DESIGN OF SINGLE AND MULTI-OBJECTIVE METAHEURISTIC ALGORITHMS FOR EFFECTIVE DATA CLUSTERING

Thesis submitted in fulfilment for the requirement of the Degree of

Doctor of Philosophy

By

ARVINDER KAUR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING AND INFORMATION TECHNOLOGY

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY,

WAKNAGHAT, SOLAN-173234, HIMACHAL PRADESH, INDIA

JUNE 2022

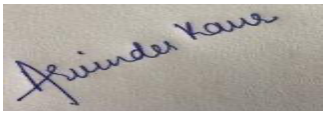
SUPERVISOR’S CERTIFICATE

This is to certify that the work reported in Ph.D. thesis entitled “**Design of Single and Multi-objective Metaheuristic Algorithms for Effective Data Clustering”** by **Arvinder Kaur** at **Jaypee University of Information Technology, Waknaghat, Solan (HP), India** is a bonafide record of her original work carried out under my supervision. This work has not been submitted elsewhere for any other degree or diploma.

Dr. Yugal Kumar
Associate Professor,
Department of Computer Science & Engineering and Information Technology
Jaypee University of Information Technology, Waknaghat, Solan (HP), India.
Date:

DECLARATION BY THE SCHOLAR

I hereby declare that work reported in Ph.D. thesis entitled "**Design of Single and Multi-objective Metaheuristic Algorithms for Effective Data Clustering"** submitted at **Jaypee University of Information Technology, Waknaghat, Solan (HP), India** is an authentic work carried out under the supervision of **Dr. Yugal Kumar**. I have not this work elsewhere for any other degree or diploma. I am fully responsible for the content of my Ph.D. thesis.



Arvinder Kaur
Enrollment Number: 196202
Department of Computer Science & Engineering and Information Technology
Jaypee University of Information Technology, Waknaghat, Solan (HP), India.
Date:

ACKNOWLEDGEMENT

I would like to express my deep sense of respect and gratitude towards my guide Dr. Yugal Kumar, Associate Professor in the Department of Computer Science and Engineering & Information Technology, for his support throughout this research work. I want to thank him for introducing me to the field of Data Mining and giving me the opportunity to work under him. Without his invaluable advice and assistance, it would not have been possible for me to complete this thesis. I am greatly indebted to him for his constant encouragement and invaluable advice in every aspect of my academic life. I consider it my good fortune to have got an opportunity to work with such a wonderful person.

I would like to express my gratitude to our Honorable Vice Chancellor, Prof. (Dr.) Rajendra Kumar Sharma to promote the research and facilitate resources in the institution. I would also like to thank Prof. (Dr.) Vivek Sehgal, Head Department of CSE & IT for constant guidance, research facilities and resources to carry out my research work. I would also like to thank my doctoral committee members Dr. Rajinder Sandhu, Dr. Amit Kumar, and Prof. (Dr.) Vineet Sharma for their valuable feedback and critical reviews during presentations and time to time help.

I am also grateful for the support received from the JUIT, Waknaghat (Himachal Pradesh). I thank all staff at the Department of Computer Science and Engineering & Information Technology, JUIT, Waknaghat (Himachal Pradesh) who has been extremely helpful at numerous occasions. I also thank my friends for their consistent help and valuable discussion. Last but not the least, I would like to thank my family for supporting me spiritually.

Arvinder Kaur

ABSTRACT

Clustering is an important data analysis technique to find similar data objects in a given dataset. It is unsupervised learning and has proven its capability in diverse research fields such as medical diagnosis, market segmentation, image segmentation, customer behaviour analysis, outlier detection, and feature selection. Clustering aims to determine the set of identical data objects and put these data objects into a single cluster. The data objects within the clusters have more similar characteristics than other clusters. The research community presents several clustering techniques- partitional, hierarchal, model-based, grid-based, density-based, etc. But, the popular one is partitional clustering. This thesis work focuses on partitional clustering. In partitional clustering, a dataset divides into k number of partitions known as clusters. A distance function is utilized for allocating the data objects to clusters based on minimum distance. However, the number of clusters (k) should be known in advance. The performances of partitional clustering algorithms depend on the selection of initial cluster centroids. Several traditional algorithms, like K-Means, K-Mediods, K-Harmonic Mean etc., are successfully implemented for solving partitional clustering problems. But, these algorithms have several drawbacks, such as being sensitive to initial cluster selection, local optima, convergence rate and predefined method for updating cluster centroids. Several researchers explore metaheuristic algorithms capabilities to overcome the issues of traditional clustering algorithms. These are GA, PSO, ACO, ABC, TS, SA etc., and provide state-of-the-art clustering results for partitional clustering problems. However, some issues are also associated with metaheuristic algorithms, such as an imbalance in local search and global search mechanisms, population diversity, sometimes stuck in local optima, and population generation. This thesis work considers the aforementioned problems of metaheuristic algorithms and proposes new algorithms to handle the partitional clustering problems efficiently. This thesis presents two new partitional clustering algorithms, improved water wave optimization (IWWO) and improved bat (IBAT) algorithm for clustering problems. The WWO algorithm is improved using the global best direction and decay operator concept. IBAT is an improved variant of the bat algorithm. It is seen that several shortcomings are associated with the bat algorithm, such as population initialization, local optima and convergence rate. These issues of the bat algorithm are successfully resolved in the IBAT algorithm using an enhanced cooperative coevolution strategy, an elitist strategy and a neighbourhood-search scheme. The performance is evaluated using well-known benchmark clustering datasets and compared with several existing clustering

algorithms. A set of performance parameters also validate the results of IWWO and IBAT algorithms. Both algorithms successfully overcome the issues related to WWO and BAT algorithms. During the experiment, it is seen that a single objective function is considered for solving the clustering problems. Still, sometimes, it generates a biased solution due to a single objective function for handling clustering problems. The biasing issue can be addressed effectively using more than one objective function. This thesis also presents a multiobjective clustering algorithm for handling the biasing issue. In multiobjective clustering, two objective functions are considered that conflict with each other. In this thesis, Euclidean distance and connectedness are objective functions for multiobjective clustering. Further, these functions are integrated into the vibrating particle system algorithm, MOVPS. The simulation results of MOVPS are compared with several multiobjective and single-objective clustering algorithms. The proposed MOVPS achieves far better clustering results than single and multiobjective clustering algorithms.

Keywords: Clustering, Single Objective Optimization, Multiobjective Optimization, Water Wave Optimization, Bat Optimization, Vibrating Particle System.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

ABC	Artificial Bee colony
ACO	Ant Colony Optimization
ACRO	Artificial Chemical Reaction Optimization
ADENS	Adaptive Differential Evolution with Neighborhood Search
ALO	Ant Lion Optimization
ARI	Adjusted rand index
BB-BC	Big bang-big crunch
BGSA	Binary Gravitational Search Algorithm
BH	Black Hole
BIRCH	Balanced Iterative Reducing and Clustering Using Hierarchies
CBPSO	Cooperative bare bone PSO
CH	Calinski–Harabasz
Chaotic TLBO	Chaotic Teaching Learning based Optimization
CHHO	Chaotic Harris Hawk's Optimization
CIEFPA	compound intensified exploration FPA
CMC	Contraceptive Method Choice
CRO-SL	Coral-Reef Optimization with Substrate-Layers
CSO	Cat Swarm Optimization
CURE	Clustering Using REpresentatives
Δ–MOXK	Improved multiobjective clustering using automatic k-determination
DBI	Davies–Bouldin Index
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DE	Differential Evolution
DI	Dunn Index
DIANA	Divisive Analysis
ELMC	Electromagnetic Clustering
EM	Expectation-Maximization
ESA	Elephant Search Algorithm
FPA	Flower Pollination algorithm
FRC	Fuzzy Relational Clustering
GA	Genetic Algorithms
GKA	Genetic K-Means

GSA	Gravitational Search Algorithm
GWO	Grey Wolf Optimizer
H-KHA	Krill Herd algorithm with Harmony search
HKA	Heuristic Kalman filtering algorithm
ICMPKHM	Improved Cuckoo search and modified PSO with K-Harmonic means
ICSO	Improved Cat Swarm Optimization
IIEFPA	inward intensified exploration FPA
IKH	Improved Krill Herd
K-MWO	K-means mussels wandering optimization
KDD	Knowledge Discovery in Databases
LBH	Levy Flight Black Hole
LD	Liver Disorder
MABC	Multiobjective Artificial Bee Colony
MBOA	Modified Butterfly Optimization Algorithm
MCSS	Magnetic charged system search
MEBB-BC	Memory-enriched Big bang-big crunch
MOVPS	Multiobjective vibrating particle system
MONA	Monothetic Analysis
MOO	Multiobjective Optimization
MOPSO	Multiobjective Particle Swarm Optimization
MPGO	Memetic Particle Gravitation Optimization
NMI	Normalized Mutual Index
NSABC	Non-Dominated Sorting Based Multi-Objective Artificial Bee Colony Algorithm
NSGA-II	Non-Dominated Sorting Genetic Algorithm II
ONVG	Overall Non-dominated Vector Generation
OPTICS	Ordering Points To Identify Cluster Structure
PF	Pareto Front
PSO	Particle Swarm Optimization
PSO-BB-BC	PSO based Big-bang-big crunch
SOS	Symbiotic Organism Search
SSD	Sum of squared distance
SSE	Sum of squared error
STING	Statistical Information Grid-based
TAE	Teaching Assistant Evaluation
TLBO	Teaching learning based Optimization

TS	Tabu Search
TSMPSO	Tabu Search with Multiobjective PSO
VPS	Vibrating Particle System
WBC	Wisconsin Breast Cancer
WWO	Water Wave Optimization

CHAPTER 1
INTRODUCTION

Digital data is growing day by day due to advancements in technology. Enormous data is generated by various sectors such as biology, banking, telecom, scientific experiments in space explorations, and business transactions. The internet also produces big data in text, images, and multimedia. Such data consist of many undiscovered patterns and information, and these patterns can be helpful in several domains for decision-making. The hands-on examination of such immense data is quite difficult due to its size and complexity. Therefore, it is time to develop new algorithms to handle such massive data for analysis, interpretation and extracting the hidden patterns. Data mining is referred to as Knowledge Discovery in Databases (KDD). It is the procedure of extracting information from big volumes of data, also considered a step in the knowledge discovery process.

1.1 DATA MINING

Data mining is a subfield of computer science that applies statistics and machine learning to reach the collective goal. Data mining discovers hidden patterns and knowledge from large datasets [1]. It helps to analyze and classify data, find the relationship between data and predict results from large datasets. The major techniques in data mining are clustering, classification, association rule learning, anomaly detection and feature selection. The data mining process is described as a three phase process, as shown in Figure 1.1. These phases are mentioned below.

(i) Data Pre-processing
(ii) Data Extraction
(iii) Post-processing

It is observed that real-world data is inconsistent, incomplete and may contain errors. Such concerns are handled through data pre-processing techniques. The first phase of the KDD process converts raw data into target data. Various tasks are performed in this phase, such as reducing the data size, image enhancement, data blending, normalizing the data, and cleaning of data. This phase also determined the features and objects for the second phase of the KDD process. In the second phase, patterns are extracted from the pre-processed data. Several data mining techniques, such as classification, clustering, and association mining, can be considered for pattern extraction.

Further, problem specific technique is also adopted for extracting the patterns from the data set, and the aim is to determine the pattern from the transformed dataset. The third phase corresponds to post-processing. This phase visualizes and validates the discovered patterns. Once the patterns are validated, it turns into information.

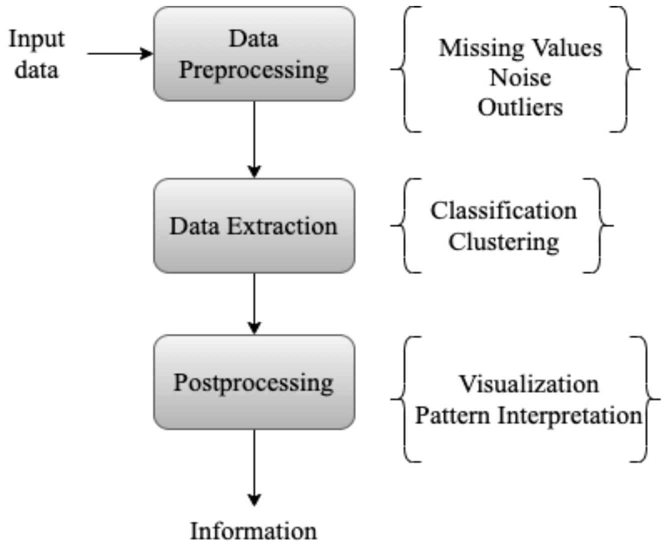


Figure 1.1: Steps in Data Mining

Data mining can also be described as descriptive or predictive [2]. Predictive data mining is supervised learning in which the target variable value is predicted through historical data. In supervised learning, first, a model is developed using the training set (with class labels), and further, it is validated through a test set (without class information). Classification is an example of a predictive data mining task. While unsupervised learning is descriptive data mining in which groups of similar objects are identified from a given dataset. The clustering is an example of a descriptive data mining task.

1.2 DATA CLUSTERING

Clustering is a well-known data mining technique that can be applied in diverse research domains like pattern detection, image segmentation, medical, data mining, etc. The objective of clustering is to determine a set of similar data objects from a given dataset, known as clusters. The data objects within a cluster have more similar features than other data objects. Various distance measures are adopted to determine the group of similar data objects. The complexity of the clustering problem increases when the number of clusters is increased. It became an NP-Hard problem when the number of clusters is more than three [3]. In data clustering, each dataset consists of many data objects and several dimensions represented as $X = x_1, x_2, \ldots\ldots x_n$

and, $X = x_{1,1}, x_{1,2}, \ldots\ldots x_{1,d}$. For i^{th} data object with j^{th} dimension, each data object is represented as $x_{i,j}$. The data is represented as $X_{i,j} = x_{i,1}, x_{i,2} \ldots x_{i,3} .. x_{i,D}$. The aim is to divide the data objects into k cluster centers. The cluster center can be represented as $C = [C_1, C_2 \ldots \ldots \ldots C_k]$. The aim is to divide the data objects into k cluster centers with minimized distance values. The clustering uses different measures such as Euclidean distance, Manhattan distance, Minkowski distance, Hamming distance etc. The most widely adopted is the Euclidean distance [4, 5]. It is used to calculate the minimum intra-cluster distance. The intra-cluster distance can be computed using Equation 1.1.

$$\text{Intra\_Distance}(X_i, C_k) = \sqrt{\sum_{j=1}^{D} \left(X_{ij}, C_{kj}\right)^2} \tag{1.1}$$

Where $\text{Intra\_Distance}(X_i, C_k)$ denotes the distance between i[th] data object and k[th] cluster, X_i and C_k indicate i[th] data object of dataset X and k[th] cluster of C. The clusters consist of the following characteristics:

- At least one data object should be presented in each cluster.
- Each data object belongs to a single cluster.
- Each data object must be allocated to a cluster.

