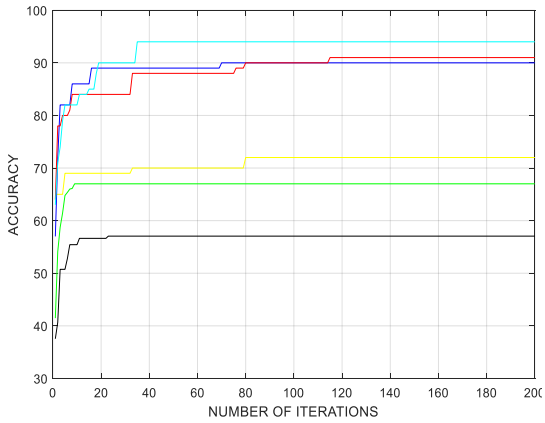

Fig. 8.8. Graph of $\boldsymbol{BF_1}$

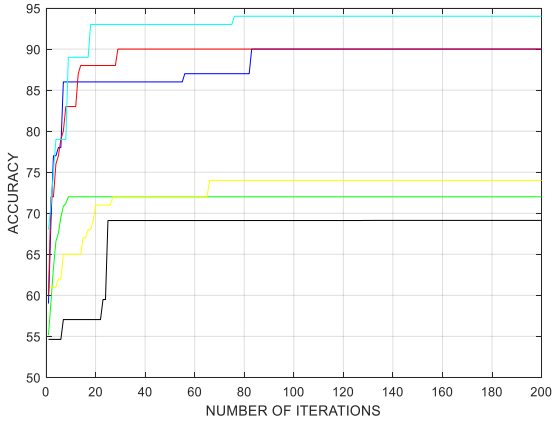

Fig. 8.9. Graph of $\boldsymbol{BF_2}$

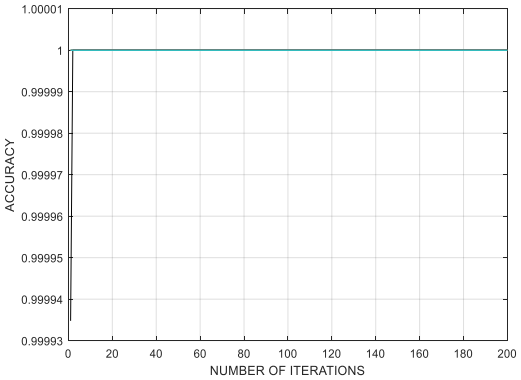

Fig. 8.10. Graph of $\boldsymbol{BF_3}$

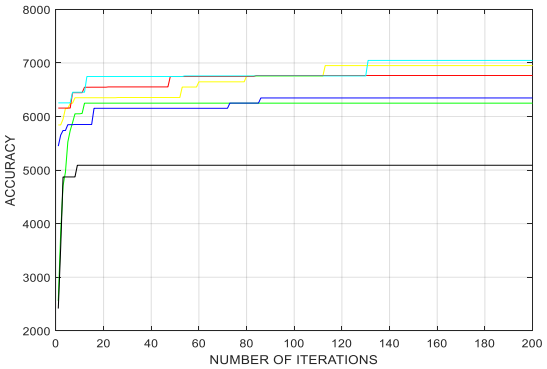

Fig. 8.11. Graph of $\boldsymbol{BF_4}$

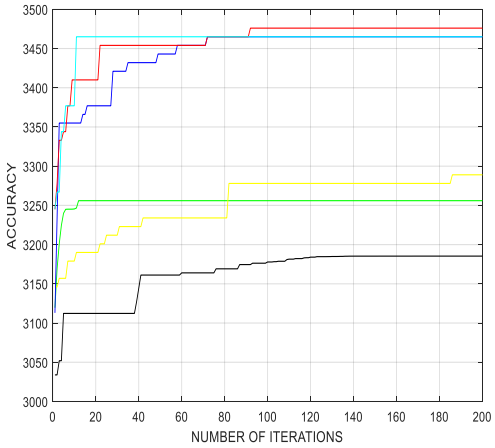

Fig. 8.12. Graph of $\boldsymbol{BF_5}$

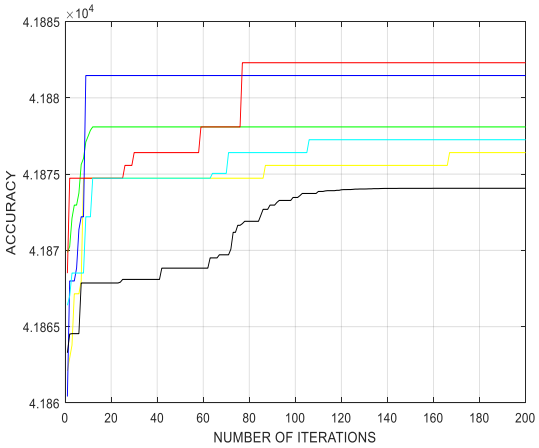

Fig. 8.13. Graph of $\boldsymbol{BF_6}$

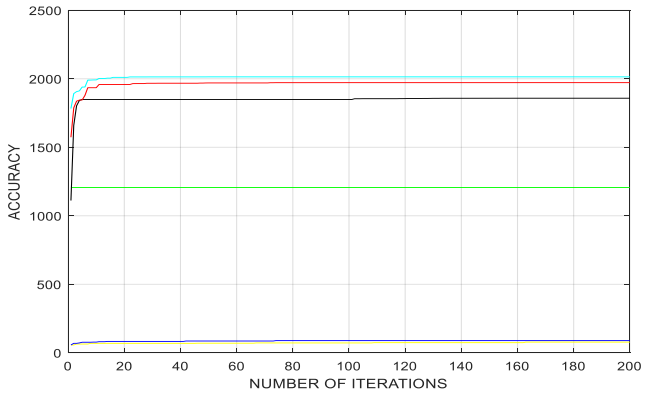

Fig. 8.14 Graph of $\boldsymbol{BF_7}$

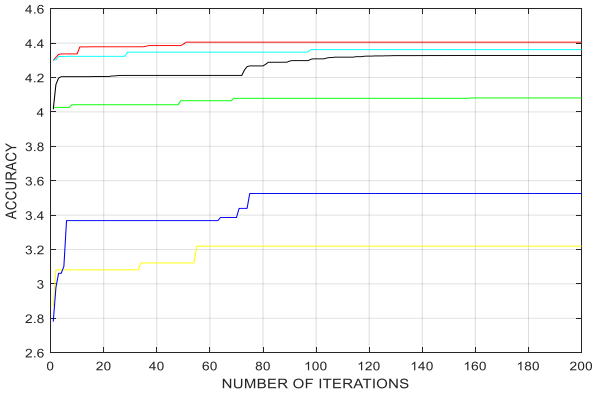

Fig. 8.15. Graph of $\boldsymbol{BF_8}$

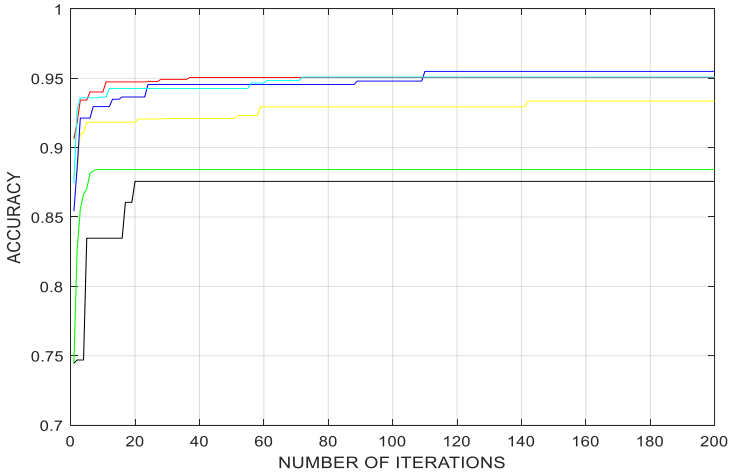

Fig. 8.16. Graph of $\boldsymbol{BF_9}$

Based on the graphs, it can be observed that BTOF2 generally provides better accuracy than BTOF1. However, these results were obtained from a single evaluation of the algorithms. To overcome this limitation, we evaluated the algorithms 20 times and calculated their mean accuracy (fitness values) presented in Table 8.1 for a dimension of 100. Then we constructed an average ranking of all the algorithms on all the BF. The average ranking is given in brackets in Table 8.1.

Table 8.1. The mean fitness value of 20 runs and their ranking when the dimension is 100.

BF	PSO	GA	HHO	SSA	BTOF1	BTOF2
BF_1	67.2 (5)	74.18 (4)	89.68 (3)	56.18 (6)	*91.28 (2)*	**91.3 (1)**
BF_2	72.8 (5)	73.55 (4)	90.2 (3)	66.87 (6)	*91.5 (2)*	**92.05 (1)**
BF_3	**1 (3.5)**	**1 (3.5)**	**1 (3.5)**	**1 (3.5)**	**1 (3.5)**	**1 (3.5)**
BF_4	6269.7 (5)	**6990.45 (1)**	6587.9 (4)	5085.95 (6)	*6970.34 (2)*	6869.16 (3)
BF_5	3257.65 (4)	3289 (5)	3452.35 (3)	3183.36 (6)	*3461.7 (2)*	**3473.25 (1)**

BF_6	41878.77 (3)	41876.67 (5)	**41882.13 (1)**	41874.83 (6)	*41878.78 (2)*	41878.13 (4)
BF_7	1168.55 (4)	74.1 (6)	90.1 (5)	1878.99 (3)	**1981.12 (1)**	*1980.40 (2)*
BF_8	4.17 (4)	3.17 (6)	3.45 (5)	**4.38 (1)**	4.36 (3)	*4.37 (2)*
BF_9	0.9 (5)	*0.93* (4)	**0.95** (3)	0.88 (6)	**0.95 (1)**	**0.95** *(2)*
Average	4.277778	4.277778	3.388889	4.833333	2.055556	2.166667
Rank						
Rank	4	5	3	6	1	2

Note: Bold values indicate the best value, italic indicates second best value, and ranks are given in the corresponding bracket

We can observe from Table 8.1 that BTOF1 gives the best result, followed by BTOF2, HHO, PSO, GA, and SSA, respectively. Then a KWT with a significance level of 0.05 is applied to compare the performance of BTOF1 against PSO, GA, HHO, and SSA. Then H-statistics and p-value are recorded. The difference is significant in all BF except BF3, as the p-value is less than 0.00001. However, since the p-value is 1, the difference is insignificant when BTOF1 is compared against other algorithms on BF_3.

Now considering Table 8.2 and the KWT statistical results, we can conclude the following results –

(1) BTOF1 gives the best solution three times and five times the second best among the nine BF, with a statistical significance difference.

(2) BTOF2 gives the best solution four times and three times the second best among the nine BF, with a statistical significance difference.

(3) PSO gives the best solution once with a statistical significance difference and no second best among the nine BF.

(4) GA gives the best solution two times and no second best among the nine BF, with a statistical significance difference.

(5) HHO gives the best solution two times and no time second best among the nine BF, with a statistically significant difference.

(6) SSA gives the best solution two times and no second best among the nine BF, with a statistical significance difference.We are getting the following observations from the graphs of the BF.

Scalability analysis of BTO

The dimension of optimization problems can affect algorithm efficiency. Therefore, we tested the algorithms on the BF with dimensions 30, 500, and 1000 to conduct a scalability analysis and assess the efficiency of BTOF1 and BTOF2. The mean values of all the algorithms on all the BF with dimensions 30, 500, and 1000 are computed. Regardless of the dimension, BTOF1 consistently outperformed BTOF2 and all other algorithms.

8.6.2 Results and Discussions on FS

We applied all seven algorithms ten times to the FS problem on each dataset and recorded their fitness values. Table 8.2 presents the mean fitness values of all seven algorithms on all the datasets, with the best values highlighted in bold and the second-best values in italics. A ranking system was employed to determine the best-performing algorithm. Additionally, we used KWT to test for statistical significance and recorded the H statistics values and p-values to conclude the statistical significance of the algorithms.

Table 8.2. The mean fitness of all the seven algorithms and their ranking on all seven datasets.