The clustering can be described as hard clustering and soft clustering. In hard clustering, each data object is assigned to the cluster center completely or not. In soft clustering, there is the probability of assigning the data objects to the cluster center. K-means is an example of a hard clustering algorithm, and fuzzy-C means is an example of soft clustering.

1.2.1 CLASSIFICATION OF CLUSTERING ALGORITHMS

The clustering algorithms are classified into the undermentioned key categories [6-9]:

(i) Partition-based
(ii) Hierarchical based
(iii) Density-based
(iv) Grid-based

The partition-based clustering algorithms divide the dataset into k number of partitions using an objective function, whereas k denotes the number of clusters. K-Means is the oldest and most popular partition-based clustering algorithm. Hierarchical clustering algorithms group the data objects into a tree structure, and further, it can be described as (i) agglomerative clustering algorithms and (ii) divisive clustering algorithms [10]. An agglomerative algorithm is based on a bottom-up approach, while a divisive algorithm follows a top-down approach. The popular examples of agglomerative algorithms are single [11] and complete linkage [12]. MONA and DIANA [12] are examples of divisive clustering algorithms. CURE [13] and BIRCH [14] are hierarchical clustering algorithms for handling large-scale data sets. In density-based, clusters are described based on density until the density in the neighbourhood exceeds some threshold. DBSCAN [15], OPTICS [16] and expectation-maximization (EM) [17] are the popular density-based algorithms. Grid-based algorithms divide the object space into a predetermined number of cells for establishing a grid structure. It helps perform the clustering tasks. STING [18] is an example of a grid-based algorithm. However, this research work focuses on the partition-based clustering algorithm.

1.3 METAHEURISTIC ALGORITHM

This thesis work focuses on the capabilities of metaheuristic algorithms for handling clustering problems. The word "metaheuristic" was coined by Fled Grover in 1986 [19]. Metaheuristics are the higher-level heuristics or procedures that incline towards providing a relatively noble solution for an optimization problem. The metaheuristics strategy helps steer the search process towards the global-best solution. Two main tasks of metaheuristics are exploration and exploitation. A proper trade-off between the two is necessary for an efficient search process.

The various features of the metaheuristics are stated as follows [20]:

- Metaheuristic algorithms are problem-independent.
- Discover the search space to determine an optimal solution.
- Different mechanisms for avoiding premature convergence.
- Support to abstraction level.
- Memorize the previous search experience using the concept of memory.
- Adopt heuristics in higher-level strategy.
- Local search is supported through complex learning techniques.
- Having the capability of parallel implementation.

The research community has developed a variety of metaheuristic algorithms to date. Some are inspired by biology, and few are based on physics/chemistry concepts. Most of them are designed based on animal (particles, bats, ants, bees, herds, swarms, and fish) behaviour. These metaheuristic algorithms have been widely adopted for solving single-objective, multiobjective, constrained and unconstrained optimization problems.

1.3.1 SINGLE OBJECTIVE OPTIMIZATION

Single objective optimization consists of a single objective function, and it can be described as either minimization or a maximization problem. Further, for these problems, a single objective function is defined, which is problem-dependent and controlled through a set of variables. A group of constraints are also imposed on single objective optimization problems, called constrained optimization problems. The mathematical formulation of single objective optimization is expressed below.

$$\text{Minimize (z)} \qquad \mathrm{F(z)} \tag{1.2}$$

Subject to

$$g_i(z) \leq 0, \qquad i = 1, \ldots..p$$

$$h_j(xz) = 0, \qquad j = 1, \ldots..r$$

Where equation 1.2, F: $\mathbb{R}^n$ to $\mathbb{R}$ denotes the objective function selected for minimization of the variable n, $g_i(z)$ denotes dissimilarity constraints, and $h_j(z)$ signifies similarity constraints. Further, $p \geq 0$ and $r \geq 0$; if p and r =0, then the problem is an unconstrained optimization problem

1.3.2 MULTIOBJECTIVE OPTIMIZATION

Multiobjective optimization (MOO) problems consider more than one conflicting objective function for solving the problems. In general, two or more objective functions are defined, and it is also noticed that these functions are problem-dependent. Mathematically, the multiobjective optimization problem is described as follows:

$$\text{Minimize/Maximize } \left(f(Z) = [f_1(Z), f_2(Z), \cdots f_n(Z)]\right) \tag{1.3}$$

Subject to $Z \in U$

Where Z denotes the solution, U represents a feasible set, n defines the number of objective functions and $f_n(Z)$ denotes the nth objective function that can be either minimized or maximized. Sometimes, it becomes difficult for any solution (Z) to optimize all objectives simultaneously.

Moreover, it is observed that multi-dimensional space comprises objective functions and decision variables. Every solution () in the decision variable space corresponds to a point in the objective function space. Scalarization and the Pareto method help represent MOO solutions [21]. The Scalarization process integrates the scalar function into the fitness function [22]. In contrast, the Pareto approach is applied if the performance measure and desired solution differ. This approach also generates a set of compromise solutions that can be displayed through the Pareto optimal function. The MOO using the Pareto approach is expressed through Equation 1.4.

$$f_{1,opt} = \min f_1(Z)$$

$$f_{2,opt} = \min f_2(Z)$$

$$f_{3,opt} = \min f_3(z)$$

$$.$$

$$.$$

$$f_{n,opt} = \min f_n(Z) \tag{1.4}$$

Pareto Optimal Solution is the set of all the optimal solutions. During the optimization process, initially, the Pareto approach divides the elements of solution space, then computes the set of dominated and non-dominated solutions using the dominance concept. MOO determines the group of dominated solutions when improvement in one objective function results in degradation of another, called Pareto optimality. At the same time, non-dominated solutions are also obtained when an improvement in one objective function does not degrade another. Such solutions are known as a non-Pareto optimal solutions [23]. It can help to find the Pareto Front (PF) or Pareto optimal solution and can be achieved by balancing all objectives. The Pareto optimal set represents objective function values and their corresponding decision variables. The scope of the Pareto optimal solution is defined through three objective vectors- ideal objective, utopian objective and nadir objective vector.

1.4 MOTIVATION

Clustering has gained wide attention from the research community in the past few decades. It is an unsupervised learning method that contains groups of similar data objects called clusters.

The data objects within a cluster have more similar traits than the data objects in other clusters. The motive of clustering is organizing the data objects into k clusters such that these are optimal in terms of data objects [2-5]. Numerous clustering algorithms have been reported in the literature, and these algorithms are classified as partition-based, density-based, hierarchical-based, and grid-based [13-18]. Further, these clustering algorithms are widely adopted in different research areas such as web analysis [24, 30], medical diagnosis [25], marketing [26], data science [27, 28], education [31], image segmentation [32], bioinformatics [33], text mining [29, 34], financial analysis [35] and business [36].

Researchers have recently focused on metaheuristic algorithms to improve clustering results. Metaheuristic algorithms contain intelligent paradigms for solving optimization problems and, in turn, produce a state of the art results than traditional methods. Some metaheuristic algorithms are genetic algorithms (GA) [37], memory-based grey wolf optimizer [38], ant colony optimization (ACO) [39], particle swarm optimization (PSO) [40], teacher-learning based optimization method (TLBO) [41], artificial bee colony optimization (ABC) [42-43], and cat swarm optimization (CSO) [44]. These algorithms are successfully implemented for solving clustering problems. It is noticed that the literature also highlights several issues related to metaheuristic algorithms. Some metaheuristics like GA and TS are reluctant to find the optimum solution. Sometimes, convergence time is an issue in the case of the ACO algorithm [45]. The problem dimension also impacts the performance of metaheuristic algorithms; for example, the performance of the ABC algorithm is affected due to problem dimension in a few cases [46]. GSA algorithms suffer from premature convergence because of their memory-less nature [47]. Sometimes, single-objective optimization converges on biased solutions [48-49]. Such an issue can be handled through MOO, but it can be suffered with the selection of objective function [50]. Evolutionary algorithms like charged system search (CSS), magnetically charged system search (MCSS), big bang-big crunch (BB-BC), and black hole (BH) are also discussed to solve hard optimization problems [54-57]. These algorithms are also hybridized with other existing algorithms, for example, krill herd algorithm with harmony search (H-KHA) [61], chaotic teaching learning-based optimization (chaotic TLBO) [51], improved cuckoo search and modified PSO with K-Harmonic means (ICMPKHM) [52], PSO based Big-bang-big crunch(PSO-BB-BC)[53], memory-enriched big bang-big crunch (MEBB-BC) [66], cooperative bare bone PSO (CBPSO) [125], improved krill herd (IKH)[131], improved cat swarm optimization (ICSO) [132] and modified butterfly optimization algorithm (MBOA)[138]. The accuracy issue of metaheuristic algorithms is one of the major concerns of the research community, especially with large datasets [139-140]. In turn, metaheuristic algorithms achieve better results and converge on optimal solutions compared to traditional methods.

1.5 OBJECTIVE

The various clustering algorithms have been presented in the literature. However, it is observed that metaheuristic algorithms provide improved results compared to traditional algorithms. Although, there exist some issues related to metaheuristic algorithms, for instance, trapped in local optima, convergence, quality solutions and partial solutions. These issues have also affected the outcome of metaheuristic algorithms. Based on the motivation and limitations of clustering algorithms presented in the previous section, the main objectives of this research work are highlighted as

1. Design a metaheuristic algorithm to guide the global search in the optimal direction and handle premature convergence issues.
2. Design a neighbourhood search-based improved metaheuristic algorithm for addressing the local optima issue.
3. Design a multiobjective metaheuristic algorithm to improve the clustering efficiency and investigate the biasing issue of single-objective clustering.

1.6 ORGANIZATION OF THESIS

The entire thesis is organized into six chapters. The description of these chapters are given as follows:

Chapter 2 reviews the related works in the field of data clustering. It includes single-objective and multiobjective clustering algorithms for handling data clustering problems. This chapter discusses different metaheuristic algorithms and investigates their performance in solving data clustering problems.

Chapter 3 discusses an improved water wave optimization (IWWO) clustering algorithm for obtaining good clustering results. Before implementing the IWWO, several modifications are incorporated into the WWO algorithm. These modifications aim to guide solutions in the global search direction and handle premature convergence issues.

Chapter 4 presents an improved bat algorithm for effective cluster analysis. Initially, this chapter addresses the issues related to the bat algorithm, such as population initialization, convergence rate and local optima. These issues are managed through an enhanced cooperative coevolution strategy, an elitist strategy and a neighbourhood-search scheme. The simulation results are discussed using various performance metrics and compared to standard and hybrid metaheuristic clustering algorithms.

Chapter 5 discusses a multiobjective vibrating particle system (MOVPS) for effective partitional clustering. This chapter investigates multiobjective optimization's capabilities and addresses the biasing issue of single-objective clustering. Further, two objective functions, namely, intra-cluster distance and connectedness, are considered for implementing the MOVPS algorithm.

Chapter 6 highlights the outcome of this thesis and the future scope of this research work.

CHAPTER 2

LITERATURE SURVEY

This chapter presents the related studies of data clustering, emphasis on metaheuristic algorithms. It includes single-objective and multiobjective data clustering algorithms.

2.1 SINGLE OBJECTIVE CLUSTERING ALGORITHM

The flower Pollination algorithm (FPA) has a strong exploitation ability but a weak exploration ability. Moreover, it gets trapped in local optima. To handle these issues of FPA, Wang et al. [58] hybridized FPA with bee pollinators (BPA). The aim is to enhance global and local search ability. The proposed algorithm utilizes the discard solution operator from the artificial bee colony algorithm as the discard pollen operator to enhance the global search ability of FPA. It works on discarding the solutions if they are not further improved to find the best global solution. Honey-bees utilize levy flight during global search procedure. The proposed hybridized algorithm has been evaluated on several standard and artificial datasets. The simulation results have been compared to ABC, DE, FPA, CS, PSO clustering algorithms. The optimal results in terms of higher accuracy and stability have been provided by the proposed hybridized algorithm. The convergence rate is also enhanced.

A Heuristic Kalman filtering algorithm (HKA) is applied on data clustering to improve the accuracy and efficiency of HKA [59]. The Kalman-filtering algorithm is hybridized with K-mean. The centroid updating step from K-means is adopted for faster convergence and better accuracy. The algorithm also considers a conditional restart mechanism. It is applied when the solutions are limited to a small region. It helps in avoiding the local optima situation. The performance of the proposed algorithm was evaluated on seven benchmark datasets- artset1, artset2, iris, wine, CMC, glass, and cancer. The results have been compared to several other metaheuristic algorithms based on ARI, Davies-Bouldin index, Intra and computation time as performance metrics. From experimentation results, it is noticed that the Kalman-filtering algorithm provides significantly better results and successfully handles the shortcoming of KM. Future works consider the hybridization of HKA with other methods.

A hybridized algorithm K-MWO was proposed by Kang et al. [60]. The authors hybridized K-Means with the mussel wandering optimization algorithm. The proposed algorithm adopted the local search capabilities of K-Means. The algorithm also benefits from mussel wandering optimization for enhancing global search ability. Similarity-based clustering ensemble method is adopted. The idea is to change the weights for data objects in each iteration. The weight information has been added in the objective function of the clustering algorithm. Weights are increased for the data if it is classified under different classes in different clustering processes. The proposed K-MWO algorithm has been tested upon nine datasets iris, wine, glass, cancer, banknote authentication, image segment, student evaluation, Landsat satellite, and Pen-based digits. The various performance measures used during evaluation are Davies–Bouldin Index (DBI), Dunn Index (DI), Calinski–Harabasz (CH), G-mean, F-measure, and Precision. The simulation results have been compared to K-PSO and KM algorithms. The results reveal that K-MWO is an effective approach for cluster analysis. New rules for updating the weight values can be investigated in future. The reasons can be explored for the datasets where the proposed algorithm does not perform better.

Krill herd (KH) gets trapped in local optima and suffers from premature convergence. To solve the limitations of KH, Abualigah et al. [61] proposed a hybrid algorithm (H-KHA). The proposed algorithm combined KH with harmony search (HS). The proposed H-KHA algorithm works in three stages: (i) motion calculation, (ii) genetic operators, and (iii) improvising a new solution. From HS, a distance factor is integrated into KH to improve the global search mechanism. It is computed through the distance between each location of the data object to the best fitness-value. The performance of H-KHA has been evaluated using performance measures accuracy and convergence rate. The datasets used in the experimentation were iris, wine, glass, seeds, vowel, cancer, CMC. The six text clustering datasets were also used in the experiment. The results have been compared to various standard and optimization clustering algorithms. From the results, it is observed that H-KHA gives the highest rank in statistical analysis using F-measure when compared to other algorithms. The KH algorithm can be hybridized with other local search procedures in future. The proposed algorithm can be explored on benchmark function datasets.

K-means algorithm is dependent on the initial solution and gets stuck in local optima. Zhou et al. [62] inspected these issues and proposed a symbiotic organism search (SOS) algorithm for data clustering. The equations of mutualism and commensalism phases have been modified and the Parasite vector has been adopted in the parasitism phase. The optimal cluster center of each spider individual has been achieved in the SMSSO algorithm. The proposed algorithm has been evaluated on ten datasets, namely, artificial set1, artificial set2, Iris, Seeds, Teaching Assistant Evaluation (TAE), Haberman's Survival, Statlog (Heart), Balance, Contraceptive Method

Choice (CMC), Wisconsin breast cancer, and Wine. The results have been compared to several other clustering algorithms – ABC, SSO, FPA, CLSPSO, DE, and K-means. It is found from the results that the proposed SOS algorithm produces more consistent clustering results. Future works consider optimizing the intra-cluster distance, automatically determining the number of clusters, and investigating high-dimension problems.

Nayak et al. [63] considered the sensitivity to the initial selection of cluster centers, premature convergence, and local optima. The authors proposed a hybrid technique for effective cluster analysis. Fuzzy-C means have been employed in chemical reaction-based metaheuristics. The objective is to best cluster centers with improved inter and intra-cluster distance. The chemical reaction-based optimization selects the initial cluster centers for FCM. This helps in improving the efficiency of the FCM algorithm. The proposed hybrid algorithm has been tested over thirteen datasets- iris, lenses, haberman, balance, WBC, CMC, hayesroth, robot navigation, heart, wine, glass, lung cancer and artificial dataset. The simulation results have been compared to several other clustering algorithms based on the inter and intra-cluster distance, fitness metric, number of iterations and error rate. The simulation results show that the proposed algorithm proves to be effective for cluster analysis. The proposed algorithm can be investigated over complex data using neural networks, and examining it for solving other datamining problems. Real-life problems like the agriculture sector, medical etc., can also be explored.

Han et al. [64] investigated the exploration and exploitation mechanism. The modified gravitational search algorithm is proposed. The authors' incorporated bird flock behaviour into GSA. They have enhanced diversity through three steps: initialization, identification of nearest neighbour and orientation change. The candidate solutions are generated in the initialization to be passed in the second step. The position of the data object is updated based on nearest neighbours. The collective response helps explore the global search space and thus avoids entrapment in local optima. Further, the performance evaluation has been done using thirteen benchmark clustering datasets. The simulation results have been compared to FA, GSA, ABC, PSO, K-Means, K-PSO, NM-PSO, CPSO and K-NM-PSO. It is observed that the proposed algorithm provides significant clustering results in terms of error rate and intra-cluster distance.

Based on the continuous greedy randomized adaptive search procedure (C-GRASP) approach, Queiroga et al. [65] developed the C-GRASP-Clu algorithm. The aim was to resolve issues of local optima issue and slow convergence. The C-GRASP-Clu algorithm automatically adjusted the step size during the search procedure. The minimum step size value is used as a stopping criterion and is reduced with each iteration. So, a filtering mechanism has been applied to avoid unpromising iterations. Nineteen datasets, including benchmark and artificial datasets, have been used for experimentation. The experiment results have been compared to various clustering algorithms such as K-means, PSO, TS, GA, and C-GRASP. The proposed C-GRASP-Clu algorithm gives stable clustering results. It can be investigated on different clustering problems in future. Also, other local-search procedures and hybridization methods can be adopted for further improvements.

To improve the exploration and exploitation, Bijari et al. [66] presented a clustering algorithm using a memory-enriched method in the Big bang-big crunch, named memory-enriched Big bang-big crunch (ME-BB-BC). In this study, memory has been used for storing previous and newly generated solutions. The centers of masses found in the earlier iterations are stored in the limited-size memory. This helps in enhancing the algorithm's exploitation capability. After the Big-bang Phase, K-means was implemented for generating improved solutions and proposed kMEBB. The proposed ME-BB-BC algorithm has been assessed using seven benchmark mathematical functions and six clustering datasets. The results have been compared with GA, BB-BC, PSO, and GWO. Also, statistical analysis is done through the t-test and Friedman test. The results revealed that the proposed ME-BB-BC gives better results for data clustering compared to other metaheuristics. In future, the proposed algorithm can be applied to multiobjective optimization and used as a parameter-setting method in applications like power dispatch systems.

The issues of lack of diversity in population, local optima, convergence rate and a tradeoff between exploration and exploitation in the cat swarm optimization (CSO) algorithm were handled by Kumar and Singh [67]. An improved CSO algorithm was proposed. In the tracing mode of CSO, a new enhanced velocity equation is devised. Also, an updated equation is given for position updating in tracing mode and seeking mode. Additionally, a local search method is applied to resolve local optima issues. The proposed ICSO is validated through twelve benchmark mathematical functions. The authors have also discussed the application of the proposed algorithm in data clustering. The performance evaluation as a clustering algorithm is done on the wine, iris, CMC, cancer, and glass datasets. The simulation results have been compared with other clustering algorithms such as CSO, K-means, PSO, GA, ACO, and TLBO.

The improved CSO yields better-quality clustering results in comparison to the different algorithms.

Das et al. [68] inspected the issues of local optima and convergence rate in clustering algorithms and developed a new clustering algorithm based on class topper optimization (CTO). This algorithm is inspired by students' intellectual performance in a particular school class. CTO works in three levels: class, section, and student. The proposed CTO algorithm uses the class topper and section topper information. The section toppers are the best-learned students of the different departments, and these students compete with each other for the class topper. The section topper also learns from the class topper. In turn, significant improvement was recorded in the performance of the section topper. The students' performance were also evaluated through examination. The proposed algorithm has been tested on five datasets (iris, wine, cancer, cmc, HV). The results have been compared with different metaheuristics and heuristics algorithms. The simulation results show that the proposed algorithm gives faster convergence and is not stuck in local optima. It is seen that CTO gives better results in terms of average percentage error. The proposed algorithm failed to give quality results for non-spherical data, which can be explored in the future. The CTO can also be applied to real-world problems.

Deb at al. [69] presented a new metaheuristic C-ESA by integrating the elephant search algorithm (ESA) into K-means. The proposed algorithm C-ESA benefits from the evolutionary operations and balance between local and global search. Determining the best centroid locations and accuracy improvements are the motives of the proposed C-ESA. The performance evaluation of the proposed C-ESA is examined via four benchmark clustering datasets- Mice, Gesture, Haberman, Iris, and ten time-series clustering datasets. The simulation results show that C-ESA gives quality results compared to other meta-heuristic algorithms for clustering based on accuracy. The C-ESA can be improved with modification for clustering. Several issues, like computational time, code tuning, and parameter-free mechanism, can be investigated.

Tsai et al. [70] presented coral-reef optimization with substrate-layers (CRO-SL) for clustering large data. The clustering results are refined by using the substrate layers concept. It integrates PSO and genetic K-means (GKA) algorithm in the substrate layer. These algorithms replace the HS, 1-point crossover, DE, and Gaussian mutation. The local-search methods have been applied to refine the results. The CRO-SL is used on the cloud platform to reduce response time for data analysis, and its performance was tested over seven benchmark datasets and two artificial clustering datasets. The results have been compared with clustering algorithms such as K-means, PSO, GKA, PSO and simple coral reef optimization (SCRO). The results show that CRO-SL has helped to improve the clustering results and efficiency. The future works consider the dynamic adjustment of parameter settings, the use of a graphics processing unit for speeding the running time and its implementation to other real-world problems.

To handle the imbalance between exploration and exploitation, Sharma and Chhabra [71] proposed a new clustering algorithm inspired by PSO and polygamous crossover called PSOPC. A polygamous-based selection crossover operator is added in PSO to achieve stability in exploration and exploitation. Also, inertia weight and alpha values have been added to enhance the exploration behaviour. The performance of the proposed PSOPC algorithm is assessed over seven benchmark clustering datasets in terms of convergence rate, cluster quality and cluster distance. The various other metrics used for evaluation are precision, accuracy, sensitivity, and g-measure. The simulation results were compared to several different clustering algorithms. It shows that the PSOPC algorithm gives better-quality results. The future directions are to propose a multiobjective version of the algorithm, investigate using dynamic data and extend it for automatic clustering for applications such as protein synthesis, image segmentation etc.

The issues of exploration, local optima, convergence rate and finding the best solutions were examined by Abdulwahab et al. [72]. The authors proposed a Levy Flight Black Hole (LBH) clustering algorithm. The proposed algorithm was integrated with the levy flight and blackhole optimization algorithm. The black hole algorithm cannot explore search space ahead of the current black hole. So, to handle this issue, the levy flight concept has been introduced. With this, the step size is increased for the movement of stars and helps to explore large search space. Thus, the global search ability of BH is improved and prevents it from getting stuck in local optima. Six benchmark datasets have been used for the performance evaluation of LBH and compared to several existing clustering algorithms such as ACO, PSO, GWO, GSA, BH, Cat Swarm algorithm and three others. It is seen from the results that LBH gives robust results for clustering. It can be applied to text clustering in future.

Mustafa et al. [73] proposed an adaptive memetic differential evolution (ADME) optimization algorithm for data clustering to address the issues of exploration and exploitation process in clustering algorithms. The proposed algorithm benefits from the memetic and adaptive differential evolution (DE) algorithms. A mutation strategy was applied to balance the local and global search. The restart mechanism generated new solutions to achieve population diversity and thus prevented the algorithm from premature convergence. It also used an approach for clearing the duplicate solutions and adopted a hill-climbing local search procedure. The proposed ADME algorithm was tested on six datasets, and the results were compared to ME and DE algorithms. From simulation results, it is noticed that ADME gives better clustering results compared to other algorithms. Using different objective functions for categorical and mixed data and extending it as a multiobjective approach are the future perspectives.

Adaptive differential evolution with neighbourhood search (ADENS) was proposed by Tarkhaneh and Moser [74]. Mantegna Levy distribution, Archimedean spiral and neighbourhood search were integrated into improved differential evolution (IDE) algorithm. The population diversity was also enhanced using a new mutation strategy (DE/Spiral-to-rand/1). The weak solutions were replaced by new solutions through a neighbourhood strategy. It helped to avoid the local optima condition. The initial solutions were also generated through Cauchy's distribution. The six datasets were utilized to examine the ADENS performance. The ADENS algorithm gives superior results compared to other clustering algorithms in terms of accuracy and intra-cluster distance. The statistical analysis was also done using Wilcoxon and Friedman. The results show that ADENS can be adopted for data clustering. In future, it can be improved using chaotic or quantum theory. The work can explore the different real-world applications, and a suitable guided method can be employed for exploration.

The clustering algorithms suffer from the problem of a random selection of initial cluster centers and premature convergence. In context to issues mentioned earlier, Agbaje et al. [75] proposed a hybrid algorithm using the firefly (FA) algorithm and PSO, called FAPSO. FA was implemented for the initial search because of its strong exploitation ability. PSO was adopted for finding optimal global solutions in exploration. The inertia-weight, acceleration coefficients and velocity parameters of PSO help to balance the exploration and exploitation. The performance of FAPSO has been evaluated over twelve standard datasets, and results have been compared to traditional clustering algorithms. The results show that FAPSO is advantageous to other clustering algorithms in the context of DB and CS index. The proposed algorithm can be enhanced further using Levy flight to reduce the foraging time. It can also be applied to solve other complex problems and use different local search methods.

Kushwaha et al. [76] proposed an electromagnetic clustering algorithm (ELMC) to address the choice of initial cluster issue of the k-means algorithm. The proposed algorithm is an improved version of electromagnetic field optimization. It used the attraction-repulsion strategy to maintain the diversity of ELMC. The authors conducted a series of experiment for evaluating the effectiveness of the proposed ELMC algorithm using two IOT datasets and other standard clustering datasets. The fitness of the electromagnetic particle was evaluated using intra-cluster distance. The experiment results were compared to different clustering algorithms, like KFCM, ACO, KFC, PCM and K-means. The proposed ELMC provides better clustering results in contrast to algorithms in comparison based on purity, rand index and normalized mutual information. The work can be extended for text clustering and image segmentation.

Senthilnath et al. [77] applied a flower pollination algorithm (FPA) to address the data clustering problems. The objective of the proposed FPA is to compute optimal cluster centres. The global search for biotic and cross-pollination was achieved through levy flight. The local search was performed through self-pollination and abiotic. Population diversity was maintained through reproduction probability. Also, the balance between local and global search was handled through the switching probability strategy. Three clustering datasets were used for performance evaluation and compared with eight clustering algorithms. From experimentation results, it is noticed that FPA gives minimum classification error in context to other algorithms in comparison. The statistical analysis shows that FPA can be adopted for data clustering. The proposed algorithm can be hybridized to solve different problems.

Further, to improve the efficacy of data clustering and avoid local optima issues, Mangeshkumar et al. [78] proposed a hybrid algorithm using ant lion optimization (ALO) with local search algorithms and ant colony optimization (ACO), named ACO-ALO. The initial random solutions were generated using ACO. The number of populations for ants was taken the same as ACO and ALO. The pheromone trails help to reach the optimal global solution. The authors also applied Cauchy's mutation operator to prevent local optima issues. The performance evaluation of the proposed hybrid ACO-ALO algorithm has been done on four datasets and compared to ACO and K-means algorithms. The results show the superiority of the clustering results given by the ACO-ALO algorithm over ACO and K-means. Future directives show that neural networks can be used in the proposed ACO-ALO to make it more independent in setting the parameter values.

Further, Xie at al. [79] examined the sensitive to initial clusters and trapping in local optima issues. The authors provided two variants of FPA called inward intensified exploration FPA (IIEFPA), and compound intensified exploration FPA (CIEFPA) to address the abovementioned problems. The IIEFPA used the randomized control matrix to enhance the exploitation ability in neighbourhood search and generating diverse solutions. CIEFPA utilizes dispersing mechanism. More similar fireflies are moved to new locations in the neighbourhood to explore the search space. The effectiveness of proposed variants of FPA has been assessed on fifteen different datasets using different performance metrics such as accuracy, intra-cluster distance, specificity, sensitivity, and f-score. The experiment results indicated that FPA variants give superior clustering results in terms of minimum distance and provide higher accuracy than other algorithms. The proposed variants will be evaluated for different jobs like image segmentation, feature selection etc., and other objective functions along with intra and inter-cluster distance will also be investigated.