Dataset	PSO	HHO	SSA	BTOF1	BTOF2	BTOF3	BTOF4
Wine	0.95 (7)	0.963077 (5)	0.953846 (6)	*0.982692* (2)	**0.98967** (1)	0.972363 (4)	0.97467 (3)
WDBC	0.958407	0.94619 (6)	0.937186 (7)	0.962832 (4)	0.964602 (3)	**0.967257** (1)	*0.966372* (2)
Ionosphere	0.894912 (6)	**0.905315** (1)	0.881487 (7)	0.896244 (4)	*0.902592* (2)	0.895458 (5)	0.898155 (3)
Sonar	**0.868628** (1)	0.854957 (3)	0.836457 (7)	*0.862671* (2)	0.852695 (4)	0.836957 (6)	0.842652 (5)
Libras	0.840236 (5)	0.840514 (4)	0.835458 (7)	**0.845833** (1.5)	**0.845833** (1.5)	*0.841931* (3)	0.838889 (6)
Hill Valley	*0.557826* (2)	**0.560452** (1)	0.543817 (7)	0.552184 (3)	0.550457	0.548891 (6)	0.551134 (4)
Musk 1	*0.915789* (2)	0.889629 (6)	0.876484 (7)	0.914737 (3)	0.912632	**0.918947** (1)	0.910526 (5)
Average Rank	4	3.714	6.8571	2.7857	2.928	3.71	4
Rank	5.5	4	7	1	2	3	5.5

Note: Bold values indicate the best value, italic indicates second best value, and ranks are given in the corresponding bracket

As we can observe from Table 8.2, BTOF1 gives the best result, followed by BTOF2, BTOF3, HHO, BTOF4, PSO, and SSA, respectively. When KWT is applied between BTOF1, PSO, HHO, and SSA, we get the following observations.

a) In Musk 1 dataset, when BTOF1 is statistically compared with PSO, HHO, and SSA. The H-statistics is 31.8294. The p-value is less than 0.00001. Hence there is a significant difference.
b) In the Hill Valley dataset, when BTOF1 is statistically compared with PSO, HHO, and SSA. The H-statistics is 10.8882. The p-value is 0.02785. Hence there is a significant difference.
c) In the Libras Moment dataset, when BTOF1 is statistically compared with PSO, HHO, and SSA. The H-statistics is 4.6906. The p-value is 0.32054. Hence there is no significant difference.
d) In the Sonar dataset, when BTOF1 is statistically compared with PSO, HHO, and SSA. The H-statistics is 16.0207. The p-value is 0.00299. Hence there is a significant difference.
e) In the Ionosphere dataset, when BTOF1 is statistically compared with PSO, HHO, and SSA. The H-statistics is 21.0473. The p-value is 0.00031. Hence there is a significant difference.
f) In the WBDC dataset, when BTOF1 is statistically compared with PSO, HHO, and SSA. The H-statistics is 36.2838. The p-value is less than 0.00001. Hence there is a significant difference.
g) In the Wine dataset, when BTOF1 is statistically compared with PSO, HHO, and SSA. The H-statistics is 12.6478. The p-value is 0.01313. Hence there is a significant difference.

Hence from the above discussions from (a) to (g) and Table 8.2, we conclude that the proposed algorithm BTOF1 gives the best result with statistical significance when compared with PSO, HHO, and SSA. The KWT is again applied between (BTOF2, PSO, HHO, and SSA) and (BTOF3, PSO, HHO, and SSA). From these results and Table 8.2, we conclude the following results

(1) The variant of BTOF2 gives the second-best solution with statistical significance when compared with PSO, HHO, and SSA.
(2) The variant of BTOF3 gives the third-best solution with statistical significance when compared with PSO, HHO, and SSA.

8.7. Conclusion

This work proposes a novel population-based optimization algorithm called fuzzy BTO to tackle different optimization tasks. BTO is inspired by the cooperative behaviors and chasing styles of predatory fish, Bluefin Trevally, in nature. Several equations are designed to simulate the social intelligence of Bluefin Trevally to solve optimization problems. Here we have taken four variants of Fuzzy BTO in which the parameters are dynamically changed using FLC. This experiment is tested with different dimensions. As a result, we observed that the Signature of fuzzy BTO is significantly better than many other optimization algorithms. Also, convergence proof helps to establish the mathematical strength of fuzzy BTO.

The proposed Fuzzy BTO algorithms were evaluated based on statistical measures and convergence rate, and their performance was tested on nine BF. The complete experimental results indicate that the established BTOF1 outperforms other optimizers regarding searchability and convergent speed when solving global optimization problems. Furthermore, statistical tests were conducted to support this conclusion. The proposed Fuzzy BTO algorithms were also applied to feature selection problems, and the performance of BTOF1 was compared against other algorithms using seven UCI datasets. Again, the experimental results demonstrate that the BTOF1 algorithm outperforms different investigated algorithms. Therefore, the proposed Fuzzy BTO algorithms have the potential to serve as an excellent global optimization algorithm and to tackle FS problems.

In the future, we will explore the application of this algorithm in MOO tasks and expand its application to real-life problems such as machine learning, medical applications, financial fields, and engineering optimization tasks. Furthermore, we will also integrate the novel algorithm with other strategies to build a better optimizer.

Part – IV

Application of the proposed hybrid metaheuristic approach on Forecasting Problem.

(Chapter 9)

Chapter 9

Picture Fuzzy Time Series Forecasting with a novel variant of Particle Swarm Optimization

9.1 Introduction

The landscape of forecasting challenges has prompted a rich tapestry of approaches, with a significant portion gravitating towards FYS or FYS-related approaches. FTS techniques, encompassing Mamdani-type and Sugeno-type FIS, have emerged as pervasive tools in the forecasting realm. The seminal work of Song and Chissom[40] in defining FTS has stimulated a surge in research activities. Kocak's contribution[99] introduced an ARMA-style FTS forecasting technique, while Güler Dincer and Akkuş[100] proposed a robust clustering-based FTS approach with a dedicated focus on fuzzification. Güler Dincer[101] brought forth an FTS approach grounded in fuzzy c-regression, and Bas[102] leveraged the neural network to classify fuzzy relations. Zeng et al.'s [41] methodology for FTS forecasting integrated subtractive clustering and an ABC algorithm. Jiang et al. [103] pioneered an advanced forecasting approach, employing a hybrid model that fused MOO and FTS approach. Tran[104] presented a multivariate FTS approach, and Sadaei[42] explored a synergistic strategy involving convolutional neural networks and FTS. Statistical aspects, including inferences, confidence intervals, and forecast distributions, have become central themes in some FTS investigations. Furthermore, various nature-inspired optimization algorithms, widely applicable in diverse domains, can be tailored to determine optimal interval lengths in FTS forecasting.

One way to think of IFYSs is as a more comprehensive and adaptable variant of FYSs. The modelling and application of intuitionistic FTS were introduced using the intuitionistic fuzzy c-means method. Various Intuitionistic FTS approaches were proposed[105,106]. There were several proposed intuitionistic FTS methods. In hesitant FYSs, The study conducted by Bisht and Kumar involved the application of triangle membership functions, showcasing their adaptability through equal and unequal intervals[107]. It was proposed to use this dual hesitant FYS in an intuitionistic FTS forecasting method. Abhishekh released a FTS technique that was high-order intuitionistic[45].

An improved form of IFYSs called PFYSs offer a more adaptable and inclusive base. Picture fuzzy clustering (PFC) and IFYSs were used by Thong and Son to diagnose medical conditions[108]. Thong and Son, who introduced the notion of PFC, integrated the PFYS set into the clustering approach[109]. Son offered a control theory application as well as a concept for an image FIS[110.]

A thorough review of the literature underscores the efficacy of various forecasting approaches, especially those with more broadly defined set types. As per the insights gathered from existing studies, the augmentation of models with latent variables, essentially additional inputs, has demonstrated a positive impact on inference outcomes. Within this conceptual framework, membership values are latent variables, providing supplementary inputs to the models. This theoretical foundation suggests that utilising PFYSs in forecasting models can be advantageous.

The chapter is structured as follows for the remainder of it. Some fundamental ideas needed for a thorough understanding this chapter are explained in Section 9.2. An innovative metaheuristic variant (EDSPSO) is discussed in Section 9.3. Section 9.4 discusses the proposed PFTS forecasting method. Section 9.5 provides an explanation of the proposed algorithm with the help of TSD from UAE and SBISP. The experimental results are discussed in Section 9.6. The conclusion of the present work is given in Section 9.7.

9.2 Background

To fully grasp the content in this chapter, it is imperative to be familiar with fundamental concepts that has been explained in Section 1.3. Additionally, a basic understanding of Forecasting, explained in Section 1.6, is essential.

9.3 Innovative metaheuristic variant (EDSPSO)

In this section a novel PSO variant is developed using dual swarm strategy and exponential function as mutation operator. Exploration and exploitation stand out as the primary techniques employed in the search for solutions within the solution space. Achieving an optimal answer requires a careful equilibrium between these two approaches to thoroughly navigate the solution domain. Traditional PSO algorithms and their modifications encounter challenges in maintaining this balance, resulting in limitations in generating effective solutions. The enhanced PSO variant (EMPSO) effectively addresses and resolves these challenges. The strategies given below are used to perform efficiently:

1) PSO is subjected to an EMO.
2) Modify the parameters to adapt.
3) A two-swarm technique is implemented, the following Eq. (9.1) is used to update h:-

$$h = h_{min} + (h_{max} - h_{min}) \times \left(\frac{\max\_iter - t}{\max\_iter}\right) \tag{9.1}$$

When particles get stuck in local minima, they experience the mutation operator, which consists of two crucial components. First, in order to improve performance, the algorithm's overall mutation probability, represented by the letter mp_1^t, gradually decreases over time. Secondly, particles whose z^p has stayed stationary in recent iterations see an enhanced chance of mutation, and is accomplished by introducing mp_2^t. After that, Eq. (9.2) is utilized to determine mu_j^t.

$$mu_j^t = mp_1^t \times mp_2^t \tag{9.2}$$

Throughout the iterations, the parameter mp_1^t decreases by a factor of λ_d. It is determined by applying Eq. (9.3) ,which is explained in [78] ,and Eq. (9.4) is used to compute mp_2^t.

$$mp_1^t = mp_1^{t-1} \times \lambda_d \tag{9.3}$$

$$mp_2^t = e^{\left({}^{(NSI_j - NI)}/_{\lambda_m}\right)} \tag{9.4}$$

Here the number of iterations of particle p_j, during which it's z^p remains unchanged, is represented as NSI_j. Parameter NI indicates how many times the particle has been iterated to take benefit of its neighbourhood.

The quest for an optimal exploration-exploitation balance leads to the design of a dual-swarm strategy, wherein the total population N is divided into two halves. The first half of the particles undergoes position and velocity updates through the classical PSO mechanism. In contrast, the second half of the particles exclusively adopts the global best position for position updates, as outlined in Eq. (9.5). Hence, the Eq. (1.17) is modified, and Eq. (9.5) is developed. The comprehensive description of the suggested EDSPSO is given below

$$dual(t) = e^{-\left({}^{(2.5\times t)}/_{\max\_iter}\right)^{2.5}}$$

$$y(t+1) = \begin{cases} z^g(t) + \left(dual(t) \times d_3 \times |z^g(t)|^{dual(t)}\right) & d_4 < 0.5 \\ z^g(t) - \left(dual(t) \times d_3 \times |z^g(t)|^{dual(t)}\right) & otherwise \end{cases} \tag{9.5}$$

Algorithm 1: EDSPSO Pseudo-code.

1. Randomised matrices are assigned to every particle to initialise its position and velocity.
2. The main iteration loop starts from here.
3. Every particle's fitness value is evaluated.
4. The particle's position and velocity are updated using Equations (9.5) and (1.16), respectively.
5. Eq. (9.1) is used to dynamically change the parameter h.
6. Eq. (9.2) is utilised to apply the EMO.
7. The z^p and z^g of every particle are modified.
8. The optimal global solution z^g will have the greatest fitness value.
9. End (***Steps 2 to 8***). The main iteration loop ends here.
10. The iterative technique concludes when the stopping criteria are met, and the final z^g is determined.

9.4 Proposed PFTS forecasting method

This work suggests a novel PFTS forecasting technique. In this case, the suggested EDSPSO-PFTS forecasting technique makes use of PFYS to incorporate non-determinacy during the fuzzification of TSD and EDSPSO to optimize the size of the intervals required to define PFYS. The suggested EDSPSO-PFTS forecasting method is made in three steps. Section 9.4.1, involves building the Universe of Discourse (UOD), generating the FYSs, and converting the FYSs into PFYSs. Section 9.4.1, involves using the max-min operator, which creates a simple FTS forecasting method. In Section 9.4.3, EDSPSO is merged with the PFTS forecasting model to maximize the interval length with MSE as the goal function.