For efficient search and fast convergence, Huang et al. [80] proposed memetic particle gravitation optimization (MPGO) algorithm using a memetic clustering algorithm based on PSO and GSA. The proposed MPGO works on two hybrid operation mechanisms and enhanced diversity mechanisms. After a predefined number of function evaluations, there is an exchange of individuals from two subpopulations in a hybrid operation. The diversity enhancement mechanism uses an enhancement operator like the crossover process of DE. The selection mechanism of individuals uses the roulette-wheel method. Six benchmark clustering datasets, along with fifty-two benchmark functions and images, were used for the performance evaluation of MPGO. The experimental results were compared to PSO, BH, GSA, K-means and WOA algorithms. The MPGO gives significantly better results in terms of fitness function and clustering accuracy rate. Future work can be extended for dimension reduction, image enhancement, diversity enhancement using levy flight, and application to an automatic object tracking system.

For addressing local optima and slow convergence issues of the clustering algorithm, Dinkar and Deep [81] presented an improved ant lion optimization (ALO) algorithm called OB-C-ALO, and two amendments were proposed. Initially, the algorithm generated the solution from the uniform distribution. The first amendment integrated the Cauchy distribution in ALO to handle the local optima problem and enhance exploration. The solutions were updated in this phase. The second amendment utilized opposition-based learning to control slow convergence. An acceleration-coefficient parameter was used for balancing exploration and exploitation. When the number of iterations increased, this parameter decreased the exploitation convergence. The performance evaluation was evaluated on six datasets and twenty-one benchmark functions. The results were compared to AO and C-ALO in terms of standard deviation, average, maximum and minimum fitness values. The results given by the proposed OB-C-ALO algorithm are significantly better in terms of distance.

The critical distance clustering algorithm was designed by Kuwil et al. [82] based on a distance-based clustering algorithm. The critical distance was computed using Euclidean distance, and statistic operations were derived to find the similarity between data. The proposed algorithm works on quantitative data only. A study on very weak, weak, good, very good and excellent students was considered. The proposed algorithm helped in merging the very weak and weak students based on the grades in one cluster. Very good and excellent students were in another cluster, and a separate cluster was formed for the good students. The proposed algorithm was assessed through twenty-six experiments, and results were compared to DBSCAN, K-means and MST-based clustering. The outliers have been handled through this approach successfully. A new feature selection will be proposed in future. For improving the speed of the proposed algorithm, the distance selected should be considered less than or equal to lambda in the enhancement.

To address issues of slow convergence and local optima, Singh et al. [83] proposed artificial chemical reaction optimization (ACRO) for data clustering. The algorithm is constructed on chemical reactions and uses reactants as its population. The aim is to find the optimal cluster centres with minimized intra-cluster distance. To improve the convergence rate, a position-based strategy was applied. This strategy was incorporated in the synthesis-reaction step. Further, the local optima problem was addressed using neighbourhood strategy. The neighbourhood strategy was used to balance the trade-off between different reactants for inter-reactant collision. Five benchmark datasets and two artificial datasets were used for the evaluation of the ACRO algorithm and compared to several other algorithms using performance measures average case, best case, standard deviation, intra-cluster distance and f-measure. The statistical analysis was validated using Friedman statistical test. The results showed the effectiveness of the proposed ACRO algorithm for data clustering.

An efficient stud krill herd clustering (ESKH-C) algorithm was proposed by Baalamurugan and Bhanu [84]. It was used to compute the optimum locations of cluster centers. The stud selection and crossover (SSC) operator were integrated through the krill herd clustering algorithm. The SSC operator helps in selecting the individuals and taking them forward to the next level. This operator was inspired by the genetic reproduction process that helps in improving the convergence rate. The proposed algorithm is also able to search the global search space. Seven experiments were conducted to measure the efficiency of the proposed ESKH-C algorithm.

Rand index, Jaccard coefficient, Beta and Distance index were used for correlation analysis with other clustering algorithms. It is seen that ESKH-C can work with densities, cluster numbers, and multi-dimensional datasets. The proposed algorithm can be extended for web applications to discover data cluster groups. It can also be used in the biomedical field for DNA sequencing clustering problems.

For effective data clustering, Singh [85] proposed chaotic harris hawk's optimization (CHHO). The HHO algorithm is based on the cooperative and hunting style of Harris' Hawks. Harris' hawks represent the candidate solutions. The best solution in each step was selected as the intended prey that is nearly the optimum solution. Further, the escaping energy of the prey helped to decide whether candidate solutions will exploit or explore the search domain. The author used a chaotic sequence number instead of random numbers. It was employed for guiding the global search. The proposed HHO algorithm was assessed over eight shape datasets and four other standard datasets. Its performance was compared with several clustering algorithms such as GWO, HHO, BOA and others, using different performance measures and statistical analysis. The results showed that HHO algorithm proves to be an effective approach for data clustering. The proposed CHHO algorithm can be investigated for other real-world problems and can be extended as a multiobjective approach.

The grey wolf optimizer (GWO) suffers from the issues such as trapping in local optima, and premature convergence. To address these issues, Alijarah et al. [86] incorporated the tabu search (TS) method in GWO, named TSGWO. The proposed algorithm utilized adaptive memory to discover neighbourhood leaders prior to update the locations of the wolves. TS was used to update the leader wolves' positions. TS-based search operator helped find the leader's first location during local exploratory search and then searching appealing regions in the neighborhood space of the first received location. The tabu list keeps a record of the previously visited sites and helps GWOTS to get stuck in local optima and reach the global optimal solution.

Thirteen datasets have been used to evaluate TSGWO and compared to the results of several clustering algorithms based on SSE, purity, and entropy performance measures. From the results, it is noticed that TSGWO effectively overcomes local optima issues and gives a better convergence rate when compared to other clustering algorithms. In future, the GWOTS will be investigated over different datasets of arbitrary shapes. It can be implemented in spatial applications, and the extension of GWOTS via parallel computing can help reduce computation time.

Similarly, to address the issues of being stuck in local minima and the accuracy of the bat algorithm, Zhu et al. [87] proposed an improved bat algorithm for data clustering. The authors incorporated two improvements for improving local and global search. Gaussian-based convergence factor and five other convergence factors were used for improving global search. It used the convergence factor that decreases with the increase in the number of iterations. Initially, the convergence factor with lower degree attenuation moves to a larger amplitude for finding the global optimal solution. During the last iterations, with an increase in attenuation of D, there is an increase in the span of movement. This helped in finding the optimal solution more accurately and balancing the exploitation and exploration abilities. Six convergence factors were proposed with the use of cosine, tangent, sine, exponential, and power function.

Further, the algorithm utilizes the reducing orbiting mechanism based on whale-based optimization and the position spiral updating mechanism based on sine and cosine algorithms for enhancing the local search. The performance of improved algorithm was assessed on seven datasets, and results were compared to various other clustering algorithms based on accuracy, ARI, and f-measure. The experimentation results showed that the improved bat algorithm gives significant consequences for accuracy. The proposed algorithm can be further enhanced to make it more stable in searching the solution.

The issues of slow convergence and local optima were investigated by Kaur et al. [88]. The authors developed a new clustering algorithm using chaos optimization and flower pollination algorithm, named chaotic FPA (CFPA). This work also presented the comparison of standard FPA and chaotic variants of FPA, with sine, Chebyshev, dyadic, and circle maps. The best chaotic map was used in further investigation of the study. The performance of the proposed algorithm was evaluated using sixteen clustering datasets. The experiment results were compared to other clustering algorithms in terms of execution time, cluster integrity, and iteration needed for convergence. From the experiment results, it is noticed that the proposed CFPA algorithm gives stable results. In future, the new nature-inspired approaches can be considered for partitional clustering. The constraints handling problems can also be considered, and the algorithm can be improved to provide more stable results.

Table 2.1: Related works on single objective clustering algorithms

Author Name [ref], year	Issues	Approach	Data Sets
Wang et al. [58], 2016	-Sensitive to initial selection of cluster centers -Local Optima -Population Diversity	-FPA using Bee Pollinator was presented. -Discard-pollen-operator applied along with crossover operator for diversity enhancement in population. -Local search enhanced through an elite-based-mutation operator.	-Iris, Balance, WBC, Seeds, CMC, Wine, Statlog (Heart), Haberman's Survival, artificial set1 and artificial set2.
Pakrashi & Chaudhuri [59], 2016	-Exploration ability -Premature Convergence	-Heuristic Kalman Algorithm (HKA) was introduced. It uses the Kalman-filtering approach. -Improved hybrid approach is given using HKA and K-means. - Proposed approach benefits from fast convergence of K-Means and global-exploration of HKA.	-Wine, Iris, Glass, CMC, Cancer, artset1, artset2
Kang et al. [60], 2016	-Balance between local and global search	- Proposed K-MWO. It uses KM with mussels-wandering-optimization (MWO). -Local search is improved thru KM. Global search is enhanced thru MWO.	-Banknote authentication, Iris, Breast cancer, Wine, Glass, image segment, Landsat satellite, Student evaluation, Pen-based digits
Abualigah [61], (2017)	-Exploration ability -Premature convergence	-Proposed H-KHA that combines KH and harmony search (HS). -Distance factor from HS enhances global search ability in KH.	-CMC, Vowel, Cancer, Glass, Wine, Iris, Seeds, Classic4, Reuters21578, 20Newsgroup
Zhou et al. [62], 2017	-Sensitive to initial choice of cluster-centers -Local Optima issue -Convergence-rate	-Presented SMSSO algorithm. -Stochastic variant strategy of the original SSO is replaced by through the simplex method. -New equations have been given for the mutualism and commensalism phases. -Adopted parasite vector in the parasitism phase.	-Iris, Balance, Wine, Statlog, Teaching Assistant Evaluation, WBC, Seeds, Haberman's Survival, CMC, artificial set1 & artificial set2.

Nayak et al. [63], 2017	-Sensitive to initial selection of cluster centers -Local Optima -Convergence rate	-Proposed a novel hybrid approach of Fuzzy C-Means with a chemical reaction-based metaheuristic to obtain optimal cluster centers.	-Iris, Balance, Spect heart, Lenses, CMC, Haberman, Hayesroth, WBC, Wine, Robot navigation, Glass, Artificial dataset, Lung cancer
Han et al. [64], 2017	-Enhancing exploration and exploitation	-Introduced bird flock behaviour into GSA. -Diversity enhancement did thru initialization, identifying nearest-neighbors, and orientation-change.	-Balance, E. Coli, Heart, Glass, Thyroid, Cancer, Cancer-Int, Dermatology, Credit, Glass, Diabetes, Wine, Horse, Iris
Queiroga et al. [65], 2018	-Local optima -Slow Convergence	-Proposed C-GRASP-Clu. The basis of the algorithm is a continuous greedy randomized-adaptive-search procedure (C-GRASP). -Automatically adjusted the step size in the search procedure -Applied a filtering mechanism to avoid unpromising iterations.	-Iris, WDBC, Glass, Vowel, Wine, Cancer, CMC, Ionosphere, Yeast, Crude oil, Abalone, a1, D31, s1, Unbalance, artset1, artset2, artset3, artset4
Bijari et al. [66], 2018	-Local optima -Slow Convergence -Imbalance between exploration and exploitation	-Proposed memory-enriched approach. -It is based on the big-bang–the big crunch algorithm.	-Iris, Wine, Glass, Contraceptive method Choice, cancer, Vowel
Kumar & Singh [67], 2018	-Lack-of-diversity in the population -Local Optima issue -Convergence rate -Unbalance between exploration and exploitation	-Proposed an improved CSO. -Incorporated new search equation. -Employed Local search method to obtain eminence solutions. It helps prevent local optima conditions.	-Iris, Wine, CMC, Cancer, Glass
Das et al. [68], 2018	-Local Optima -Convergence rate	-Proposed a new class toper optimization (CTO) algorithm. -It is based on the intelligence behaviour of students in a particular class of a school.	-Iris, Wine, Cancer, CMC, Hill Valley (HV)
Deb et al. [69], 2018	-Initial cluster selection -Local Optima	- Presented C-ESA. -Integrated Elephant search algorithm into K-means. -The algorithms adopted evolutionary operations to balance local and global search.	-Gesture, Mice Protein, Haberman, Iris

Tsai et al. [70] (2019)	-Managing large data -Advanced clustering result	-Developed Coral reef optimization with substrate layers. -Integrated substrate-layers into PSO & genetic-k-means algorithm (GKA).	-Iris, User Locations, Wine, BCW, HTRU2, Abalone Spam base, Finland, c20d6n2000, & c20d6n200000
Sharma & Chhabra [71] (2019)	-Local Optima issue -Convergence rate -Exploration and exploitation tradeoff	-Proposed PSOPC, a hybrid approach. -PSO is applied for global search. The polygamous approach used in crossover to balance tradeoff exploration and exploitation. -Dynamically tuned the parameters to refine the optimization process.	-Wine, CMC, Bhupa Cancer, Glass, Haberman, Iris
Abdulwahab et al. [72] (2019)	-Convergence rate -Local Optima issue	-Presented Levy Flight Black Hole for data clustering. -Levy flight is combined to the Black hole algorithm. -Step size generated by Levy distribution controls the movement of each star.	-Iris, Vowel, CMC, Glass, Wine, Cancer
Mustafa et al. [73] (2019)	-Exploration and exploitation tradeoff	-Introduced an adaptive memetic differential-evolution optimization-algorithm. -Advanced mutation strategy has been used in adaptive DE and memetic algorithms.	-WBC, Glass, CMC, Iris, Vowel, Wine
Tarkhaneh & Moser [74] (2019)	-Convergence rate -Exploration and exploitation tradeoff	-Presented Adaptive Differential Evolution with Neighborhood Search approach. -Robust solutions are generated by using a new mutation strategy formed by combining Archimedian Spiral with Mantegna Levy flight. -Self-adaptive strategy is applied for tuning control parameters	-Iris, CMC, WBC, Yeast, Heart, Vehicle, Diabetes, Wine, Letters, Liver, Ionosphere, and Cars
Agbaje et al. [75] (2019)	-Premature convergence -Imbalance between local and global search	-Proposed hybrid algorithm using firefly algorithm (FA) and PSO algorithm, named FAPSO. -Initial search performed using FA. - Optimal solution is reached by applying PSO.	-Breast, Yeast, Path-based, Compound, Flame, Spiral, Glass, Iris, Two moons, Jain, Statlog, Thyroid, Wine

Kushwaha et al. [76] (2019)	-Choice of initial clusters -Issue of Local Optima	-Proposed an enhanced variant of electromagnetic field optimization. -Attraction & repulsion concept helps in improving diversity in the population of the EFO algorithm.	-Iris, GAS, CMC, Vowel, Crude oil, Thyroid, Ionosphere, Human Activity Recognition
Senthilnath et al. [77] (2019)	-Prior info required for the number of clusters	-Developed FPA for data clustering. -Minimized objective function to achieve the optimal position of cluster centers.	-Image segmentation, glass, vehicle, Crop Type & Synthetic datasets
Mangeshkumar et al. [78] (2019)	-Exploration and exploitation tradeoff -Entrapping in local minima -Minimum intra-cluster distance	-Presented hybrid ACO-ALO algorithm. -Local optima situation avoided using Cauchy's mutation operator	-Zoo, iris, glass, wine
Xie et al. [79] (2019)	-Sensitive to initial selection -Entrapment in Local optima	-Introduced two-variants of FA IIEFA and CIEFA. -Exploration and exploitation enhancement did using Matrix-based search. -Feature reduction handled through minimum Redundancy Maximum Relevance based feature selection method.	-ALL IDB2 database, , Sonar, Thyroid, Iris, Ozone, WBC1, Wine, WBC2, Balance, E. coli & a skin lesion data set
Huang et al. [80], 2019	-Clustering accuracy -Convergence rate	-Proposed memetic particle gravitation optimization. -PSO is employed for the exchange of individuals. -GSA is applied as an enhancement operator to improve the diversity of the population.	-Wine Statlog, Yeast, Iris, Breast cancer, and Car evaluation, six images in image segmentation &52 benchmark function.
Dinkar & Deep [81], (2019)	-Local optima -Slow convergence	-Introduced opposition-based ALO thru Cauchy distribution. -Local optima is handled thru random walk based on Cauchy distribution. -Utilized opposition-based learning model and acceleration coefficient.	- Glass, Iris, CMC, Wine, LD, WBC & 21 benchmark test-functions

Kuwil et al. [82], 2019	-Improving accuracy -Optimized Cluster numbers	-Proposed Critical distance clustering algorithm. -Employed Euclidean distance basic statistic operations	-Conducted 26 experiments on real and synthetic datasets
Singh et al. [83], 2019	-Slow Convergence -Local Optima	-Proposed an artificial chemical reaction optimization (ACRO) -Convergence rate improved through the location-based method. -Neighborhood strategy applied for avoiding local optima	-CMC, Wine, Glass, Iris, artset1, artset2
Baalamurugan & Bhanu [84], 2019	-Convergence rate -Efficiency	-Developed efficient stud krill herd clustering (ESKH-C) algorithm. -Integrated Stud selection and crossover (SSC) operator in krill herd clustering algorithm.	-2D–4C, 10D–4C, Iris, Wine, Glass, Zoo, &Ionosphere
Singh [85], (2020)	-Manage large data -Improving the performance -Dependency on random numbers	-Harris hawk's optimization algorithm is presented. -A chaotic sequence of numbers is used as an alternative to the random numbers. - It guides the search-pattern of the HHO algorithm.	-Jain, R15, Flame, Compound, Aggregation, D31, Spiral, Path-based, Wine, Iris, Glass &Yeast
Aljarah et al. [86], (2020)	-Local optima issue -Convergence rate -Unbalance within exploration and exploitation	-A proposed hybrid approach using Tabu search and GWO algorithm. -Employed TS as an operator in GWO.	-Iris, breast cancer, blood, glass, wine, seeds, Australian, diabetes, heart, liver, Haberman, tic-tac-toe & planning index,
Zhu et al. [87], (2020)	-Caught in local optima -Clustering accuracy	-Proposed an improved bat algorithm. -Gaussian-like convergence-factor and 5 more convergence-factors have been designed to improve global search. -Adopted hunting mechanism from WOA. - Local search enhanced using the sine position updating strategy.	-Iris, Seeds, Heart statlog, Wine, WDBC, Bupa, & WBC
Kaur et al. [88], 2020	-Slow Convergence -Local Optima	- Presented chaotic FPA (CFPA) for data clustering. -It is based on chaos-optimization and FPA thu K-means.	-Iris, Balance, Glass, E. coli, Dermatology, Breast Cancer, TAE, Spambase, Haberman, Heart, Leaf, LPD, Libras, Qualitative Bankruptcy, Wine, & Synthetic

2.2 MULTI OBJECTIVE CLUSTERING ALGORITHM

İnkaya et al.[89] proposed a multiobjective clustering method based on ACO and named ACO-C. The proposed algorithm optimises two objective functions, adjusted compactness and relative separation. The clustering solution is evaluated utilizing each objective function and neighbourhood information. The proposed algorithm can obtain Pareto-optimal solutions. The algorithm uses two steps during processing. First is neighbourhood construction, and second is data-set reduction. The first step extracts the local features of the data object, such as its local connectivity, density information, and proximity. The second step helps to reduce the storage and computation time needed for clustering. The proposed algorithm considered conditional merging as a local search method to enhance the exploitation ability. The cluster evaluations is performed through clustering evaluation comparative to the neighbourhood and weighted clustering evaluation close to the neighbourhood. The proposed ACO-C algorithm was tested over thirty-two datasets. The results were compared to several algorithms: single- linkage, *k*-means, NC closures, DBSCAN, and NOM. The performance measures used for evaluation are the Jaccard and the rand indexes. From the results, it is evaluated that the ACO-C algorithm is capable of working on varying densities and can form arbitrary-shaped clusters. In future, the proposed algorithm can be improved for outlier detection and reduced processing time.

Prakash and Singh proposed a multiobjective particle swarm optimization (TSMPSO) for solving hard partitional clustering problems[90]. The proposed TSMPSO adopted a two-stage diversity mechanism. Two objective functions: the sum of squared error (SSE) and connectedness functions, were considered for optimizing the clustering results. The homogeneity between the cluster was measured using SSE, whereas the separation was evaluated through connectedness. The algorithm initialized the positions of the particles as zero, and initial positions determine the personal best. The non-dominated solutions were preserved over the iterations, and the best was selected as the leader from the archived set of solutions. This further guided the search process. The performance was evaluated using benchmark clustering datasets Iris, Vowel, Glass, Wine, Zoo, Wisconsin Breast cancer (WBC), and Dermatology. The results given by TSMPSO were improved compared to MOPSO, MOABC and NSGA on coverage, distribution, convergence, f-measure and overall Non-dominated Vector Generation (ONVG).

Kishor et al. [91] proposed a new multi-objective optimization algorithm, namely NSABC (non-dominated sorting-based multi-objective artificial bee colony algorithm) to address local optima, population diversity and biasing issues. The proposed algorithm used Pareto-rank and crowding distance approaches to steer the population towards optimal PF. It was implemented as a fitness strategy. The non-dominated solutions were stored on their rank; the best was given to rank one, and so on. If the archive reached its maximum capacity, the crowding distance was computed, and it helped to push the solutions to non-crowded regions. The authors have optimized SSE and connectedness. The performance of NSABC was compared to various synthetic benchmark problems and nine standard clustering datasets. The different performance measures like accuracy, f-measure, NMI, B-cubed, coverage, convergence, distribution and ONVG were utilized. From the results, it is found that NSABC was capable of solving multi-objective optimization problems. The study will be applied to other multiobjective issues and investigated for different parameter settings.

Armano and Farmani [92] introduced a multi-objective particle swarm optimization (MCPSO) for partitional clustering problems. Data connectivity and cohesion were adopted for optimizing conflicting objectives. The proposed algorithm used the K-means for generating the initial swarm. The max-min strategy was utilized to get the Pareto optimal solutions, which helped improve the convergence and diversity of Pareto-optimal solutions. Further, the leader was chosen randomly from the top-ranked solutions. The particles used their cognitive and social knowledge to update the locations and move back to feasible/viable sites. The performance evaluation of the proposed MCPSO algorithm was evaluated using eleven real-life and sixteen synthetic datasets. The multiobjective particle swarm optimization (MCPSO) algorithm was proved to be an effective algorithm in terms of accuracy compared to other clustering algorithms. It is capable of finding the number of clusters automatically. Future works consider the enhancement of MCPSO and the investigation of it to high-dimensional datasets. Brain parcellation can also be studied using MCPSO from fMRI images.

Improved multiobjective clustering using an automatic k-determination algorithm (Δ-MOCK) was developed by Fabre et al. [93]. The amendments were regarding the initialization routine, efficiency, and reduced length of candidate representation. The algorithm utilized the NSGA-II-based search strategy during initialization to exploit high-quality partitions. The proposed algorithm uses the two reduced-length encodings scheme. The new-reduced length of partitions helped shorten large portions of search-space so that the exploration process emphasised the most promising solutions. The computational complexity was also reduced with the proposed amendments. The proposed algorithm showed improved clustering results by enhancing the convergence. The future directive is to develop opportunities for model selection of MOCK and Δ-MOCK in the study.

Penaloza et al. [94] contributed to the performance evaluation of four multiobjective approaches using an automatic k-determination algorithm that is original MOCK ($MOCK_{PESA-II}$) and its three variants: $MOCK_{MOEA/D}$, $MOCK_{NSGA-II}$, and $MOCK_{SPEA-2}$. Two constraints were employed. The first was related to the number of clusters that must be between 1 to 25. The second constraint was applied to a set of feasible solutions. Among these solutions, the dominating solution was chosen. If mutual solutions are viable and non-dominated by each other, randomly select one. If the tie is between one possible solution and the infeasible one, then a feasible one is selected. For the two infeasible solutions, choose the one with the lowest-sum-of-constraint violation. The two objective functions, compactness and connectivity, were optimized. Two sets of experiments were conducted. The first experiment used hypervolume and two-set coverage metrics. The second experiment used f-measure and silhouette coefficient. The performance of these algorithms was tested over seven datasets. The study revealed that $MOCK_{NSGA-II}$, performs significantly better than $MOCK_{PESA-II}$, $MOCK_{SPEA-2}$ and $MOCK_{MOEA/D}$.

A multi-objective teaching learning-based optimization algorithm was proposed by Esfahani and Saghaeic [95] for cluster analysis. Fuzzy C-means was incorporated as performance enhancement. Two objective functions used are degrees of proximity and cluster separation. The algorithm used J_m measure to increase the compactness of data in the cluster by minimizing the degree of proximity. Partition coefficient and exponential separation (PCAES) were used to evaluate the separation between cluster. The larger value indicated that clusters were more separated and compact. The different validity measures used during the evaluation of the proposed approach were: DB index, PBM, PC and XB index. The performance was evaluated on four artificial and four standard datasets. The simulation results were compared to single-objective algorithms.

Further, the authors have also evaluated the performance of the proposed algorithm for noise and compared it to MOITLBO. The objective function, J_m and XB index were used. The experiment results showed that the algorithm performs better with noisy data. Future work will investigate the proposed algorithm via different distance measures.

Zhou and Zhu [96] employed a kernel-based attribute weighting method and proposed a multiobjective genetic algorithm. The proposed algorithm optimized two objective functions, inter-cluster separation and compactness, to obtain optimal cluster results. The proposed algorithm contributed in three aspects. Firstly, the authors introduced multiobjective optimization using a feature-weighted-kernel clustering algorithm. Secondly, a new objective function was designed for optimization. In addition to it, the PSVIndex + CE method was adopted to handle large datasets. This method helps in achieving the final clustering results. It is observed that the proposed algorithm produces more promising results than the other competitive multiobjective algorithms. The future directives will consider extending the proposed algorithm to reduce computation time, investigate more suitable criteria and develop more objective functions.

Paul and Shill [97] combined fuzzy relational clustering (FRC) and multiobjective GA (NSGA-II). Two methods were proposed, namely, FRC-NSGA and IFRC-NSGA. The first one was used in data clustering and computed PageRank score for every object from every cluster. It measured the centrality of every cluster. It worked in two steps: (i) the Expectation step and (ii) the Maximization step. These helped in optimizing the cluster membership-values and mixing-coefficients. The latter was used in NSGA-II for producing initial membership-values of FRC-NSGA by reducing the randomness of the initial membership-values. The proposed algorithm optimized objective functions, cohesion and separation effectively. Further, FRC and IFRC were employed to handle various clustering problems, such as being stuck in local optima, overlapping clusters, and complexity in automatic clustering. Different benchmark datasets, including the gene-expression dataset and non-gene-expression dataset, were considered for experimentation. The experimentation results were compared to various other single and multiobjective classes of algorithms. The results show that the proposed methods successfully achieve stable, well-separated clusters and work with complex data. In future, the proposed methods will be employed for different real-world problems, like image segmentation, outlier detection, medical analysis and text mining.

Liu et al. [98] proposed a multi-objective clustering approach for novel multiple-distance measures. The improvements were made in terms of initial, crossover and mutation operator, and objective function design. The initial population was generated through NCUT pre-clustering method. The crossover operator selected the individual to produce some probability. For the selection of crossover individuals, it followed two strategies. The first strategy considered all individuals from the population. In contrast, the second utilized non-dominated individuals. But at once, only one crossover strategy can be applied. It is observed that the first strategy helped in achieving the diversity of new solutions. The latter strategy helped the algorithm to reach a globally optimum solution. Further, the mutation operator was also implemented in two ways for the crossover operator. This helped in increasing the diversity of the population. Additionally, the selection operator from NSGA-II was employed to select the individuals with the best fitness and reject those having low fitness values. Two objective functions were designed based on the modularity method from the literature. To find the Pareto-set, both objective functions need to be minimized. The performance evaluation was done through different standard datasets based on the f-measure and rand index. The proposed algorithm was able to determine an optimal number of clusters. The proposed algorithm can be improved to automatically choose the best individual from the nondominated set and reduce the computation time.

Wang et al. [99] proposed the posterior EMO-KC method to determine the number of clusters. For multi-clustering, the authors have investigated the use of EMO. The bi-objective model was developed. This model considered the sum of squared distance (SSD) and the number of clusters. Further, a new transformation strategy has been applied to SSD to ensure that the two objectives are conflicting. The model is then solved through EMO. The proposed algorithm employed parallelism as a feature in multiobjective evolutionary optimization (EMO). The performance of the proposed EMO-KC has been evaluated using three datasets and compared to NSGA-II and EMO algorithm. The EMO-KC gives better clustering results for multiple k values during a single run. The future directives will consider the proposed approach on other complicated datasets, using other validity indices as objective functions. Other transformation strategies will be explored to develop more efficient algorithms for large-scale optimization.