9.4.1 Step 1: Utilizing PFYS for fuzzification

Stage 1.1 The UOD is well-defined as $V = [V_{min}, V_{max}]$. Here $V_{min} = E_{min} - Q_1$ and $V_{max} = E_{max} - Q_2$, where E_{min} and E_{max} are the min and max values of TSD. It is ensured that UOD can support any TSD by properly choosing Q_1 and Q_2.

Stage 1.2 After dividing the UOD into intervals of equal length, construct triangle FYSs on each interval.

Stage 1.3 PFC is used to generate PFYSs in this instance[109]. The cluster centres, degree of positive, neutral, and refusal membership are iterated multiple times until the stopping condition is met. Subsequently, final membership values are noted. Thong and son provide a thorough description of this process[109].

Stage 1.4 Apply the subsequent pseudo-algorithm to incorporate non-determinacy into TSD.

for $j = 1$ to *num* (end of TSD)

for $k = 1$ to *dim* (end of no. of intervals)

$\xi_{I_l} = 1 - \mu_{I_l} - \eta_{I_l} - \gamma_{I_l}$

Choose $\xi_{I_l} = min(\xi_{I_1}, \xi_{I_2}, \ldots \xi_{I_k})\ for\ 1 \leq l \leq dim$

If f_l is a FYS associated to l^{th} PFYS

Allocate l^{th}FYS to the corresponding TSD

End if

End for

End for

9.4.2 Step 2: FTS model Construction

Stage 2.1 First-order fuzzy logical relations (FLRs) $G(t) \rightarrow G(t+1)$ are established via fuzzified TSD. Here, $G(t)$ and $G(t+1)$ are existing and subsequent states, respectively. Once FLRs are generated, merge them to form FLRs (groups), as illustrated. $G_1 \rightarrow G_1, G_1 \rightarrow G_2$ and $G_1 \rightarrow G_3$ can be united as $G_1 \rightarrow G_1, G_2, G_3$ and obtain first-order fuzzy time relation $S = \cup S_i$.

Stage 2.2 The fuzzy forecast is computed using the equation $G_i = G_{i-1} \cdot S$. Here, the fuzzified values G_{i-1}, G_i and $\cdot$ correspond to the present state, the subsequent state, and the max-min operator, respectively.

Stage 2.3 The predicted outcome is defuzzified using the formula below to determine its numerical value

$$\textit{forecasted output} = \frac{\sum_{j=1}^{num}(g_j \times mid_j)}{\sum_{j=1}^{num}(g_j)} \tag{9.6}$$

Here, g_j and mid_j indicate the interval's intermediate point and fuzzy outcome, respectively.

9.4.3 Step 3: Integration of EDSPSO with PFTS model

This step involves combining a PFTS model with EDSPSO to determine the optimal interval duration. Initializations are made for all necessary EDSPSO parameters as well as the maximum number of iterations. TSD is divided into intervals, which are reflected in each particle in the suggested EDSPSO-PFTS forecasting technique. Each particle is a collection of $s-1$ components. where $q_{j-1} < q_j$ for $1 \leq j \leq s-1$ $\left(ie.\ q_1 < q_1 \cdots < q_j < \cdots < q_{s-2} < q_{s-1}\right)$ These $s-1$ components create s interval as $v_1 = [q_0, q_1]$, $v_2 = [q_1, q_2], \cdots, v_j = \left[q_{j-1}, q_j\right], \cdots, v_{s-1} = [q_{s-2}, q_{s-1}], v_s = [q_{s-1}, q_s]$. When a particle improves its position, components of the connected new set are instantly modified so that each member $q_j (1 \leq j \leq s-1)$ is arranged in an ascending sequence. A sample particle is shown below.

q_1	q_2	$\cdots$	q_j	$\cdots$	q_{s-1}

$$MSE = \frac{\sum_{j=1}^{num}(OR_j - FO_j)^2}{num} \tag{9.7}$$

MSE is used in every iteration to evaluate each particle's effectiveness, and results are provided in Eq. (9.7). Here num, FO_j, and OR_j stand for the number of forecasted data points, the predicted j^{th}, and actual j^{th} TSD. Each particle adjusts its position using Eqns. (1.16) and (9.5), and the process is carried out until the predefined stopping criterion is satisfied to evaluate each particle's forecasting outcomes. The outcomes are compared among all previous individual best positions for each particle, provided that the stopping requirements are satisfied. This step is fully explained in Algorithm 1. The suggested EDSPSO-PFTS forecasting method is described in Algorithm 2, and its flowchart is displayed in Fig 9.1.

***Algorithm 2*:** Proposed EDSPSO-PFTS Algorithm

***INPUT*:** A) The parameters are specified and every particle's position has been initialized.

B) UOD is established.

***OUTPUT*:** Forecasted output values

1) for $k = 1$ to max_iter (No.of iterations)
2) *for* $j = 1$ to *NP* (No. of particles)
3) Phase in steps 1 and 2 are applied first.
   (In this step PFYSs are constructed using the concept of PFC).
4) Now, Phase 3: Incorporation of EDSPSO with PFTS model starts from this step.
5) The MSE value is computed using Eq. (9.7).
6) The position and velocity are computed using Eq. (9.5) and Eq. (1.16) respectively.
7) Then EMO is applied using Eq. (9.1).
8) The personal best of each particle and the global best particle are updated.
9) *End for*
10) *End for*

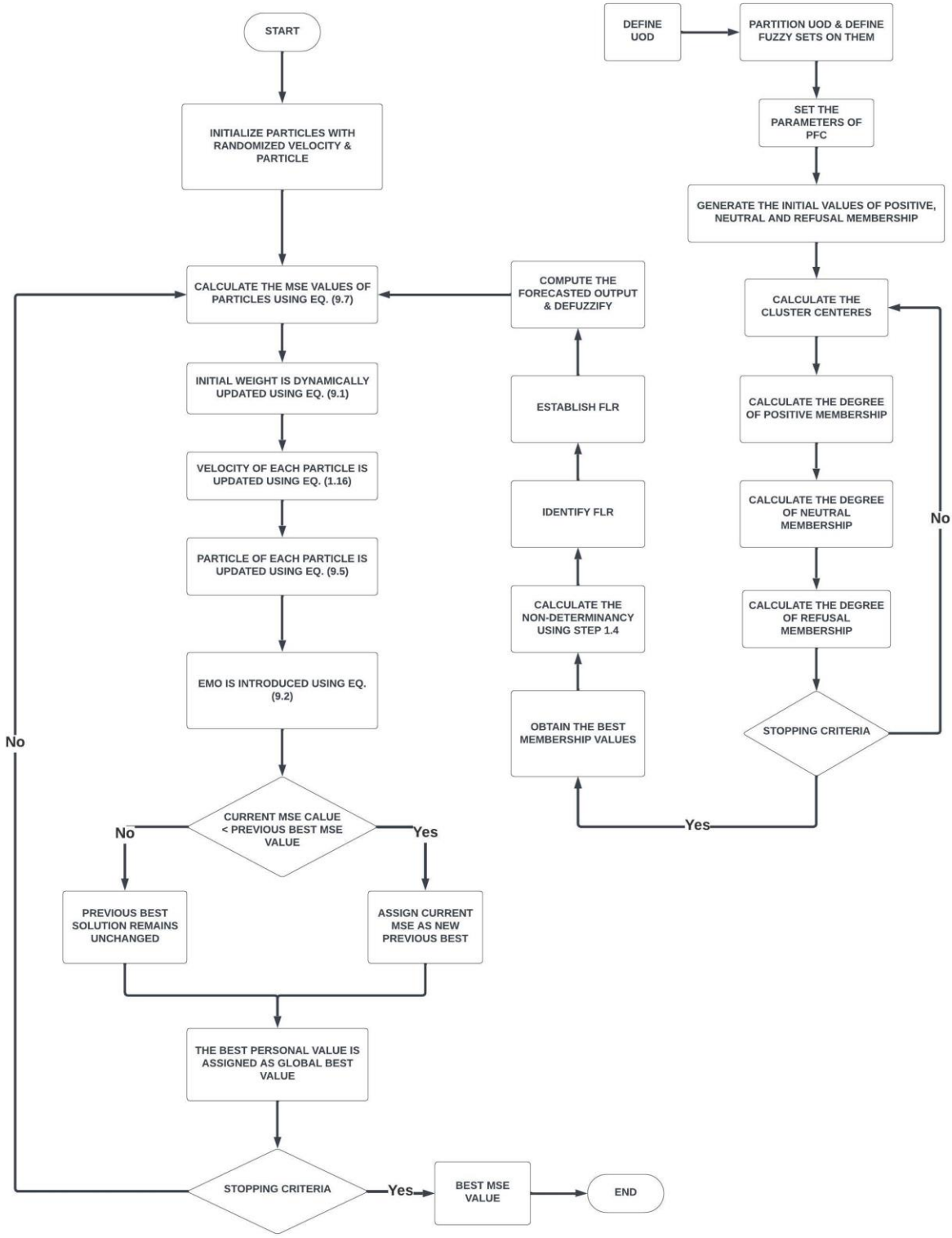

Fig. 9.1. Flowchart of EDSPSO-PFTS

9.4.4 Statistical analysis

The effectiveness of the forecasting approach is also assessed using a variety of statistical criteria[111], such as the correlation coefficient (R), tracking signal (TS), coefficient of determination (R^2), and performance parameters [112]. These criteria are in addition to the error measures (MSE and AFE). The RMSE, AFE, and other statistical metrics formulas that were used to evaluate the efficacy and statistical validity of the suggested EMPSO-PFTS forecasting approach are shown in Table 9.1. Table 9.1 shows the expected TSD, actual TSD, number of TSD, and standard deviation of the TSD, respectively, as well as $forecasted_j$, $Original_j$, *num*, and σ are forecasted TSD, actual TSD, number of TSD and the standard deviation of the TSD respectively. The positive and negative values of M_{ad} represent the forecasting model's tendency towards under- and over-forecasting. Biased under-forecasting is indicated by $TS > 4$, while biassed over-forecasting is shown by $TS < 4$.

Table 9.1 statistical parameters and error measures.

Sl.No	Term	Mathematical expression	Acceptable Range
1	MSE	$\frac{\sum_{j=1}^{num}(OR_j - FO_j)^2}{num}$	
2	AFE (in %)	$\frac{\sum_{j=1}^{num}\left(\frac{\lvert FO_j - OR_j\rvert}{OR_j} * 100\right)}{num}$	
3	R	$\frac{num\sum OR_j * FO_j - (\sum OR_j)(\sum FO_j)}{\left(\sqrt{num(\sum {OR_j}^2) - (\sum OR_j)^2}\right) - \left(\sqrt{num(\sum {FO_j}^2) - (\sum FO_j)^2}\right)}$	$-1 < R < 1$
5	PP	$1 - \frac{(\sqrt{MSE})}{\sigma}$	$PP > 0$
6	M_{ad}	$\frac{\sum_{j=1}^{num}\lvert FO_j - OR_j\rvert}{num}$	
7	TS	$\frac{\sum_{j=1}^{num}(FO_j - OR_j)}{M_{ad}}$	$-4 < TS < 4$

9.5 Application of EDSPSO-PFTS

This section describes how the suggested EDSPSO-PFTS method is implemented and performs better than others. This technique is applied by using historical TSD from two sources: UAE in Section 9.5.1 and SBISP in Section 9.5.2.

9.5.1 Forecasting UAE

The following steps describe how the proposed EDSPSO-PFTS method is used to forecast the UAE.

Step 1*:* Here, the FYSs are constructed, and the PFYS are constructed for modelling UAE.

Stage *1.1*: $V = [13000, 20000]$ has been defined as UOD by using $Q_1 = 55$ and $Q_2 = 663$. $E_{min} = 13055$ and $E_{max} = 19337$ are noted from TSD of UAE. This UAE is specified in Table 9.2.

Stage *1.2:* $V = [13000, 20000]$ has been separated into fourteen intervals $v(j) = [13000 + (j-1)h, 13000 + jh], (\, j = 1\ to 14, h = 500)$. The purpose of FYSs G_j (j=1 to 14) are to fuzzify the UAE's TSD. The grade of membership is given in Table 9.3.

$$G_j = [13000 + (j-1)h, 13000 + jh, 13000 + 2jh], j = 1\ to 13, h = 500$$

$$G_j = [13000 + (j-1)h, 13000 + jh, 13000 + jh], j = 14, h = 500$$

Stage *1.3:* PFYSs are established from FYSs using the concept of PFC.