Liu et al. [100] proposed a local search reference vector-based method. The proposed method optimizes multiple clustering criteria simultaneously. The proposed algorithm generated reference vectors uniformly distributed over the objective space. The objective function was computed for each solution. Crossover and mutation operators were adopted from the NSGA-III algorithm to develop offspring solutions. Further, the algorithm used a local-search method for fast convergence. Environmental selection of NSGA-III was used for updating the population. The final selection was made through a knee-pruning fuzzy ensemble method. The proposed method was tested over fifteen datasets. The results were evaluated based on rand index, f-measure and NMI. From the experimental result, it is found that the proposed method helps in achieving robust and accurate clustering results. The proposed algorithm can be extended for categorical data and can be investigated in real-world clustering applications.

Prakash and Singh [101] proposed a multiobjective algorithm named IMBGSAFS. A genetic crossover operator was employed to improve the convergence and diversity in the MBGSAFS. The genetic crossover operator was introduced for enhancing diversity in binary gravitational search algorithm (BGSA) multi-objective optimization and named an improved multi-objective

BGSA for feature selection (IMBGSAFS). An unsupervised feature-selection approach was used in which no prior class information was needed. Two conflicting objectives optimized the silhouette index and feature cardinality. The Pareto-based method was used for obtaining diverse solutions and its performance was evaluated on eleven datasets. The results were compared to MBGSAFS, NSGA-II, MOPSO, and FM-CC based on coverage, convergence, distribution and ONVG. F-measure was used for finding single solution from pareto-solutions. The proposed IMBGSAFS proved to an efficient approach for feature selection and provided better clustering results. The work will be extended for text clustering.

An improved multi-objective artificial immune algorithm (KIFCM-IMOIA) using kernel-based intuitionistic fuzzy C-means clustering was proposed by Zang et al. [102]. Two optimised objective functions were intuitionistic fuzzy entropy (IFE) and kernel trick. The local optima issue was handled through an improved multi-objective optimization immune algorithm (IMOIA). A new active antibody selection strategy, a hybrid evolution strategy and an adaptive mutation operator were utilized in the proposed algorithm. The grid-based selection method was applied for selecting the active antibodies instead of crowding distance. The hybrid evolution strategy uses rand/1/bin and best/1/bin strategies. The first one helped improve the diversity of the population, and the latter helped enhance the convergence speed. The performance was evaluated using fourteen datasets. The experimental results were compared to six other clustering algorithms using accuracy, adjusted rand index, and normalized mutual index. The proposed algorithm has helped to achieve better convergence and provide quality clustering results.

Further, for handling the dynamic clustering problems and biasing issue of the single objective function, Prakash, Singh and Kishor [103] developed multiobjective particle swarm optimization (MOPSO) algorithm using a fitness operator named FPO-MOPSO. The proposed algorithm optimized the conflicting objective functions, the sum of squared error and connectedness. Thus, it applied the Pareto-based approach to generate different trade-off solutions. To handle the problem of local optima, a fitness predator optimizer (FPO) was introduced that enhances the diversity within PSO in multi-objective optimization scenarios. Additionally, a dynamic clustering method was presented that decides an optimum number-of-clusters in the range of 2 to $\sqrt{N}$, where N denotes the number of data objects present in the data set. MABC, NSGA-II and MOPSO have been considered for performance evaluation and evaluated results regarding the rand index, f-measure, convergence, diversity, coverage, and ONVG. The simulation results given by the FPO-MPSO are superior to algorithms in comparison. The proposed algorithm can be considered for application to document clustering in future.

A new hybrid partition selection algorithm (HSA) was proposed by Antunes et al. [104] to select the partitions. Variance and connectivity have been optimized together with the Pareto Front approximation algorithm. The aim was to divide regions with high-quality partitions. The proposed algorithm worked on a diverse set of partitions generated by other clustering algorithms having different purposes and biases. The HSA algorithm returned the reduced set-of-partitions. It was a three-step procedure- (i) the first step contained the PF for a set of base partitions using the multiobjective algorithm, (ii) the solutions from the PF were divided into several regions in the second step, and (iii) the third step used the ARI to select a solution from each region. The performance was evaluated on nine artificial and six real-life datasets. The simulation results were compared to Multi-Objective Clustering with automatic K-determination (MOCK) and Multi-Objective Clustering Ensemble algorithm (MOCLE). It is seen that HSA preserves the quality results by reducing the number of solutions. The future directives of the HSA algorithm will consider its application on other datasets and study its impact on the selection strategy.

Kuo and Zulvia [105] developed original gradient evolution (GE) algorithm for multi-objective clustering using k-means. It optimized two objective functions, namely, (i) sum of squared error between clusters (SSB) and (ii) the sum of distance within the cluster (SSW). The authors applied GE to the multiobjective problem. GE computed the initial centroids for K-means. The improvement was made in terms of vector updating. The Pareto rank assignment sorted the vectors based on their fitness value. Further, the authors applied the K-means algorithm for the final clustering result. The proposed GE-based K-means algorithm was tested over iris, wine, glass, yeast, Libras movement, Aggregation, Banknote Authentication, user knowledge, R15, and D31 datasets. The experimental results were compared to multiobjective clustering algorithms, namely, PSO, ABC, DE, GA and ABC. The results reveal that the proposed algorithm gives better results than other multiobjective metaheuristic-based algorithms. The proposed algorithm can be improved further to generate more diverse solutions.

For dimensionality reduction and multiobjective clustering, Liu et al. [106] developed an evolutionary algorithm named PCA-MOEA. The proposed algorithm uses the decision variables related to convergence and diversity. Diversity variables were initialised through a uniform sampling technique, and variables were converged randomly. For the convergence variable, PCA was applied to attain a lower representation. The decision variables count controlled the variance percentage. The inter-dependence analysis after this step helps find the relationship among convergence variables. The co-operative co-evolution frame was used to generate the population. The experiment was conducted using UF and WFG test suites through thirty variables. Problems with 200 and 1000 variables were considered in the experimentation. The performance measure used is Inverted generational distance (IGD), and it measures convergence and uniformity at once. The results of the PCA-MOEA were compared to several multiobjective approaches, namely, MOEA/D, MOEA/DVA, LMEA and MOEA/D-RGD. The only disadvantage is the extensive use of computing resources while grouping variables.

A new multiobjective differential evolution approach for simultaneous clustering and feature selection was developed by Hancer [107], named MODE-CFS. A variable string-length encoding scheme was employed to signify the promising solutions in the population. The encoding scheme uses a set of real numbers (in the range of $[-1,1]$) that represent locations of likely cluster-centroids and a set of activation codes (in the range of $[0,1)$ that represent the likely feature subset. The proposed algorithm used the centroid-based and feature-based mutation approaches for search space. The Pareto-optimal solutions were obtained using rank and crowding distance measures. The objective functions used in the algorithm are the Silhouette index, WB index and the number of features. The performance of the proposed algorithm was assessed using fourteen real-life and six synthetic datasets. The results showed that the proposed approach competes with various other clustering algorithms to improve the clustering performance and reduce dimensionality. The algorithm can be extended by investigating different objective functions and improving the efficiency of the proposed algorithm.

Further, single-objective/ multiobjective cat swarm optimization was proposed by Yan et al. [108]. The algorithm works in two modes- seeking mode and tracing mode. The cat position was updated using the simulated annealing method at some probability. In contrast, the quantum model was used in tracing mode to update the positions of cats in the global search space. The single-objective proposed method used the kernel method for cohesion-of-clustering. The multiobjective method utilized two objective functions, connectivity and cohesion, to discover the diverse set of solutions. Pareto optimization was applied to balance the conflicting objectives. The performance was evaluated on eight datasets, including one field dataset, four datasets from UCI, and three artificial datasets. The experiment results showed that the proposed multiobjective gives the highest accuracy and performs better than algorithms. The computation speed is affected during the iterations. Further, improvements may be made in this regard.

Chen et al. [109] introduced a multiobjective approach inspired by pigeon optimization, named combinatorial multi-objective pigeon-inspired optimization (CMOPIO). The proposed CMOPIO utilized the delta-locus encoding scheme for encoding pigeons. The dimensions of search space and length of pigeon representation were reduced in the proposed algorithm. It helped to reduce computational time. The proposed algorithm also used index-based ring topology to maintain diversity in the population. The performance was evaluated using various benchmark datasets, and results were compared to single-objective algorithms (K-means, K-medoids) and multiobjective algorithms (NSGA-II). It is noticed that the proposed CMOPIO algorithm is effective for obtaining optimal clusters.

Quality metrics along with ensemble strategy were integrated by Zhu et al. [110], and proposed enhanced EMO-based automatic clustering. Two encoding schemes were presented: (i) MOAC-L and (ii) MOAC-C. The authors contributed a new mating selection strategy from the neighbourhood. The reproduction strategy of MOAC-L included (i) a genetic operation that considered the entire population in the evolution process and (ii) a local search strategy applied to each solution. In comparison, MOAC-C used the centroid-based reproduction scheme. It makes use of simulated-binary-crossover and polynomial-mutation. The simulation was conducted using several real-life and synthetic datasets. From the results, it is observed that computational cost time was considerably decreased with proposed schemes as compared with state-of-art algorithms. The proposed algorithm did not require the prior information for cluster numbers to divide the data into clusters. A superior mechanism was required To represent chromosomes more effectively for unbalanced data, high-density regions etc.

Table 2.2: Related works on multiobjective clustering algorithms

Author Name [Ref], year	Issues	Approach	Data Sets
İnkaya et al. [89], 2015	-Prior information for the number of clusters -Local search ability -Biased solutions	-Developed ACO-C multiobjective algorithm for clustering. -Optimized two objective functions: adjusted compactness and relative separation	- Thirty-two datasets of 2 and higher dimensions with different shapes and densities
Prakash & Singh [90], 2015	-Population diversity -Local Optima -Biased solutions	-Proposed multiobjective particle swarm optimization (TSMPSO). -Optimized connectedness and sum of squared error.	-Iris, Glass, Vowel, Wine, WBC, Zoo, Dermatology
Kishor et al. [91], 2016	-Population diversity -Local Optima -Convergence rate -Biased solutions	-Proposed NSABC (non-dominated sorting-based multi-objective ABC) algorithm. -Employed fitness strategy for guiding the search -Optimized SSE and connectedness	-Iris, Glass, Vowel, Wine, Dermatology, CMC, Yeast, Segmentation, Wisconsin breast cancer (WBC)
Armano & Framani [92], 2016	-Biased Solution -Diversity	-Presented multi-objective PSO for data clustering. -Connectivity and cohesion were used as objective functions.	-Jain, Flame, Thyroid, Path-based, Spiral, WDBC, Compound, Aggregation, Unbalance, Glass, Yeast and 16 synthetic datasets
Fabre et al. [93], 2017	-Improve the scalability of the MOCK algorithm	-Proposed an improved version of the multiobjective clustering with an automatic k-determination algorithm. -Incorporated changes through a specialized initialization routine and two alternative reduced-length representations.	-Eight large datasets from street-level crime named as: UKC1, UKC2, UKC3, UKC4, UKC5, UKC6, UKC7, UKC8; 350 problems are synthetic datasets
Penaloza et al. [94], 2017	-Initial number of clusters -Local Optima -Biased Solution	-Compared performance of three algorithms for multiobjective problems -Optimized connectivity and compactness	-Wine, Dermatology, Wisconsin, Yeast, Iris, Seeds, User Knowledge Modeling

Esfahani & Saghaei [95], 2017	-Efficient cluster analysis	-Proposed multiobjective fuzzy-based clustering approach based on (multi-objective improved teaching-learning–based optimization) MOITLBO. - It helped in dynamically finding a number of clusters and exploring solution search space. '-Optimized two functions Jm and VPCAES index	-Artificial dataset1, artificial dataset2, artificial dataset3, artificial dataset4, Iris, Wine, Thyroid, & Red Wine
Zhou & Zhu [96], 2018	-Reduce computing time for large data sets	-Introduced multiobjective kernel-based clustering algorithm with attribute-weighted. -Utilized two objective functions-compactness (intracluster) and separation. (intercluster). -For quality clustering solutions, included sampling-operation and clustering ensemble-method accompanied by the projection-similarity-validity-index method.	-Iris, Wine, New thyroid, Breast, Image, Vertebral, WDBC, Bupa, & Seismic
Paul & Shill [97], 2018	-Problem of local optima -Number of clusters in advance	-Proposed two multiobjective algorithms as, FRC-NSGA and IFRC-NSGA. -FRC deals with coinciding issues of clusters. -NSGA-II generates initial values for FRC-NSGA in IFRC-NSGA. -Two objective functions considered are: Fuzzy compactness and overlap-separation	-AD_5_2, AD_10_2, Square-1, Square-4, Long-1, Glass, Wine, Iris, Liver Disorders, Leukemia, Lymphoma, Prostate tumour, Colon
Liu et al. [98], 2018	-Diversity - Initial number of clusters	-Based on different distance-measures, developed a multi-objective evolutionary algorithm. -The proposed algorithm integrated different distance measures using compactness. -Separation objective function helped find the appropriate number of clusters. -Optimized two objective - functions: Compactness and Separation	-Data_separated1 & 2, Data_connected1 & 2, Data_rect, Data_spiral, Data_circle1 & 2 R15, Jain, Pathbased, Iris, Spiral2, Wine, Soybean, & Glass

Wang et al. [99], 2018	-Initial number of clusters	-Presented EMO-k-clustering algorithm. -NSGA-II implemented for optimizing the objective functions. -Considered objective functions are sum-of-squared and the number of clusters.	-DS_100_4, DS_500_6, DS_900_7
Liu at al. [100], 2019	-Inaccurate Clustering results -Convergence rate	-Proposed a multi-objective evolutionary algorithm without any predefined coefficients. -Designed a local-search approach using reference-vectors to accelerate convergence. -Developed a knee-pruning fuzzy-ensemble method to choose the conclusive solution. -Optimized intra-cluster dispersion, inter-cluster separation, & negative Shannon-entropy of dimension-weights	-CNS tumours, Leukaemia, Lung Cancer, Normal, Novartis, St. Jude
Prakash & Singh [101], 2019	-Local Optima trapping -Feature-subset choice -Prior knowledge of class-info	-Proposed improved multiobjective approach for data clustering called IMBGSAFS. -Algorithm adopted: (i) unsupervised feature-selection method, (ii) Pareto-based method to attain diverse trade-off solution and (iii) genetic crossover-operator for improving diversity -Optimized Silhouette index and Feature cardinality	-Dermatology, Libras movement, Ionosphere, Heart Disease (HDD), Sonar, Parkinson, Vehicle, Sonar, Wisconsin breast (WPBC), Soybean Small, LSVT_voice_rehabilitation (LSVT-VR)
Zang et al. [102], 2019	-Local optima -Robustness against noise	-Presented KIFCM-IMOIA. -It is an improved multi-objective artificial-immune algorithm that utilizes kernel-based-intuitionistic fuzzy C-means clustering. -Hybrid DE -strategy and mutation-operator utilized in reaching an optimal solution and preventing local optima. -Optimized two objective	-Hayes-Roth, Haberman's Survival Data, Seeds, Glass Identification, E.coli, Balance Scale, Optical Recognition of Handwritten digits, CMC, Skin Segmentation, Letter Recognition (A/B) & (C/D), Shuttle, Occupancy Detection, & Electrical-Grid Stability-Simulated Data.

		functions: Compactness and Separation	
Prakas, Singh & Kishor [103], 2019	-Local optima -Lack of balance between exploration and exploitation -Prior information regarding the number of clusters -Premature convergence -Choice of the objective function -Biased Solution	-Proposed FPO-MOPSO. -Pareto-based methodology is adopted for finding diverse trade-off solutions. -To handle premature convergence of PSO, introduced fitness predator optimizer (FPO) in multi-objective optimization. -Optimized sum-of-squared-error & connectedness.	-Iris, Vowel, Wine, LD2, Parkinson's, Sonar, Vehicle, WPBC, Size5, Square1
Antunes et al. [104], 2020	-Local Optima -Biased solution	-Proposed new multiobjective clustering technique -Two objective functions, connectivity and variance, are used for dividing the dataset into partitions.	-Aggregation, chainlink, compound, monkey, R15, ds2c2sc13, spiral, spiralsquare, two diamonds, dyrskjot, eTongueSugar, glass, golub, iris, leukemia
Kuo & Zulvia [105], 2020	-Biased solution -Local Optima -Diversity	-Proposed a new GE-based k-means multiobjective algorithm. -Optimized sum-of-squared-error between clusters & sum-of-distance-within-cluster	-Iris, Wine, Glass, R15, D31, Libras movement, Bank note authorization, User knowledge modelling, Yeast, Aggregation."
Liu et al. [106], 2020	-Enhancing convergence -Population diversity -Biased Solution	-Developed PCA-MOEA -Adopted clustering approach: one correlated with convergence, and the other connected with diversity using a uniform sampling technique. -Optimized average squared error (ASE), the covariance matrix	-UF1–UF9, WFG1–WFG9 test problems
Hancer [107], 2020	-Reduce dimensionality -Optimal Solution	-Introduced a new multi-objective DE approach. -It helps generate homogeneous clusters. -Number-of-clusters are evolved automatically & reduce-dimensionality.	-Liver disorder, E-coli, Appendicitis, Pima, Iris, WBCD, Ionosphere, Sonar, Thyroid, Wine, UKM, Dermatology, Breast-tissue, & D10CB, D10C10, D50C10,

		-Optimized Silhouette-index, WB-index and feature-subset-size functions	D50C20, D100C10, D100C20
Yan et al. [108], 2020	-Optimal Solution -Improving accuracy	-Proposed single objective cat swarm optimization by optimizing cohesion, adopted kernel method. -Multi-objective CSO considers cohesion and connectivity. -Pareto-optimization is used for balancing the contradicting objectives.	-DS1, DS2 & DS3, Iris, WDBC, WCDS, Wine & BCW
Chen et al. [109], 2020	-Computational load -Population diversity	-Proposed combinatorial multi-objective pigeon-inspired optimization with ring-topology. -Employed delta-locus-based coding for encoding pigeons. -Reduces length of pigeon representation and search space dimensions. -Optimized connectivity and compactness	-Smile, Spiral, The first square, The second square, The third square
Zhu et al. [110], 2020	-Prior information for the number of clusters	-Proposed EMO-based automatic clustering algorithm. -Enhanced using two schemes-MOAC-L and MOAC-C	-Iris, Thyroid, Wine, Spiral, Flame, Aggregation, Path-based, 20d-10c, 50d-10c, 100d-10c, & 200d-10c

CHAPTER 3

IMPROVED WWO ALGORITHM FOR PARTITIONAL CLUSTERING

This chapter adopts a water wave optimization (WWO) algorithm for solving the partitional clustering problems. To date, different metaheuristic algorithms have been developed for solving the clustering problems for solving clustering problems. As per the no free lunch theorem, there is not a single metaheuristic algorithm that can be applied to a variety of datasets and also produces optimal clustering solutions. Hence, there is a scope to develop a new algorithm that can effectively solve optimization problems. This chapter investigates the capability of the water wave optimization algorithm for solving clustering problems. Before implementing the WWO algorithm, it was noticed that several shortcomings are associated with the WWO algorithm, like lack of global information, premature convergence, and imbalance between local and global search. First, these shortcomings are addressed through two amendments- the global best information component and decay operator. A detailed description of these amendments is explained in subsections 3.3.1-3.3.2. Further, the competency and effectiveness of the WWO are evaluated using benchmark clustering problems, and the simulation results are compared with several standards and hybrid metaheuristic clustering algorithms.

3.1 CONTRIBUTION

The main contributions of this chapter are given below:

- A global best information component is integrated into the search mechanism of the WWO algorithm to guide the search in the optimal direction.
- The premature convergence problem is handled through the decay operator.
- Eight benchmark clustering datasets are considered to evaluate the competency of the WWO algorithm.

3.2 FUNDAMENTAL OF WATER WAVE OPTIMIZATION

The WWO algorithm is inspired by shallow water wave theory and applied to solve global optimization problems [111]. The WWO algorithm denotes the solution search space in the seabed area. It represents the solution in terms of through waves, and a wave contains two components - height (h) and wavelength (λ). Further, each wave has a maximum height, denoted by h_{max}, and the wavelength of a wave is set to 0.5. The fitness of a wave fitness is assessed through seabed depth. The waves with higher fitness have a smaller distance to the stagnant water level. In WWO, three operations are defined at each iteration to achieve the optimal global solution. These operations are propagation, refraction and breaking. The propagation operation generates a new wave (X′) corresponding to wave (X) with displacement at each dimension (d). If the fitness of the new wave is better than an old wave, it replaces the old wave and sets the height of wave h_{max}; otherwise, wave height is decremented by one. The solutions search equation for WWO is expressed in equation 3.1.

$$X' = X + rand(-1, 1) \times \lambda \times L_d \tag{3.1}$$

Where rand generates random numbers between [-1, 1], L_d indicates length for d^{th} dimension in search space.

It is assumed that deep water waves contain higher wavelengths but lower heights. In comparison, shallow water waves have higher wave heights but lower wavelengths. The wavelength is decreased whenever a wave moves from deep to shallow water. This decrement of wavelength is computed using equation 3.2.

$$\lambda = \lambda \times \alpha^{\frac{-(f(X)-fmin+\varepsilon)}{(fmax-fmin+\varepsilon)}} \tag{3.2}$$

Where f_{max}denotes the maximum fitness value, f_{min} indicates the minimum fitness value of the current population, and α is the wavelength reduction-coefficient parameter. A small constant ε is used to prevent divide-by-zero, and f(X) denotes the fitness of the wave. When the wave height tends to zero, the refraction operator is applied. The new wave (X′) is computed using equation 3.3.

$$X' = \text{Gaussian}(\mu, \sigma) \tag{3.3}$$

Mean (μ), and standard deviation (σ) are calculated using equations 3.4-3.5, respectively.

$$\mu = \frac{X_{bestd} + X_d}{2} \tag{3.4}$$

$$\sigma = \frac{X_{bestd} - X_d}{2} \tag{3.5}$$

μ is calculated using existing wave (X_d) and best wave (X_{bestd}). σ is an average difference between the best wave (X_{bestd}) and the existing wave (X_d). The wavelength is computed through equation 3.6.

$$\lambda' = \lambda \frac{f(X)}{f(X')} \tag{3.6}$$

In equation 3.6, λ' denotes the wavelength of the next wave, $f(X')$ represents the fitness of the new wave (X'), f(X) denotes the fitness of the old wave, and its wavelength is denoted by λ.

The breaking operator is applied to break the wave (X) when a wave obtains a better local best solution to the current best solution (X_{best}). The functionality of the breaking operator to generate a new wave (X') is summarized in equation 3.7.

$$X' = X + \text{Gaussian}(0,1) \times \text{Ld} \times \beta \tag{3.7}$$

Where β indicates the breaking coefficient, $\text{Gaussian}(0,1)$ generates a random number between 0 and 1. If the wave X' is found better than X; X' replaces the old wave X.

The algorithmic steps of the basic WWO algorithm are mentioned in Algorithm 3.1

Algorithm 3.1: Steps of WWO algorithm	
Step 1:	Set initial population of P of n waves (solutions)
Step 2:	While (stopping criteria not met), do the following
Step 3:	For each wave X ∈ P
Step 4:	Propagate the wave (X) to a new position X' using equation 3.1.
Step 5:	If $f(X') > f(X)$, then
Step 6:	If $f(X') > f(X^*)$, then
Step 7:	Break the wave X' using equation 3.7.
Step 8:	Update the X^* with X'
Step 9:	Replace X with X'.
Step 10:	Else, reduce wave (X).h by 1
Step 11:	If wave(X).h=0, then
Step 12:	Refract the wave (X) to new X' using equations 3.3 and 3.6.
Step 13:	Update the wavelength via equation 3.2.
Step 14:	End while
Step 15:	Return best position of waves

3.3 PROPOSED IMPROVED WWO FOR PARTITIONAL CLUSTERING

This subsection presents an improved WWO algorithm for solving clustering problems called IWWO. It is noticed that the local search mechanism of the WWO algorithm is powerful, but the global search mechanism is weak [112]. So, due to a weak global search mechanism, sometimes the WWO algorithm does not converge on an optimal global solution. The literature shows that the PSO algorithm has a strong global-search procedure [113-114]. Hence, the weakness of the WWO algorithm is reduced through the global search process of the PSO algorithm. In WWO, the propagation phase is responsible for the global search mechanism, but the search equation lacks the global best component. Hence, a modified search equation is designed for the propagation phase inspired by the global search procedure of the PSO algorithm. WWO also suffers from premature convergence issues [115]. In the refraction phase, sometimes, wave height drastically decreases, in turn, tends to zero, and the algorithm converges without finding an appropriate solution, called premature convergence. This problem of the WWO algorithm is handled through a decay operator. It is integrated into the refraction operation phase of the WWO algorithm.

3.3.1 GLOBAL BEST INFORMATION COMPONENT

This subsection presents the proposed global best information component for the WWO algorithm. As discussed earlier, the WWO algorithm has a weak global search mechanism, so the WWO algorithm needs to get the optimal solution for fewer problems. Hence, to address this issue of WWO, an updated solution search equation is designed, which is inspired by the global search procedure of the PSO algorithm. The original global solution search equation is expressed in equation 3.1. It is seen that a new wave is produced without the support of the global-best and local-best information of the previous wave. The lack of information on global-best and local-best, in turn, affects the global search capability of WWO. Hence, to improve the global solution search mechanism of WWO, a global best information component is integrated into the solution search equation, inspired by the PSO algorithm's search mechanism. The updated solution search equation is mentioned in equation 3.8.

$$X' = X + C_{best} + rand(-1,1).\, L(d) \times \lambda \tag{3.8}$$

3.3.2 DECAY OPERATOR

In refraction operation, wave height is uninterruptedly decreased and suddenly tends to zero. In turn, the algorithm converges without finding the optimal solution, called premature convergence. The solution to this problem is to decrease the wavelength gradually in a stepwise manner. Hence, a decay operator is inculcated into the refraction operation to reduce the wavelength gradually. An updated search equation to determine the location of the wave using the decay operator is expressed in equation 3.9.

$$X' = N\left(\frac{X^* + X}{2}, \frac{|X^* - X|}{2}\right) \times [(1-\rho) + \Delta X] \tag{3.9}$$

In equation 3.9, ρ signifies the decay operator. It is defined as $\rho \in [0,1]$. ΔX denotes the difference between two consecutive waves; X denotes the current wave, whereas X^* denotes the current-best wave.

3.3.3 STEPS FOR IMPROVED WWO ALGORITHM

Algorithm 3.2 presents the step-by-step working of the proposed improved WWO algorithm. The flow diagram of the proposed improved WWO algorithm is explained in Figure 3.1.

Algorithm 3.2: Steps of Improved WWO algorithm	
Step 1:	Set the initial population of wave (C), such as $C_j \in (j = 1, 2, \ldots, n)$
Step 2:	Compute the objective-function value via equation number 1.1.
Step 3:	Consign the data objects to different waves based on minimum objective-function value. Find the best wave (C_{best}).
Step 4:	While (the stopping condition is not met), do the following
Step 5:	For each wave $X \in C$
Step 6:	Propagate the wave (X) to a new position X' using equation 3.8.
Step 7:	If $f(X') > f(X)$ then
Step 8:	If $f(X') > f(X^*)$
Step 9:	Break the wave X' using equation 3.7.
Step 10:	Update the X^* with X'
Step 11:	Replace X with X'.
Step 12:	Else, Refract the wave (X) to new X' using equations 3.9 and 3.6.
Step 13:	Update the wavelength using equation 3.2.
Step 14:	Determine the best wave (C_{best})

Step 15: End while

Step 16: Achieved the best position of waves (cluster centers)

3.4 EXPERIMENTAL RESULTS

This section presents the experimental results of the IWWO algorithm. The efficacy of the proposed IWWO algorithm is tested over eight datasets, namely, Glass, Vowel, Wine, Balance, Thyroid, Iris, Liver disorder (LD), and Contraceptive Method Choice (CMC). These well-known benchmark clustering datasets are downloaded from the UCI repository. The proposed improved WWO algorithm is evaluated over these eight standard clustering datasets. Table 3.1 gives information on standard clustering datasets used in this work. The proposed IWWO algorithm is implemented in MATLAB tool with window operating system on a corei5 processor with 8 GB.

3.4.1 PARAMETER SETTING

The user-defined parameter settings of IWWO are given as h_{max} =12, α=1.0026, k_{max}=12, β linearly decreasing from 0.25 to 0.001 and n=10. The parameter settings of other algorithms are considered the same as described in the corresponding literature. The simulation results of IWWO are described as the average results of thirty independent runs.