Stage *1.4:* The computation non-determinacy of the PFYSs of the UAE is described below-

- The UAE of the year 1971 is related with the FYSs G_1, G_4 and G_5 FYS. Hence PFYSs $J_1 = \langle 0.333987, 0.170279, 0.000034 \rangle$, $J_4 = \langle 0.56003, 0.223867, 0 \rangle$ and $J_5 = \langle 0.560229, 0.22392, 0 \rangle$ are associated with this enrolment. The amount of non-determinacy of the above three PFYSs are 0.495699, 0.216103, 0.215851 respectively. Since the amount of non-determinacy 0.215851 is minimum in $J_5$and J_5 is corresponding to a FYS G_5, therefore fuzzified enrollment of the year 1971 is taken as G_5.

- Similarly, the UAE of the years (1972- 1992) that are related with the FYSs are computed. The amount of the non-determinacy values of the PFYSs are computed and are given in Table 9.4. After calculating each enrolment's minimum non-determinacy, the fuzzified enrolments are displayed in Table 9.5.

Step 2: Fuzzified TSDs are utilised to generate FLRs. $G(t) \rightarrow G(t+1)$. The detailed process is given in Section 9.4.1. Fuzzy forecasted UAE are defuzzified by utilizing Eq. (9.6).

Step 3: EDSPSO integration with the PFTS gives the optimal interval length for FYSs and, by extension, PFYSs, by minimizing the MSE in UAE forecasting. Here we have taken *N* as 7. Tables 9.6 and 9.7 show the initial velocity and random positions. Each particle can be seen as a distinct collection of

$v_1 = [q_0, q_1], v_2 = [q_1, q_2], v_3 = [q_2, q_3], v_4 = [q_3, q_4], v_5 = [q_4, q_5], v_6 = [q_5, q_6],$
$v_7 = [q_6, q_7], v_8 = [q_7, q_8], v_9 = [q_8, q_9], v_{10} = [q_9, q_{10}], v_{11} = [q_{10}, q_{11}],$
$v_{12} = [q_{11}, q_{12}], v_{13} = [q_{12}, q_{13}], v_{14} = [q_{13}, q_{14}].$

For instance, particle 1 categorizes a single set of 14 intervals based on its initial location.

$v_1 = [13000,13500], v_2 = [13500,14000], v_3 = [14000,14500], v_4 = [14500,15000],$
$v_5 = [15000,15500], v_6 = [15500,16000], v_7 = [16000,16500], v_8 = [16500,17000],$
$v_9 = [17000,17500], v_{10} = [17500,18000], v_{11} = [18000,18500], v_{12} = [18500,19000],$
$v_{13} = [19000,19500]$ and $v_{14} = [19500,20000].$

By applying the PFYSs at the intervals given by particle 1, the MSE value of 312915.4 is produced, and this value is computed using Eq. (9.7). To forecast UAE, the same process is repeated by all the particles. The comprehensive ESDPSO procedures are provided in Algorithm 1. In this case, z^g is computed using z^p and the lowest MSE value. This process continues until the termination condition is met.

Table 9.2 UAE from 1971 to 1992

Year	UAE	Year	UAE
1971	13055	1982	15433
1972	13563	1983	15497
1973	13867	1984	15145
1974	14696	1985	15163
1975	15460	1986	15984
1976	15311	1987	16859
1977	15603	1988	18150
1978	15861	1989	18970

1979	16807	1990	19328
1980	16919	1991	19337
1981	16388	1992	18876

Table 9.3 Membership grades of UAE in different FYSs

UAE	G_1	G_2	G_3	G_4	G_5	G_6	G_7	G_8	G_9	G_{10}	G_{11}	G_{12}	G_{13}	G_{14}
13055	0.11	0	0	0	0	0	0	0	0	0	0	0	0	0
13563	0.874	0.126	0	0	0	0	0	0	0	0	0	0	0	0
13867	0	0.734	0	0	0	0	0	0	0	0	0	0	0	0
14696	0	0	0.608	0.392	0	0	0	0	0	0	0	0	0	0
15460	0	0	0	0.08	0.92	0	0	0	0	0	0	0	0	0
15311	0	0	0	0.378	0.662	0.206	0	0	0	0	0	0	0	0
15603	0	0	0	0	0.794	0.722	0	0	0	0	0	0	0	0
15861	0	0	0	0	0.278	0	0	0	0	0	0	0	0	0
16807	0	0	0	0	0	0	0.386	0.614	0	0	0	0	0	0
16919	0	0	0	0	0	0.224	0.162	0.838	0	0	0	0	0	0
16388	0	0	0	0	0	0	0.776	0	0	0	0	0	0	0
15433	0	0	0	0.134	0.866	0	0	0	0	0	0	0	0	0
15497	0	0	0	0.006	0.994	0	0	0	0	0	0	0	0	0
15145	0	0	0	0.71	0.29	0	0	0	0	0	0	0	0	0
15163	0	0	0	0.674	0.326	0.968	0	0	0	0	0	0	0	0
15984	0	0	0	0	0.032	0	0	0	0	0	0	0	0	0
16859	0	0	0	0	0	0	0.282	0.718	0	0	0	0	0	0
18150	0	0	0	0	0	0	0	0	0	0.7	0.3	0	0	0
18970	0	0	0	0	0	0	0	0	0	0	0.06	0.94	0	0
19328	0	0	0	0	0	0	0	0	0	0	0	0.344	0.656	0
19337	0	0	0	0	0	0	0	0	0	0	0	0.326	0.674	0
18876	0	0	0	0	0	0	0	0	0	0	0.248	0.752	0	0

Table 9.4 Non-determinacy values of PFYS for UAE

UAE	G_1	G_2	G_3	G_4	G_5	G_6	G_7	G_8	G_9	G_{10}	G_{11}	G_{12}	G_{13}	G_{14}
13055	0.496	1	1	0.217	0.216	1	1	1	1	1	1	1	1	1
13563	1	0.283	0.878	0.217	0.216	1	1	1	1	0.905	1	1	1	1
13867	1	0.597	1	0.217	0.216	1	1	1	1	1	1	1	1	1
14696	0.216	1	1	1	0.217	1	1	1	1	1	1	1	1	1
15460	0.216	1	0.861	0.252	1	1	1	1	1	0.874	1	1	1	1
15311	0.216	1	1	1	1	1	1	1	1	0.884	1	1	1	1
15603	0.217	0.911	1	0.216	1	1	1	1	1	0.839	1	1	1	1
15861	0.217	1	0.839	0.216	1	0.262	1	1	1	0.818	1	1	1	1
16807	0.217	1	1	0.217	0.216	1	0.27	1	1	0.823	1	1	1	1

16919	0.217	1	0.802	0.217	0.216	1	0.455	1	1	1	1	1	1	1
16388	0.217	0.837	1	0.217	0.216	1	1	1	1	1	1	1	1	1
15433	0.215	0.905	0.885	0.354	1	1	1	1	1	0.874	1	1	1	1
15497	0.215	1	0.848	0.216	1	1	1	1	1	1	1	1	1	1
15145	0.215	0.855	0.819	1	1	1	1	1	1	1	1	1	1	1
15163	0.215	0.854	1	1	1	1	1	1	1	1	1	1	1	1
15984	0.217	1	1	0.216	0.22	0.236	1	1	1	0.888	1	1	1	1
16859	0.217	0.877	0.813	0.217	0.216	1	0.217	1	1	1	1	1	1	1
18150	0.216	1	1	0.217	0.215	1	1	1	1	1	1	1	1	1
18970	0.216	1	1	0.217	0.215	1	1	1	1	1	0.2182	0.232	1	1
19328	0.216	0.824	1	0.217	0.217	1	1	1	1	0.215	1	1	0.879	1
19337	0.216	0.857	1	0.217	0.217	1	1	1	1	0.215	1	1	1	1
18876	0.216	1	0.828	0.217	0.217	1	1	1	1	1	1	0.215	1	1

Table 9.5. Fuzzified UAE using PFTS

Year	***E***	**Fuzzified *UAE***	**Year**	***E***	**Fuzzified *UAE***
1971	13055	G_5	1982	15433	G_1
1972	13563	G_5	1983	15497	G_1
1973	13867	G_5	1984	15145	G_1
1974	14696	G_1	1985	15163	G_1
1975	15460	G_1	1986	15984	G_4
1976	15311	G_1	1987	16859	G_5
1977	15603	G_4	1988	18150	G_5
1978	15861	G_4	1989	18970	G_5
1979	16807	G_5	1990	19328	G_{10}
1980	16919	G_5	1991	19337	G_{10}
1981	16388	G_5	1992	18876	G_{12}

Table 9.6 Randomized initial positions of particle

q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8	q_9	q_{10}	q_{11}	q_{12}	q_{13}	MSE
13500	14000	14500	15000	15500	16000	16500	17000	17500	18000	18500	19000	19500	312915.4
13398.44	13767.05	14500	14861.72	15500	16000	16405	16816	17244	18000	18500	18898	19500	380051.268
13500	13741	14500	15000	15500	16000	16500	17000	17459	17827	18227	18693	19500	381760.59
13191	14000	14186	15000	15362	16000	16391	17000	17500	17944.22	18500	18632	19500	403090.65
13301	13710.73	14500	15000	15500	16000	16495	17000	17387	17800	18500	18683	19244	343400.66
13500	14000	14500	14600	15500	15837	16250	16798.21	17500	18000	18246	19000	19500	489981.92
13500	13785.51	14500	14955	15275	16000	16500	17000	17227	18000	18500	19000	19500	371665.197

Table 9.7 Randomized initial velocity of particles

v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	v_{11}	v_{12}	v_{13}
78.44	-100	-89.7	92.52	100	95.71	100	-100	-100	100	100	-100	100
7.43	66.05	-3.33	93.71	-57.84	-100	100	-100	-100	-88.34	100	-100	100
-52.22	100	100	-100	100	-50.13	100	-37.96	-100	100	100	-100	-40.35
100	-100	100	93.07	100	14.76	100	-100	-100	-93.78	100	-100	100
100	16.73	100	-100	91.04	-100	100	-100	-100	100	100	-100	100
57.83	-100	72.81	100	100	100	100	-56.79	-100	-100	100	-100	42.04
-100	-19.48	100	100	100	-100	100	-100	-100	-100	59.2	-100	1.95

9.5.2 Forecasting SBISP

Investors must be able to forecast volatility in stock prices since it is an indication of high risk and affects their capacity to make well-informed decisions that maximize returns. We use the proposed EDSPSO-PFTS technique for the TSD of the SBISP. The SBISP dataset used in this study includes records from April 2008 to March 2010 (Table 9.8). The calculation to forecast the SBISP implementing the suggested EDSPSO-PFTS method is shown below.

Step 1:

Stage *1.1:* By considering the max and min values from Table 9.8, the UOD is determined as $V = [741, 2892]$ from the TSD of SBI share.

Stage *1.2:* $V = [741, 2892]$ has been divided into fourteen intervals $v(j) = [741 + (j - 1)h, 741 + jh], (\ j = 1\ to 14, h = 153.64)$. FYSs G_j (j=1 to14) are created to fuzzify the TSD of the SBISP and its membership grade is given in Table 9.9.

$$G_j = [741 + (j-1)h, 741 + jh, 741 + 2jh], j = 1\ to 13, h = 153.64$$

$$G_j = [741 + (j-1)h, 741 + jh, 741 + jh], j = 14, h = 153.64$$

Stage *1.3:* Using the concept of PFC (Algorithm 1), membership values are computed and fourteen PFYSs are constructed from FYSs G_j.

Stage *1.4:* Non-determinacy is included in the fuzzification of TSD using the ideas given in phase 1.4 of Section 9.4.1. We can observe that the share price of the year 9-Mar (1132.25) is associated with the FYSs G_3, G_4, and G_6 FYSs. Hence, $J_3 = \langle 0.5414,0.218942,0\rangle, J_4 = \langle 0.560387,0.22396,0\rangle$, $J_6 = \langle 0.560404,0.223965,0\rangle$, are the three PFYSs that are associated with this enrolment. The amount of non-determinacy of the above three PFYSs are 0.239658, 0.215652, and 0.21563 respectively. Since the amount of non-0.21563 is minimum in J_6 and J_6 is corresponding to FYS G_6, therefore fuzzified enrollment of the year 8-April is taken as G_6.

Step 2: In Step 2, the FTS model is created. On FLRs and FLR (group), max-min composition operations are then applied. Using Eq. (9.6), fuzzy forecasted enrolments are defuzzified.

Step 3: For constructing $V = [741, 2892]$, q_0, and q_s are assumed to be 741 and 2892, respectively. The number of particles is assumed to be 7, and the number of intervals is assumed to be 14, correspondingly. As an illustration, particle 1's initial position categorizes a specific set of fourteen intervals-

$v_1 = [741,894.64], v_2 = [894.64,1048.28], v_3 = [1048.28,1201.92],$
$v_4 = [1201.92,1355.57], v_5 = [1355.57,1509.21], v_6 = [1509.21,1662.857],$
$v_7 = [1662.857,1816.5], v_8 = [1816.5,1970.14], v_9 = [1970.14,2123.786],$
$v_{10} = [2123.786,2277.42], v_{11} = [2277.42,2431.07], v_{12} = [2431.07,2584.714],$
$v_{13} = [2584.714,2738.357]$ and $v_{14} = [2738.357,2892]$.

.

By applying the PFYSs at the intervals specified by particle 1, the MSE value of 14848.5517 is generated, and this value is determined using Eq. (9.7). Every particle goes

through the same procedure to forecast UAE. In Algorithm 1, the detailed ESDPSO processes are given. Here, the lowest MSE value and z^p are used to calculate z^g. Up until the termination condition is satisfied, this process keeps going.

Table 9.8 Actual SBISP

Months	SBISP	Months	SBISP
April-08	1819.95	April-09	1355
May-08	1840	May-09	1891
June-08	1496.7	June-09	1935
July-08	1567.5	July-09	1840
August-08	1638.9	August-09	1886.9
September-08	1618	September-09	2235
October-08	1569.9	October-09	2500
November-08	1375	November-09	2394
December-08	1325	December-09	2374.75
January-09	1376.4	January-10	2315.25
February-09	1205.9	February-10	2059.95
March-09	1132.25	March-10	2120.05

Table 9.9 Membership grades of the SBISP of different FYSs

SBISP	G_1	G_2	G_3	G_4	G_5	G_6	G_7	G_8	G_9	G_{10}	G_{11}	G_{12}	G_{13}	G_{14}
1819.95	0	0	0	0	0	0.712	0.288	0	0	0	0	0	0	0
1840	0	0	0	0	0	0.659	0.341	0	0	0	0	0	0	0
1496.7	0	0	0	0.419	0.582	0	0	0	0	0	0	0	0	0
1567.5	0	0	0	0.156	0.844	0	0	0	0	0	0	0	0	0
1638.9	0	0	0	0	0.711	0.289	0	0	0	0	0	0	0	0
1618	0	0	0	0	0.917	0.083	0	0	0	0	0	0	0	0
1569.9	0	0	0	0.147	0.853	0	0	0	0	0	0	0	0	0
1375	0	0	0	0.869	0.131	0	0	0	0	0	0	0	0	0
1325	0	0	0.407	0.593	0	0	0	0	0	0	0	0	0	0
1376.4	0	0	0	0.864	0.136	0	0	0	0	0	0	0	0	0
1205.9	0	0.351	0.649	0	0	0	0	0	0	0	0	0	0	0
1132.25	0	0.616	0.384	0	0	0	0	0	0	0	0	0	0	0
1355	0	0	0	0.943	0.057	0	0	0	0	0	0	0	0	0
1891	0	0	0	0	0	0.524	0.476	0	0	0	0	0	0	0
1935	0	0	0	0	0	0.408	0.592	0	0	0	0	0	0	0
1840	0	0	0	0	0	0.659	0.341	0	0	0	0	0	0	0
1886.9	0	0	0	0	0	0.535	0.465	0	0	0	0	0	0	0
2235	0	0	0	0	0	0	0	0	0.551	0.449	0	0	0	0
2500	0	0	0	0	0	0	0	0	0	0	0	0.791	0.209	0
2394	0	0	0	0	0	0	0	0	0	0	0.809	0.191	0	0
2374.75	0	0	0	0	0	0	0	0	0	0	0.993	0.007	0	0
2315.25	0	0	0	0	0	0	0	0	0.124	0.876	0	0	0	0
2059.95	0	0	0	0	0	0.077	0.923	0	0	0	0	0	0	0
2120.05	0	0	0	0	0	0	0	0.562	0.438	0	0	0	0	0

9.6 Results and Discussions

The UAE and SBISP are forecasted by performing the EDSPSO-PFTS approach 20 times. The best result from each run is used to decide the final output. Numerous parameters, including the N, max_iter, h, h_1 and h_2 shape the dynamics of EDSPSO. The following are the EDSPSO parameters, which are taken from work[56].

Fig 9.2 shows that the MSE value of the z^g drops as iterations increase. An evaluation of the suggested EDSPSO-PFTS method's capability for UAE is conducted by comparing it with techniques[40,106,107,113–120]. The suggested EDSPSO-PFTS method's performance is contrasted with Chen and Chung GA-based FTS approaches[121]. When analyzed alongside other algorithms, the suggested EDSPSO-PFTS has a better MSE impact (Tables 9.10 and 9.11).

Error metrics and a variety of statistical factors are used for comparative studies to evaluate the effectiveness of EDSPSO-PFTS. In comparison to existing approaches [40,106,107,114–119] the suggested technique EDSPSO-PFTS is evaluated using statistical parameters listed in Table 9.1. The comparison results are shown in Table 9.12. The statistical parameters are found to be within reasonable bounds by the evaluation.

When analysed against several different FTS forecasting models, the effect of the suggested EDSPSO-PFTS approach for the TSD of SBISP shows better outcomes[25,106,107,114,115,119,120,122]. When analysed alongside other techniques, the suggested EDSPSO-PFTS method performs superior (Table 9.13).

A comparison is shown between the proposed EDSPSO-PFTS method and other methods[106,107,114,115,119,122] using statistical parameters listed in Table 9.14. The comparison results are shown in Table 9.15. The statistical parameters are found to be within reasonable ranges by the analysis.

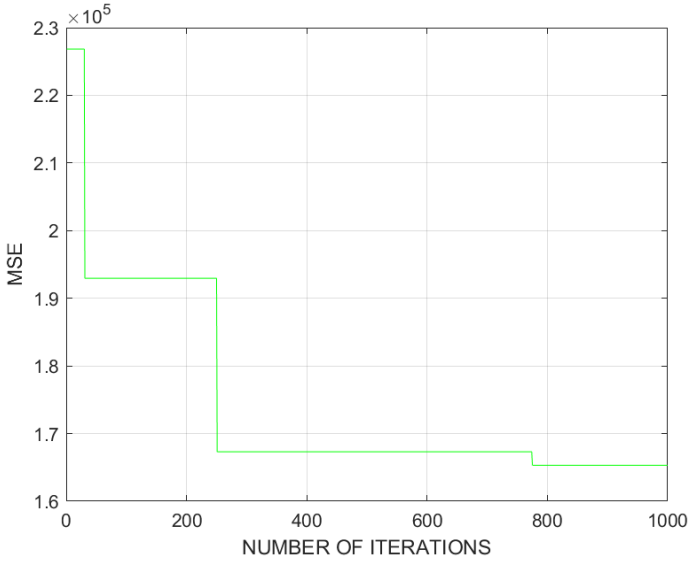

Fig. 9.2. Curve of MSE values

Table 9.10 Forecasted UAE

UAE	Song and Chissom[40]	Song and Chissom[123]	Chen[114]	Huarng[115]	Lee and Chou[116]	Chen and Chung[121]	Yolcu et al.[117]	Qiu et al.[118]
MSE	423020.16	775685.33	407503.49	227500.38	251281.638	209003	648290	261458
RANK	13	15	12	8	11	7		12
13055	-	-	-	-	-	-	-	-
13563	140 0 0	-	140 0 0	-	14025	-	14031.35	14195
13867	14000	-	14000	-	14568	14146	14795.36	14424
14696	14000	-	14000	14000	14568	14878	14795.36	14593
15460	15500	14700	15500	15500	15654	14878	14795.36	15589
15311	16000	14800	16000	15500	15654	15609	16406.57	15645
15603	16000	15400	16000	16000	15654	15609	16406.57	15634
15861	16000	15500	16000	16000	15654	16214	16406.57	16100
16807	16000	15500	16000	16000	16917	16214	16406.57	16188
16919	16813	16800	16833	17500	17823	16818	17315.29	17077
16388	16813	16200	16833	16000	17283	16818	17315.29	17105
15433	16789	16400	16833	16000	16197	15609	17315.29	16369
15497	16000	16800	16000	16000	15654	15609	16406.57	15643
15145	16000	16400	16000	15500	15654	14146	16406.57	15648
15163	16000	15500	16000	16000	15654	14146	16406.57	15622
15984	16000	15500	16000	16000	15654	16818	16406.57	15623
16859	16000	15500	16000	16000	16197	16818	16406.57	16231
18150	16813	16800	16833	17500	17283	17992	17315.29	17090
18970	19000	19300	19000	19000	18369	19126	19132.79	18325
19328	19000	17800	19000	19000	19454	19126	19132.79	19000
19337	19000	19300	19000	19500	19454	19126	19132.79	19000
18876	-	19600	19000	19000	-	19126	19132.79	19000

Table 9.11 Forecasted UAE

UAE	Joshi and Kumar[119]	Kumar and Gangwar[106]	Bisht and Kumar[107]	Gupta and Kumar[120]	Pattnayak et al.[113]	Pant and Kumar[25]	Proposed Algorithm (EDSPSO-PFTS)
MSE	1881474	243601	183723	186313	1800964	178665	172024.06
RANK	6	10	4	5	3	2	1
13055	-	-	-	-	-	-	-
13563	14250	13693	13595.67	13680.75	13637	13682	14276.352
13867	14246	13693	13817.75	13844.43	14120	13682	15025.843
14696	14246	14867	14929.79	14951.36	14408	14722	15025.843
15460	15491	15287	15541.27	15532.34	15195	15427	15068.448
15311	15491	15376	15540.62	15533.19	15712	15544	15068.448
15603	15491	15376	15540.62	15533.19	15635	15544	15068.448
15861	16345	15376	15540.62	15533.19	15786	15544	15068.448
16807	16345	16523	16254.5	16298.77	15918	16665	15068.448
16919	15850	16606	17040.41	17113.79	16406	15994	15339.114
16388	15850	17519	17040.41	17113.79	16406	17230	15776.959
15433	15850	16606	16254.5	16298.77	16190	15994	15776.959
15497	15450	15376	15540.62	15533.19	15698	15544	15776.959
15145	15450	15376	15540.62	15533.19	15731	15544	16655.35
15163	15491	15287	15541.27	15532.34	15550	15516	16655. 35
15984	15491	15287	15541.27	15532.34	15559	15516	16655. 35
16859	16345	16523	16254.5	16298.77	15982	16665	16655. 35
18150	17950	17519	17040.41	17113.79	16433	17230	18506.346
18970	18961	19500	18902.3	18741.35	17366	18820	18506.346
19328	18961	19000	19357.3	19190.44	17967	19311	18506.346
19337	18961	19500	19168.56	18972.15	18230	19311	19196.290
18876	18961	19500	19168.56	18972.15	18236	19311	19196.290

Table 9.12 Statistical performance analysis of EDSPSO-PFTS on UAE

Model	MSE	AFE	R	R^2	PP	M_{ad}	TS
Song and Chissom[40]	423020.16	3.22	0.9173	0.8418	0.6419	516.35	2.6861
Song and Chissom[123]	775685.33	3.75	0.8317	0.6917	0.5151	729.05	-4514
Chen[114]	407503.489	3.11	0.9262	0.8579	0.6485	498.80	3.2377
Huarng[115]	227500.38	2.36	0.9467	0.8962	0.7374	383.45	0.5554
Lee and Chou[116]	251281.638	2.67	0.9542	0.9105	0.7240	428.95	4.1240
Yolcu et al.[117]	648298.728	4.29	0.9121	0.83	0.5567	643.41	13.44
Qiu at al.[118]	261458.368	2.65	0.9599	0.9219	0.7185	430.76	2.0521
Joshi and Kumar[119]	188147.737	2.24	0.9688	0.9387	0.7612	358.71	-4.853
Kumar and Gangwar[106]	243601.473	2.33	0.9594	0.9254	0.7235	368.68	1.554
Bisht and Kumar[107]	183723.676	1.94	0.9667	0.9346	0.7640	318.69	-0.214
Proposed Method (EDSPSO-PFTS)	165322.2209	1.8462	0.9708	0.9425	0.7762	288.3228	0.5145

Table 9.13 Forecasted SBISP.

SBISP	Chen[114]	Huarng[115]	Pathak and Singh[122]	Joshi and Kumar[119]	Kumar and Gangwar[106]	Bisht and Kumar[107]	Gupta and Kumar[120]	Pant and Kumar[25]	Proposed Algorithm (EDSPSO-PFTS)
MSE	35066	26909	42419	40068	17234.438	32051	34762	31483	9521.94
RANK	7	3	9	8	2	5	6	4	1
1819.95	-	-	-	-	-	-	-	-	
1840	1900	1855	1770	1777.8	1725.98	1877.657	1860.08	1716	1878.92
1496.7	1900	1855	1832.5	1865.71	1725.98	1877.657	1860.08	1776	1409.265
1567.5	1500	1575	1470	1531.5	1512.39	1466.36	1452.59	1491	1409.265
1638.9	1500	1505	1570	1531.5	1512.39	1466.36	1452.59	1491	1636.838
1618	1600	1610	1670	1777.8	1574.35	1533.504	1544.29	1491	1620.118
1569.9	1600	1610	1603.33	1531.5	1574.35	1533.504	1544.29	1491	1409.265
1375	1500	1505	1670	1531.5	1512.39	1466.36	1452.59	1491	1335.415
1325	1433	1482	1382.5	1504.23	1305.52	1520.652	1682.31	1542	1335.415
1376.4	1433	1365	1332.5	1504.23	1665.9	1520.652	1682.31	1542	1335.415
1205.9	1433	1482	1332.5	1504.23	1305.52	1520.652	1682.31	1542	1270.086
1132.25	1433	1155	1195	1258.23	1294.27	1144.718	1264.98	1270	1409.265
1355	1300	1365	1145	1258.23	1294.27	1322.446	1264.98	1270	1335.415
1891	1433	1482	1357.5	1504.23	1665.9	1520.652	1682.31	1542	1859.46
1935	1900	1890	1882.5	1865.71	2006.51	1877.657	2138.21	2041	2004.963
1840	1900	1890	1970	1883.93	2006.51	1895.491	1853.54	2041	1878.92
1886.9	1900	1855	1470	1865.71	1725.98	1877.657	1860.08	1776	1878.92
2235	1900	1855	1970	1865.71	2006.51	1877.657	2138.21	2041	2247.028
250	2300	2485	2245	2142.04	2520	2311.382	2466.99	2200	2752.743
2394	2300	2415	2470	2245.65	2420	2374.204	2328.48	2422	2340.217
2374.75	2300	2345	2395	2191.75	2365.99	2352.723	2321.66	2422	2340.217
2315.25	2300	2205	2395	2191.75	2365.99	2352.723	2321.66	2422	2247.028

Table 9.14 Statistical performance analysis of EDSPSO-PFTS on SBISP

Model	MSE	AFE	R	R^2	PP	M_{ad}	TS
Chen[114]	35066.3076	8.26	0.8839	0.7813	0.5313	136.32	0.825
Huarng[115]	26909.12	6.29	0.911	0.8314	0.5894	105.3	0.698
Pathak and Singh[122]	42419.52	8.95	0.868	0.7544	0.4845	155.1	-.3.604
Joshi and Kumar[119]	40068.029	9.52	0.882	0.778	0.499	164.3	-4.221
Kumar and Gangwar[106]	17234.438	6.3	0.9446	0.8924	0.6714	101.73	1.713
Bisht and Kumar[107]	32051.741	7.86	0.9001	0.8101	0.5519	131.28	0.882
Proposed Method	9521.94	4.109	0.971	0.943	0.756	66.62	1.848

9.7 Conclusion

This study marks the pioneering introduction of the integration of PFYS and a metaheuristic approach in FTS forecasting within existing literature. The novel hybrid algorithm, named EDSPSO-PFTS, is unveiled as a powerful tool for FTS forecasting, combining the strengths of Exponentially Mutated PSO (EDSPSO) and PFYS. While PSO is widely acclaimed for its procedural advantages, it's susceptible to rash convergence at local optimal point. In response, this work enhances PSO by integrating an innovative EMO, enriching the exploration phase of optimization. Our suggested forecasting method combines EDSPSO and PFYS to add non-determinacy into the fuzzification of TSD and optimise interval lengths.

To underscore the efficiency of the proposed innovative FTS forecasting method, it undergoes application to diverse TSD sets, including UAE and SBISP. The lower values of AFE and MSE stand as empirical proof that, in terms of error metrics, the proposed method surpasses the performance of the compared methods in predicting both UAE and SBISP. The values of R and R^2 affirm the strong correlation between actual and forecasted enrolments.

Future research in FTS forecasting will likely continue to focus on improving prediction performance and lowering computing complexity. Additionally, investigating the integration of nature-based optimization algorithms into PFYS-based FTS is an interesting direction for further investigation. Furthermore, the goal of subsequent research will be to develop a multivariate PFTS model and propose a novel multivariate forecasting method based on this framework.

Part V

Conclusions and Scope for Future Work

(Chapter 10)

Chapter 10

Conclusions and Scope for Future Work

This thesis is dedicated to developing different hybrid metaheuristic algorithms, specifically tailored for job scheduling on computational grids, feature selection, and time series forecasting. The fundamental aim was to examine and elevate the performance capabilities of metaheuristic algorithms in diverse applications. Section 10.1 presents the conclusion of this thesis, and Section 10.2 presents potential research areas for future work.

10.1 Conclusions

In conclusion, this research unfolds to reveal not merely a compilation of findings but a narrative of discovery of innovative metaheuristic algorithms and their applications. The journey cast a brilliant light on the metaheuristic algorithms and their real-world applications under both Fuzzy and Deterministic environments. The primary conclusions from the eight contribution chapters—Chapters 2 through Chapter 9—are covered in this section.

1. Addressing the job scheduling on the computational grid challenge, **Chapter 2** introduces a fuzzy Particle Swarm Optimization (PSO) approach employing both trapezoidal and pentagonal fuzzy numbers. The algorithm's efficacy is examined, initially with trapezoidal fuzzy numbers and subsequently with pentagonal fuzzy numbers. Comparative analysis reveals similar outcomes between fuzzy PSO employing pentagonal and trapezoidal fuzzy numbers.

2. The complicated challenge of multi-objective job scheduling on a computational grid is addressed in **Chapter 3** through fuzzy PSO employing both trapezoidal and pentagonal fuzzy numbers. The makespan and flowtime values serve as optimal criteria, with the selection of a particle that minimizes both simultaneously. The calculated results of fuzzy PSO, employing trapezoidal and pentagonal fuzzy numbers, are systematically compared. Strikingly, despite differences in scheduling, the objective values remain consistent.

3. **Chapter 4** analyses statistical metrics and convergence rates of the proposed HPSO algorithm. It becomes evident that the proposed HPSO algorithm consistently outperforms other metaheuristic algorithms (PSO, GA, HHO, and SSA) considered in this work, particularly on high-dimensional and medium-dimensional datasets. However, it is worth noting that the results on low-dimensional datasets, while satisfactory, do not exhibit the same level of superiority. Since the practical applications of FS problems involve large datasets, the proposed HPSO is more application-oriented and useful. This satisfies our objective to increase CA and decrease the NF.

4. **Chapter 5** analyses the application of the proposed hybrid metaheuristic algorithm, HPSO, to MOFS problems, which involves a thorough comparison with PSO across seven UCI datasets. The experimental outcomes conclusively establish the outperformance of the HPSO algorithm over PSO for MOFS problems, validated through rigorous statistical tests. The algorithm exhibits superior accuracy in higher-dimensional scenarios, while its performance remains comparable on low-dimensional problems. This positions the HPSO algorithm as a promising solution for MOO problems and feature selection challenges.

5. PSOHHO and PSOHHO-V algorithms are developed in **Chapter 6** which is based on Exponential Mutation and Dual-Swarm Strategy. Their performance is evaluated and comprehensively analysed based on statistical metrics and convergence rates. These algorithms were subjected to rigorous testing on ten BF, showcasing superior searchability and convergent speed compared to other optimizers in solving global

optimization problems. Statistical tests further substantiated these findings. Additionally, when applied to feature selection problems using seven UCI datasets, both PSOHHO and PSOHHO-V outperformed other metaheuristic algorithms, particularly in high-dimensional and medium-dimensional scenarios. But on low dimensional datasets, the results are not much useful. The algorithms' consistent success across varied dimensions positions them as potent solutions for practical feature selection challenges involving large datasets.

6. In **Chapter 7**, PSOMHHO algorithm is developed with the help of Trapezoidal and Pentagonal Fuzzy Numbers. The distinctive advantage of PSOMHHO over HHO is prominently displayed in its Signature. The provision of a convergence proof serves to strengthen the mathematical foundation, affirming the strength and reliability of the PSOMHHO algorithm. Employing BF with distinct properties such as Unimodal Peak (UP), Multimodal Peak (MP), and Fixed-Dimension Multimodal BF, the PSOMHHO algorithm consistently outperforms HHO, GA, PSO, and SSA in terms of mean fitness values across dimensions of 100, 500, and 1000. Statistical measures, including MWUT and the Friedman test, affirm the statistical significance of these results. Notably, the observed superior fitness values across different BF types position PSOMHHO as a promising algorithm applicable to diverse real-world challenges.

7. Fuzzy BTO algorithms are developed in **Chapter 8** and their effectiveness, particularly BTOF1, was rigorously assessed through statistical metrics and convergence rates across nine BF. The comprehensive experimental outcomes affirm the superior searchability and convergent speed of BTOF1 compared to other optimizers in solving global optimization challenges. Statistical tests further validate this conclusion. Additionally, when applied to feature selection problems using seven UCI datasets, BTOF1 consistently outperformed alternative algorithms, establishing its potential as a robust global optimization solution adept at addressing feature selection challenges.

8. **Chapter 9** presents the concept of incorporating PFYS and a metaheuristic method in FTS forecasting. No significant work is available in the literature which uses the above concept. With the combined strength of EDSPSO and PFYS, a novel hybrid algorithm (EDSPSO-PFTS) for FTS forecasting is presented in this work. Because of its

procedural advantages, PSO is a well-liked optimization technique; yet, it is prone to premature convergence at local optima. In order to overcome this constraint, we improve PSO by presenting the idea of exponential mutation, which makes the exploration stage of the optimization procedure easier. The suggested forecasting technique incorporates non-determinacy into the fuzzification of TSD and optimizes interval durations by combining EDSPSO and PFYS. Our innovative hybrid approach finds the ideal length for discretization and addresses non-determinacy in the fuzzification process, in contrast to previous PSO-based FTS forecasting models that ignored non-determinacy without explanation.

The suggested unique hybrid FTS forecasting method is applied to a variety of TSD, such as the UAE and the SBISP, to illustrate its effects. The suggested strategy performs better than the compared methods in terms of error measures when it comes to forecasting both the UAE and the SBISP, as shown by the lower values of AFE and MSE. Greater collaboration between anticipated and actual enrolments is ensured by both R and R^2 values. Additionally, the numerical values of TS and PP are within the predicted range, confirming the objectivity of the enrolment estimates generated with the suggested methodology.

10.2 Future Work

As we embark on the conclusion of this research journey, the horizon of possibilities for future work extends with promise. The groundwork laid in this thesis serves as a trigger for the exploration of several intriguing opportunities that could enhance and expand the territories of development and application of metaheuristic algorithms in fuzzy environment. In the chapters to come, we explore into the potential trajectories for further investigation for the continued evolution of knowledge in this dynamic domain.

1. The intriguing results indicates exploring of alternative fuzzy numbers to recognize potential variations and analyse their impact on the algorithm's performance.

2. The stability analysis of the proposed algorithms given in Chapters 6, 7 and 8 has not been discussed. Hence in future the stability analysis of the proposed algorithms can be discussed.

3. Single-Objective FS problems discussed in Chapters 6, 7 and 8 can be extended work into multi-objective FS problems.

4. As discussed in Chapter 9, exploring the incorporation of alternative nature-based optimization algorithms within PFYS-based FTS represents a promising avenue for future research. Additionally, future studies will aim to establish a multivariate PFTS model and suggest a new multivariate forecasting algorithm centred on this framework.

5. Improving forecast performance and reducing computational complexity continue to be important areas of focus for future researchers in FTS forecasting.

6. In the future, the applications of the proposed algorithms can be expanded to real-life problems such as machine learning, medical applications, financial fields, and engineering optimization tasks.

REFERENCES

1. Zhang, Q. *et al.* Chaos Enhanced Bacterial Foraging Optimization for Global Optimization. *IEEE Access* **6,** 64905–64919 (2018).

2. Mirjalili, S. & Lewis, A. The Whale Optimization Algorithm. *Adv. Eng. Softw.* **95,** 51–67 (2016).

3. Opara, K. R. & Arabas, J. Differential Evolution: A survey of theoretical analyses. *Swarm Evol. Comput.* **44,** 546–558 (2019).

4. Heidari, A. A., Faris, H., Aljarah, I. & Mirjalili, S. An efficient hybrid multilayer perceptron neural network with grasshopper optimization. *Soft Comput.* **23,** 7941–7958 (2019).

5. Kennedy, J. & Eberhart, R. Particle swarm optimization. in *Proceedings of ICNN'95 - International Conference on Neural Networks* **4,** 1942–1948 (IEEE).

6. Gandomi, A. H., Yang, X.-S. & Alavi, A. H. Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Eng. Comput.* **29,** 17–35 (2013).

7. Dorigo, M. & Gambardella, L. M. Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Trans. Evol. Comput.* **1,** 53–66 (1997).

8. Rashedi, E., Nezamabadi-pour, H. & Saryazdi, S. GSA: A Gravitational Search Algorithm. *Inf. Sci. (Ny).* **179,** 2232–2248 (2009).

9. Wolpert, D. H. & Macready, W. G. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1,** 67–82 (1997).

10. Mangla, C., Ahmad, M. & Uddin, M. Optimization of complex nonlinear systems using genetic algorithm. *Int. J. Inf. Technol.* **13,** 1913–1925 (2021).

11. Rahkar Farshi, T. & Orujpour, M. Multi-level image thresholding based on social spider algorithm for global optimization. *Int. J. Inf. Technol.* **11,** 713–718 (2019).

12. Yang, X. & Hossein Gandomi, A. Bat algorithm: a novel approach for global engineering optimization. *Eng. Comput.* **29,** 464–483 (2012).

13. Li, S., Chen, H., Wang, M., Heidari, A. A. & Mirjalili, S. Slime mould algorithm: A new method for stochastic optimization. *Futur. Gener. Comput. Syst.* **111,** 300–323 (2020).

14. Aggarwal, D. & Kumar, V. Performance evaluation of distance metrics on Firefly Algorithm for VRP with time windows. *Int. J. Inf. Technol.* **13,** 2355–2362 (2021).

15. Mirjalili, S. *et al.* Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Adv. Eng. Softw.* **114,** 163–191 (2017).

16. Mirjalili, S., Mirjalili, S. M. & Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **69,** 46–61 (2014).

17. Mirjalili, S., Mirjalili, S. M. & Hatamlou, A. Multi-Verse Optimizer: a nature-inspired algorithm for global optimization. *Neural Comput. Appl.* **27,** 495–513 (2016).

18. Xu, G. *et al.* On convergence analysis of multi-objective particle swarm optimization algorithm. *Eur. J. Oper. Res.* **286,** 32–38 (2020).

19. Wu, D., Xu, S. & Kong, F. Convergence Analysis and Improvement of the Chicken Swarm Optimization Algorithm. *IEEE Access* **4,** 9400–9412 (2016).

20. Zadeh, L. A. Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. *Fuzzy Sets Syst.* **90,** 111–127 (1997).

21. Tran, B., Xue, B. & Zhang, M. Overview of Particle Swarm Optimisation for Feature Selection in Classification. in *Simulated Evolution and Learning: 10th International Conference, SEAL 2014, Dunedin, New Zealand, December 15-18, 2014. Proceedings 10* 605–617 (Springer, 2014). doi:10.1007/978-3-319-13563-2_51

22. Houssein, E. H., Gad, A. G., Hussain, K. & Suganthan, P. N. Major Advances in Particle Swarm Optimization: Theory, Analysis, and Application. *Swarm Evol. Comput.* **63,** 100868 (2021).

23. Shami, T. M. *et al.* Particle Swarm Optimization: A Comprehensive Survey. *IEEE Access* **10,** 10031–10061 (2022).

24. Pant, M. & Kumar, S. Fuzzy time series forecasting based on hesitant fuzzy sets, particle swarm optimization and support vector machine-based hybrid method. *Granul. Comput.* **7,** 861–879 (2022).

25. Pant, M. & Kumar, S. Particle swarm optimization and intuitionistic fuzzy set-based novel method for fuzzy time series forecasting. *Granul. Comput.* **7,** 285–303 (2022).

26. Mirjalili, S. in 43–55 (2019). doi:10.1007/978-3-319-93025-1_4

27. Eiben, A. E. & Smith, J. E. *Introduction to Evolutionary Computing*. (Springer Berlin Heidelberg, 2015). doi:10.1007/978-3-662-44874-8

28. Yuan, S., Li, T. & Wang, B. A co-evolutionary genetic algorithm for the two-machine flow shop group scheduling problem with job-related blocking and transportation times. *Expert*

Syst. Appl. **152,** 113360 (2020).

29. Cruz-Piris, L., Lopez-Carmona, M. A. & Marsa-Maestre, I. Automated Optimization of Intersections Using a Genetic Algorithm. *IEEE Access* **7,** 15452–15468 (2019).

30. Defersha, F. M. & Rooyani, D. An efficient two-stage genetic algorithm for a flexible job-shop scheduling problem with sequence dependent attached/detached setup, machine release date and lag-time. *Comput. Ind. Eng.* **147,** 106605 (2020).

31. Ghannami, A., Li, J., Hawbani, A. & Al-Dubai, A. Stratified opposition-based initialization for variable-length chromosome shortest path problem evolutionary algorithms. *Expert Syst. Appl.* **170,** 114525 (2021).

32. Kobeaga, G., Merino, M. & Lozano, J. A. An efficient evolutionary algorithm for the orienteering problem. *Comput. Oper. Res.* **90,** 42–59 (2018).

33. Park, H., Son, D., Koo, B. & Jeong, B. Waiting strategy for the vehicle routing problem with simultaneous pickup and delivery using genetic algorithm. *Expert Syst. Appl.* **165,** 113959 (2021).

34. Remesh Babu, K. R. & Samuel, P. in 67–78 (2016). doi:10.1007/978-3-319-28031-8_6

35. Tsai, C.-W. & Rodrigues, J. J. P. C. Metaheuristic Scheduling for Cloud: A Survey. *IEEE Syst. J.* **8,** 279–291 (2014).

36. Chen, G. & Chen, J. A novel wrapper method for feature selection and its applications. *Neurocomputing* **159,** 219–226 (2015).

37. Simumba, N., Okami, S., Kodaka, A. & Kohtake, N. Multiple objective metaheuristics for feature selection based on stakeholder requirements in credit scoring. *Decis. Support Syst.* **155,** 113714 (2022).

38. Abdollahzadeh, B. & Gharehchopogh, F. S. A multi-objective optimization algorithm for feature selection problems. *Eng. Comput.* **38,** 1845–1863 (2022).

39. Dua, D. & Casey, G. UCI machine learning repository, 2017. *URL http//archive. ics. uci. edu/ml* **7,** (2017).

40. Song, Q. & Chissom, B. S. Fuzzy time series and its models. *Fuzzy Sets Syst.* **54,** 269–277 (1993).

41. Zeng, S., Chen, S.-M. & Teng, M. O. Fuzzy forecasting based on linear combinations of independent variables, subtractive clustering algorithm and artificial bee colony algorithm. *Inf. Sci. (Ny).* **484,** 350–366 (2019).

42. Sadaei, H. J., de Lima e Silva, P. C., Guimarães, F. G. & Lee, M. H. Short-term load forecasting by using a combined method of convolutional neural networks and fuzzy time series. *Energy* **175,** 365–377 (2019).

43. Wang, G.-Y., Cheng, D.-D., Xia, D.-Y. & Jiang, H.-H. Swarm Intelligence Research: From Bio-inspired Single-population Swarm Intelligence to Human-machine Hybrid Swarm Intelligence. *Mach. Intell. Res.* **20,** 121–144 (2023).

44. Li, D.-J., Li, Y.-Y., Li, J.-X. & Fu, Y. Gesture Recognition Based on BP Neural Network Improved by Chaotic Genetic Algorithm. *Int. J. Autom. Comput.* **15,** 267–276 (2018).

45. Abhishekh, Gautam, S. S. & Singh, S. R. A Score Function-Based Method of Forecasting Using Intuitionistic Fuzzy Time Series. *New Math. Nat. Comput.* **14,** 91–111 (2018).

46. Egrioglu, E., Yolcu, U. & Bas, E. Intuitionistic high-order fuzzy time series forecasting method based on pi-sigma artificial neural networks trained by artificial bee colony. *Granul. Comput.* **4,** 639–654 (2019).

47. Abraham, A., Liu, H. & Zhao, M. in 327–342 (2008). doi:10.1007/978-3-540-78985-7_13

48. Johnson, D. S. The NP-completeness column. *ACM Trans. Algorithms* **2,** 473–489 (2006).

49. Boindala, S. P. & Arunachalam, V. in 119–126 (2020). doi:10.1007/978-981-15-0751-9_11

50. Zaheer, H. & Pant, M. in 191–199 (2018). doi:10.1007/978-981-10-5699-4_19

51. Nayak, S. K., Panda, C. S. & Padhy, S. K. in 131–147 (2018). doi:10.1007/978-981-10-8049-4_7

52. Nguyen, H. B., Xue, B., Liu, I. & Zhang, M. Filter based backward elimination in wrapper based PSO for feature selection in classification. in *2014 IEEE Congress on Evolutionary Computation (CEC)* 3111–3118 (IEEE, 2014). doi:10.1109/CEC.2014.6900657

53. Mohammadzadeh, H. & Gharehchopogh, F. S. A novel hybrid whale optimization algorithm with flower pollination algorithm for feature selection: Case study Email spam detection. *Comput. Intell.* **37,** 176–209 (2021).

54. Neshatian, K. & Zhang, M. in 97–108 (2012). doi:10.1007/978-3-642-29139-5_9

55. Xue, B., Cervante, L., Shang, L., Browne, W. N. & Zhang, M. Multi-Objective Evolutionary Algorithms For Filter Based Feature Selection In Classification. *Int. J. Artif. Intell. Tools* **22,** 1350024 (2013).

56. Xue, B., Zhang, M. & Browne, W. N. Particle Swarm Optimization for Feature Selection in Classification: A Multi-Objective Approach. *IEEE Trans. Cybern.* **43,** 1656–1671 (2013).

57. Niyomsat, T., Hlangnamthip, S. & Puangdownreong, D. in 154–163 (2021). doi:10.1007/978-3-030-68154-8_16

58. Yücel, M., Bekdaş, G. & Nigdeli, S. M. in 592–601 (2022). doi:10.1007/978-3-030-93247-3_58

59. Hanchuan Peng, Fuhui Long & Ding, C. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans. Pattern Anal. Mach. Intell.* **27,** 1226–1238 (2005).

60. Fay, M. P. & Proschan, M. A. Wilcoxon-Mann-Whitney or t-test? On assumptions for hypothesis tests and multiple interpretations of decision rules. *Stat. Surv.* **4,** (2010).

61. Zhang, Y., Liu, R., Wang, X., Chen, H. & Li, C. Boosted binary Harris hawks optimizer and feature selection. *Eng. Comput.* **37,** 3741–3770 (2021).

62. Chabbouh, M., Bechikh, S., Hung, C.-C. & Ben Said, L. Multi-objective evolution of oblique decision trees for imbalanced data binary classification. *Swarm Evol. Comput.* **49,** 1–22 (2019).

63. Miao, J. & Niu, L. A Survey on Feature Selection. *Procedia Comput. Sci.* **91,** 919–926 (2016).

64. La Cava, W. *et al.* Multidimensional genetic programming for multiclass classification. *Swarm Evol. Comput.* **44,** 260–272 (2019).

65. Xue, B., Zhang, M., Browne, W. N. & Yao, X. A Survey on Evolutionary Computation Approaches to Feature Selection. *IEEE Trans. Evol. Comput.* **20,** 606–626 (2016).

66. Liu, Y., Tang, F. & Zeng, Z. Feature Selection Based on Dependency Margin. *IEEE Trans. Cybern.* **45,** 1209–1221 (2015).

67. Miguel Antonio, L. & Coello Coello, C. A. Coevolutionary Multiobjective Evolutionary Algorithms: Survey of the State-of-the-Art. *IEEE Trans. Evol. Comput.* **22,** 851–865 (2018).

68. Tran, B., Xue, B. & Zhang, M. Variable-Length Particle Swarm Optimization for Feature Selection on High-Dimensional Classification. *IEEE Trans. Evol. Comput.* **23,** 473–487 (2019).

69. Nguyen, B. H., Xue, B. & Zhang, M. A survey on swarm intelligence approaches to feature selection in data mining. *Swarm Evol. Comput.* **54,** 100663 (2020).

70. Zhang, Y., Cheng, S., Shi, Y., Gong, D. & Zhao, X. Cost-sensitive feature selection using two-archive multi-objective artificial bee colony algorithm. *Expert Syst. Appl.* **137,** 46–58 (2019).

71. Wang, X., Zhang, Y., Sun, X., Wang, Y. & Du, C. Multi-objective feature selection based on

artificial bee colony: An acceleration approach with variable sample size. *Appl. Soft Comput.* **88,** 106041 (2020).

72. Wei, W., Chen, S., Lin, Q., Ji, J. & Chen, J. A multi-objective immune algorithm for intrusion feature selection. *Appl. Soft Comput.* **95,** 106522 (2020).

73. Zhang, Y., Gong, D., Gao, X., Tian, T. & Sun, X. Binary differential evolution with self-learning for multi-objective feature selection. *Inf. Sci. (Ny).* **507,** 67–85 (2020).

74. Mahapatra, A. K., Panda, N. & Pattanayak, B. K. Quantized Salp Swarm Algorithm (QSSA) for optimal feature selection. *Int. J. Inf. Technol.* **15,** 725–734 (2023).

75. Salcedo-Sanz, S. Modern meta-heuristics based on nonlinear physics processes: A review of models and design procedures. *Phys. Rep.* **655,** 1–70 (2016).

76. Ranjan, R. & Chhabra, J. K. Automatic feature selection using enhanced dynamic Crow Search Algorithm. *Int. J. Inf. Technol.* **15,** 2777–2782 (2023).

77. Jovic, A., Brkic, K. & Bogunovic, N. A review of feature selection methods with applications. in *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)* 1200–1205 (IEEE, 2015). doi:10.1109/MIPRO.2015.7160458

78. Molaei, S., Moazen, H., Najjar-Ghabel, S. & Farzinvash, L. Particle swarm optimization with an enhanced learning strategy and crossover operator. *Knowledge-Based Syst.* **215,** 106768 (2021).

79. Clerc, M. *Guided Randomness in Optimization*. (John Wiley & Sons, Inc., 2015). doi:10.1002/9781119136439

80. Solis, F. J. & Wets, R. J.-B. Minimization by Random Search Techniques. *Math. Oper. Res.* **6,** 19–30 (1981).

81. Jiang, M., Luo, Y. P. & Yang, S. Y. Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm. *Inf. Process. Lett.* **102,** 8–16 (2007).

82. Chakraborty. Genetic algorithm with fuzzy fitness function for feature selection. in *Proceedings of the IEEE International Symposium on Industrial Electronics ISIE-02* 315–319 vol.1 (IEEE, 2002). doi:10.1109/ISIE.2002.1026085

83. Haibin Duan, Daobo Wang & Xiufen Yu. Markov Chains and Martingale Theory Based Convergence Proof of Ant Colony Algorithm and Its Simulation Platform. in *2006 6th World Congress on Intelligent Control and Automation* 3057–3061 (IEEE, 2006). doi:10.1109/WCICA.2006.1712928

84. Clerc, M. & Kennedy, J. The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans. Evol. Comput.* **6,** 58–73 (2002).

85. Suzuki, J. A Markov chain analysis on simple genetic algorithms. *IEEE Trans. Syst. Man. Cybern.* **25,** 655–659 (1995).

86. Suzuki, J. A further result on the Markov chain model of genetic algorithms and its application to a simulated annealing-like strategy. *IEEE Trans. Syst. Man Cybern. Part B* **28,** 95–102 (1998).

87. Xu, G., Wu, Z. H. & Jiang, M. Z. Premature convergence of standard particle swarm optimisation algorithm based on Markov chain analysis. *Int. J. Wirel. Mob. Comput.* **9,** 377 (2015).

88. Gidas, B. Nonstationary Markov chains and convergence of the annealing algorithm. *J. Stat. Phys.* **39,** 73–131 (1985).

89. Xin Yao, Yong Liu & Guangming Lin. Evolutionary programming made faster. *IEEE Trans. Evol. Comput.* **3,** 82–102 (1999).

90. Digalakis, J. G. & Margaritis, K. G. On benchmarking functions for genetic algorithms. *Int. J. Comput. Math.* **77,** 481–506 (2001).

91. Agarwal, P., Dutta, A., Agrawal, T., Mehra, N. & Mehta, S. Hybrid Nature-Inspired Algorithm for Feature Selection in Alzheimer Detection Using Brain MRI Images. *Int. J. Comput. Intell. Appl.* **21,** (2022).

92. Sadeghi, H. & Ajoudanian, S. Optimized Feature Selection in Software Product Lines using Discrete Bat Algorithm. *Int. J. Comput. Intell. Appl.* **21,** (2022).

93. Heidari, A. A. *et al.* An enhanced associative learning-based exploratory whale optimizer for global optimization. *Neural Comput. Appl.* **32,** 5185–5211 (2020).

94. Zadeh, L. A. Fuzzy sets. *Inf. Control* **8,** 338–353 (1965).

95. Zadeh, L. A. Fuzzy logic. *Computer (Long. Beach. Calif).* **21,** 83–93 (1988).

96. Ganie, A. H. Applicability of a novel Pythagorean fuzzy correlation coefficient in medical diagnosis, clustering, and classification problems. *Comput. Appl. Math.* **41,** 410 (2022).

97. Ganie, A. H. A picture fuzzy distance measure and its application to pattern recognition problems. *Iran. J. Fuzzy Syst.* **20,** 71–85 (2023).

98. Ganie, A. H., Singh, S., Khalaf, M. M. & Al-Shamiri, M. M. A. On some measures of similarity and entropy for Pythagorean fuzzy sets with their applications. *Comput. Appl. Math.*

41, 420 (2022).

99. Kocak, C. ARMA(p,q) type high order fuzzy time series forecast method based on fuzzy logic relations. *Appl. Soft Comput.* **58,** 92–103 (2017).

100. Güler Dincer, N. & Akkuş, Ö. A new fuzzy time series model based on robust clustering for forecasting of air pollution. *Ecol. Inform.* **43,** 157–164 (2018).

101. Güler Dincer, N. A New Fuzzy Time Series Model Based on Fuzzy C-Regression Model. *Int. J. Fuzzy Syst.* **20,** 1872–1887 (2018).

102. Yolcu, U., Bas, E. & Egrioglu, E. A new fuzzy inference system for time series forecasting and obtaining the probabilistic forecasts via subsampling block bootstrap. *J. Intell. Fuzzy Syst.* **35,** 2349–2358 (2018).

103. Jiang, P., Yang, H. & Heng, J. A hybrid forecasting system based on fuzzy time series and multi-objective optimization for wind speed forecasting. *Appl. Energy* **235,** 786–801 (2019).

104. Tran, N., Nguyen, T., Nguyen, B. M. & Nguyen, G. A Multivariate Fuzzy Time Series Resource Forecast Model for Clouds using LSTM and Data Correlation Analysis. *Procedia Comput. Sci.* **126,** 636–645 (2018).

105. Joshi, B. P., Mukesh Pandey & Sanjay Kumar. in 843–852 (2016). doi:10.1007/978-981-10-0448-3_70

106. Kumar, S. & Gangwar, S. S. Intuitionistic Fuzzy Time Series: An Approach for Handling Nondeterminism in Time Series Forecasting. *IEEE Trans. Fuzzy Syst.* **24,** 1270–1281 (2016).

107. Bisht, K. & Kumar, S. Fuzzy time series forecasting method based on hesitant fuzzy sets. *Expert Syst. Appl.* **64,** 557–568 (2016).

108. Thong, N. T. & Son, L. H. HIFCF: An effective hybrid model between picture fuzzy clustering and intuitionistic fuzzy recommender systems for medical diagnosis. *Expert Syst. Appl.* **42,** 3682–3701 (2015).

109. Thong, P. H. & Son, L. H. Picture fuzzy clustering: a new computational intelligence method. *Soft Comput.* **20,** 3549–3562 (2016).

110. Son, L. H., Van Viet, P. & Van Hai, P. Picture inference system: a new fuzzy inference system on picture fuzzy set. *Appl. Intell.* **46,** 652–669 (2017).

111. Chakraverty, S. & Gupta, P. Comparison of neural network configurations in the long-range forecast of southwest monsoon rainfall over India. *Neural Comput. Appl.* **17,** 187–192 (2008).

112. Wang, J.-W. & Liu, J.-W. in 408–415 (2010). doi:10.1007/978-3-642-12145-6_42

113. Pattanayak, R. M., Panigrahi, S. & Behera, H. S. High-Order Fuzzy Time Series Forecasting by Using Membership Values Along with Data and Support Vector Machine. *Arab. J. Sci. Eng.* **45,** 10311–10325 (2020).

114. Chen, S.-M. Forecasting enrollments based on fuzzy time series. *Fuzzy Sets Syst.* **81,** 311–319 (1996).

115. Huarng, K. Effective lengths of intervals to improve forecasting in fuzzy time series. *Fuzzy Sets Syst.* **123,** 387–394 (2001).

116. Lee, H.-S. & Chou †, M.-T. Fuzzy forecasting based on fuzzy time series. *Int. J. Comput. Math.* **81,** 781–789 (2004).

117. Yolcu, U., Egrioglu, E., Uslu, V. R., Basaran, M. A. & Aladag, C. H. A new approach for determining the length of intervals for fuzzy time series. *Appl. Soft Comput.* **9,** 647–651 (2009).

118. Qiu, W., Liu, X. & Li, H. A generalized method for forecasting based on fuzzy time series. *Expert Syst. Appl.* **38,** 10446–10453 (2011).

119. Joshi, B. P. & Kumar, S. Intuitinonistic fuzzy sets based method for fuzzy time series forecasting. *Cybern. Syst.* **43,** 34–47 (2012).

120. Gupta, K. K. & Kumar, S. Hesitant probabilistic fuzzy set based time series forecasting method. *Granul. Comput.* **4,** 739–758 (2019).

121. Chen, S.-M. & Chung, N.-Y. Forecasting enrollments using high-order fuzzy time series and genetic algorithms. *Int. J. Intell. Syst.* **21,** 485–501 (2006).

122. Pathak, H. K. & Singh, P. A New Bandwidth Interval Based Forecasting Method for Enrollments Using Fuzzy Time Series. *Appl. Math.* **02,** 504–507 (2011).

123. Song, Q. & Chissom, B. S. Forecasting enrollments with fuzzy time series — part II. *Fuzzy Sets Syst.* **62,** 1–8 (1994).

List of Published Papers

1. Dutta, D., & Rath, S. (2021). Scheduling of jobs on computational grids by fuzzy particle swarm optimization algorithm using trapezoidal and pentagonal fuzzy numbers. Computational Sciences-Modelling, Computing and Soft Computing: Vol 1, pp. 175-185. Springer.

2. Dutta, D., & Rath, S. (2022). Job scheduling on computational grids using multi-objective fuzzy particle swarm optimization. Soft Computing: Theories and Applications: Proceedings of SoCTA 2020, Vol 1, pp. 333-347. Springer.

3. Dutta, D., & Rath, S. (2022). Hybrid Particle Swarm Optimization for a Feature Selection Problem with Stability Analysis. Intelligent Computing & Optimization, Vol 569, pp. 991-1004. Springer.

4. Rath, S., & Dutta, D. (2023). A hybrid swarm optimization with trapezoidal and pentagonal fuzzy numbers using benchmark functions. International Journal of Information Technology, Vol 15, pp.1-12, Springer.

5. Dutta, D., & Rath, S. (2023) Innovative Hybrid Metaheuristic Algorithms: Exponential Mutation and Dual-Swarm Strategy for Hybrid Feature Selection Problem, International Journal of Information Technology.

6. Rath, S., & Dutta, D. (2024) Picture Fuzzy Time Series Forecasting with a novel variant of Particle Swarm Optimization, SN Computer Science, Vol 5.

List of Communicated Papers

1. Rath, S., & Dutta, D., Comparative study between Hybrid Particle Swarm Optimization and Particle Swarm optimization on a Multi-Objective Feature Selection Problem.

2. Rath, S., & Dutta, D., Bluefin Trevally Optimizer (BTO): A Metaheuristic Algorithm Using Fuzzy Logic Controller for Feature Selection Problem.