Table 3.1: Standard clustering datasets description

Sr. No.	Dataset	Clusters	Dimension	Instances
1	Glass	7	9	214
2	Vowel	6	3	871
3	Wine	3	13	178
4	Thyroid	3	5	215
5	Balance	3	4	625
6	Iris	3	4	150
7	CMC	3	9	1473
8	LD	2	6	345

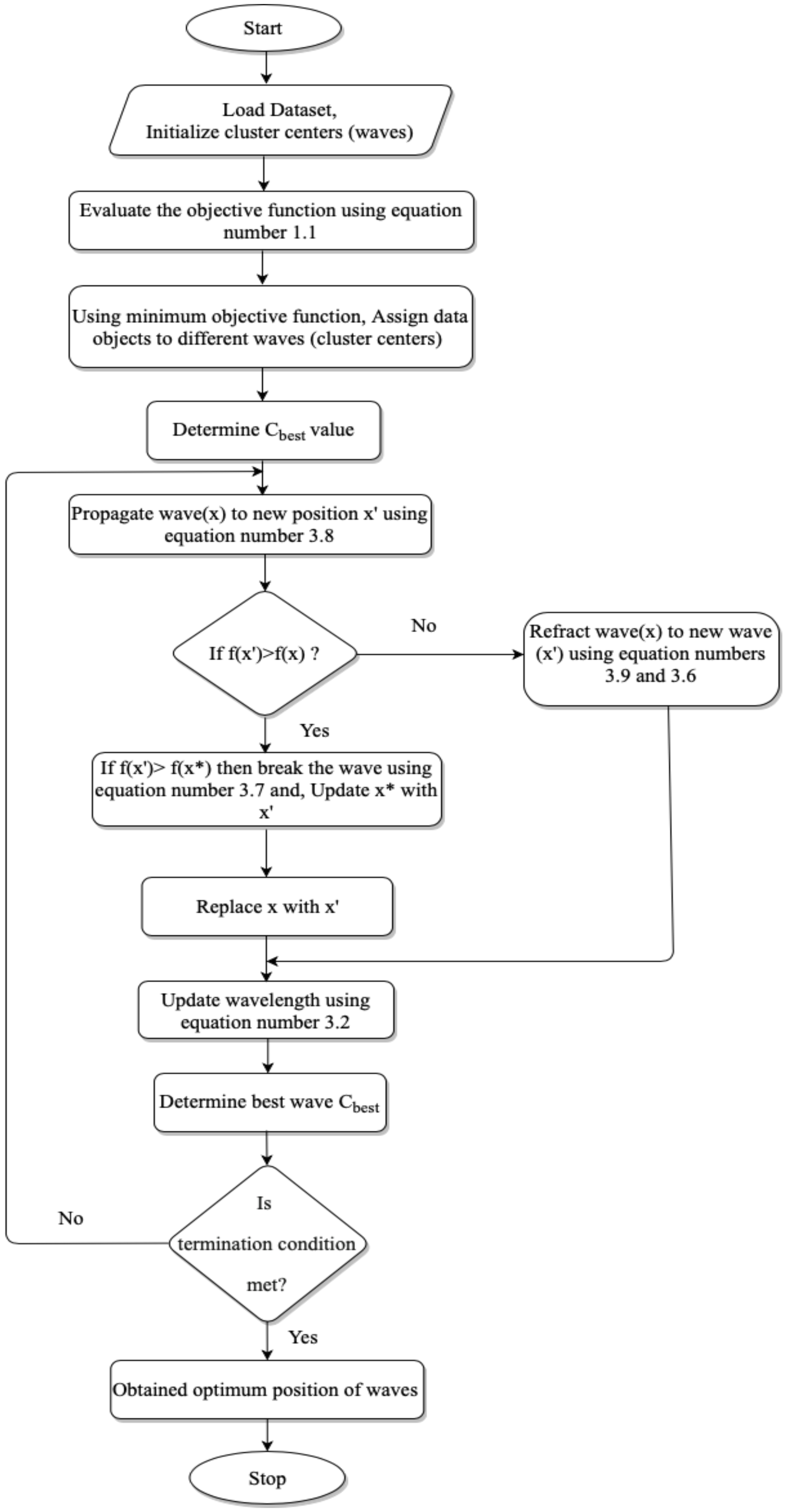


Figure 3.1: Flow diagram of proposed improved WWO algorithm

3.4.2 PERFORMANCE MEASURES

The various performance measures are applied to evaluate the simulation results of the IWWO algorithm. The description of these performance measures is given as

(i) Accuracy: The acceptability of an algorithm concerning actual class labels is delivered through accuracy. It is measured by mapping the actual label of an object "O" to cluster "c". The higher the value given, denotes more accurate the clustering results are. It is represented using equation 3.10.

$$\text{Accuracy} = \sum_{O=1}^{n} \delta(\text{Actual\_Label}, \text{map}(c))/n \tag{3.10}$$

(ii) Intra-cluster distance: It is calculated using the sum of-squared Euclidean distance between the cluster center and other data objects of the respective cluster. Minimum intra-cluster distance gives a better solution. It is computed through equation 1.1 given in Chapter 1.

(iii) F-measure: It is computed via precision and recall. The precision is the number of true positive results divided by the number of all true positives. In contrast, recall is the number of true positives divided by the number of all relevant results. For finding the value of F–measure, every cluster defines a query's outcome, and every class is represented through a set of permissions for a query. Thus, if each cluster "c" consists of a set of "n_c" data objects as an outcome of a query, and each class "p" consists of a set of "n_p" data objects needed for a query, then "$n_{p,c}$" gives the number of instances of class "p" within cluster "c". The recall and precision for each cluster "c" and class "p" is computed as equations 3.11 and 3.13, respectively.

$$\text{Recall}\,(R(p,c)) = \frac{n_{p,c}}{n_p} \tag{3.11}$$

$$\text{and} \quad \text{Precision}\,(P(p,c)) = \frac{n_{p,c}}{n_c} \tag{3.12}$$

Therefore, the f-measure is computed using equation 3.13.

$$F - \text{measure} = 2 * \left(\frac{P(p,c) * R(p,c)}{P(p,c) + R(p,c)}\right) \tag{3.13}$$

3.4.3 RESULTS AND DISCUSSION

This subsection discusses the simulation results of the proposed IWWO algorithm and other standard and hybrid clustering algorithms. The simulation results are evaluated using three well-known performance measures – accuracy, intra-cluster distance and f-measure. For a fair comparison, eight standard benchmark clustering algorithms widely adopted in literature are selected for comparative analysis. These algorithms are K-Means, GA, BAT, DE, ABC, PSO, BB-BC, and ACO. Despite this, several popular hybrid clustering algorithms are also chosen from the literature to compare the simulation results of the proposed IWWO clustering algorithm. These algorithms are the Modified Butterfly Optimization Algorithm (MBOA), Chaotic Teaching Learning Based Optimization (Chaotic TLBO), Improved Cat Swarm Optimization (ICSO), Krill-Herd algorithm with Harmony search (H-KHA), Memory-enriched Big bang-big crunch (MEBB-BC), Improved Krill-Herd (IKH), Improved Cuckoo search and modified PSO with K-Harmonic means (ICMPKHM), PSO based Big bang big crunch (PSO-BB-BC), and Cooperative bare bone PSO (CBPSO).

3.4.3.1 COMPARISON WITH STANDARD CLUSTERING ALGORITHMS

This subsection discusses the simulation results of the proposed IWWO algorithm and other standard clustering algorithms. The simulation results of the proposed IWWO algorithm and standard clustering algorithms using accuracy measures are reported in Table 3.2. The proposed IWWO algorithm provides more accurate results than other algorithms except for the balance dataset. In the case of the balance dataset, the PSO algorithm achieves a better accuracy rate, i.e. 89.76%. In contrast, the accuracy rate of the proposed IWWO algorithm is 88.78% for the balance dataset, which is the second highest among all other algorithms. It is stated that IWWO algorithms give more accurate clustering results with all datasets except balance. In the case of the balance dataset, it is also said that the proposed IWWO is a competitive algorithm compared to others.

Table 3.2: Accuracy measure results of proposed improved WWO and standard clustering algorithms

Dataset	Standard Clustering Algorithms							
	K-means	PSO	ACO	ABC	DE	GA	BB-BC	BAT	Improved WWO
CMC	39.69	44.1	36.89	40.06	39.58	43.3	44.67	42.62	**48.23**
LD	52.16	54.05	52.89	49.89	52.01	49.28	50.2	53.07	**96.51**
Thyroid	63.76	68.93	64.87	64.39	65.76	63.2	63.86	63.82	**90.23**
Iris	82.33	84.13	72.87	89.03	88.37	78.34	83.25	90.5	**93.21**

Glass	51.87	53.73	37.36	48.43	48.48	48.97	55.53	48.76	**68.81**
Wine	67.53	67.94	59.21	70.34	71.1	65.73	66.43	65.48	**74.63**
Vowel	51.16	84.04	51.69	56.31	53.41	84.7	84.32	57.21	**89.47**
Balance	84.99	**89.76**	74.28	76.67	74.96	78.01	79.69	86.75	88.78

The intra-cluster distance measure is also adopted for evaluating the results of the proposed IWWO concerning standard clustering algorithms. Table 3.3 demonstrates the intra-cluster distance results of the proposed IWWO algorithm and other standard clustering algorithms. From the results, it is seen that the proposed IWWO algorithm gives minimum intra-cluster distance for datasets- thyroid (1.04E+03), wine (1.63E+04), balance (5.22E+04) and vowel (1.53E+05). While IWWO also competes with the other standard clustering algorithms for the rest of the datasets.

Table 3.3 Intra-cluster distance measure results of improved WWO and other standard clustering algorithms

Datasets	Standard Clustering Algorithms								
	K-means	PSO	ACO	ABC	DE	GA	BB-BC	BAT	Improved WWO
CMC	**5.59E+03**	5.85E+03	5.83E+03	5.94E+03	5.95E+03	5.76E+03	5.71E+03	5.79E+03	5.75E+03
LD	1.17E+04	2.39E+02	2.41E+03	9.85E+03	1.15E+04	5.44E+03	**2.32E+02**	2.36E+02	1.13E+03
Thyroid	2.39E+03	1.11E+04	1.99E+03	1.98E+03	2.96E+03	1.22E+04	1.94E+03	1.39E+03	**1.04E+03**
Iris	**9.20E+01**	9.86E+01	1.01E+02	1.08E+02	1.21E+02	1.25E+02	9.68E+01	1.15E+02	9.30E+01
Glass	3.79E+02	2.76E+02	**2.19E+02**	3.29E+02	3.62E+02	2.82E+02	6.64E+02	3.75E+02	2.37E+02
Wine	1.81E+04	1.64E+04	1.64E+04	1.69E+04	1.68E+04	1.65E+04	1.67E+04	1.71E+04	**1.63E+04**
Vowel	1.60E+05	1.58E+05	1.89E+05	1.70E+05	1.81E+05	1.59E+05	1.94E+05	1.96E+05	**1.53E+05**
Balance	1.20E+05	6.20E+04	5.94E+04	6.61E+04	6.78E+04	6.91E+04	5.96E+04	6.02E+04	**5.22E+04**

Table 3.4 shows the results of IWWO with respect to standard clustering algorithms using f-measure. It is revealed that the proposed IWWO obtains a higher F-measure rate than other algorithms except for the wine dataset. For the wine dataset, BB-BC archives a higher f-measure rate, i.e. 0.566, while the proposed IWWO achieves a second higher f-measure rate, i.e. 0.531. The results show that IWWO gives better f-measure rates and a competitive algorithm for clustering problems.

Table 3.4: Demonstrate F-measure results for improved WWO algorithm and other standard clustering algorithms

Dataset	Standard Clustering Algorithms								
	PSO	GA	K-means	ACO	ABC	DE	BB-BC	BAT	IWWO
CMC	0.331	0.324	0.334	0.328	0.428	0.343	0.446	0.462	**0.509**
LD	0.493	0.482	0.467	0.487	0.508	0.485	0.524	0.536	**0.585**
Thyroid	0.778	0.763	0.731	0.783	0.796	0.768	0.784	0.789	**0.812**
Iris	0.782	0.778	0.78	0.779	0.783	0.773	0.781	0.782	**0.793**
Glass	0.412	0.561	0.426	0.402	0.411	0.406	0.462	0.431	**0.611**
Wine	0.518	0.515	0.521	0.522	0.519	0.518	**0.566**	0.529	0.531
Vowel	0.648	0.647	0.652	0.649	0.638	0.645	0.641	0.645	**0.655**
Balance	0.726	0.716	0.724	0.741	0.743	0.730	0.739	0.740	**0.744**

3.4.3.2 COMPARISON WITH HYBRID CLUSTERING ALGORITHMS

The comparative analysis of simulation results of the IWWO algorithm and other hybrid clustering algorithms based on accuracy measure is presented in Table 3.5. It is revealed that the proposed IWWO algorithm obtains better accuracy results for most of the datasets except the iris, glass, balance, and wine datasets. It is seen that MBOA gives a better accuracy rate for the iris (95.43), whereas improved WWO gives an accuracy rate of 93.21%. For the glass dataset, Chaotic TLBO obtains a higher accuracy rate of 69.52%, whereas improved WWO offers 68.81%. Similarly, for the wine dataset, H-KHA gives an accuracy rate of 75%, and improved WWO attains 74.63% of accuracy. The PSO-BB-BC provides higher accuracy of 89.21% for the balance dataset, whereas the improved WWO gives 88.78%. However, it is also stated that the proposed IWWO algorithm achieves comparable performances for most datasets and obtains either the second or third-highest results among all algorithms.

Table 3.5: Illustrates accuracy results of the proposed IWWO algorithm and hybrid clustering algorithms

Dataset	Hybrid Clustering Algorithms									
	MBOA	ICSO	Chaotic TLBO	H-KHA	MEBB-BC	IKH	ICMPKHM	PSO-BB-BC	CBPSO	IWWO
CMC	44.23	46.78	46.54	47.45	46.58	46.63	46.69	47.61	39.58	**48.23**
LD	50.67	53.02	53.12	51.91	49.86	52.96	52.15	52.17	53.65	**96.51**
Thyroid	59.36	68.24	67.38	65.4	65.22	66.91	66.82	56.83	72.21	**90.23**
Iris	**95.43**	91.35	91.19	89.24	90.02	89.87	92.44	90.52	90.79	93.21
Glass	58.73	69.06	**69.52**	58.89	58.73	68.39	69.02	69.52	51.92	68.81
Wine	70.31	73.24	72.53	**75**	72.63	72.37	72.88	73.58	71.31	74.63
Vowel	56.92	65.28	64.91	66.98	59.21	61.76	59.65	60.18	51.72	**89.47**
Balance	71.95	78.61	81.04	75.42	68.42	76.42	77.42	**89.21**	76.62	88.78

The simulation results of the proposed IWWO algorithm are also compared to other hybrid clustering algorithms using intra-cluster distance. The comparison of simulation results using intra-cluster distance measure is reported in Table 3.6. It is noticed that the IWWO algorithm gives minimum intra-cluster distance for the iris (9.30E+01), wine (1.63E+04) and balance (5.22E+04) datasets. For the CMC dataset, MBOA obtains minimum intra-cluster distance with a value of 5.21E+03, while improved WWO gives 5.75E+03. Similarly, for the LD dataset, ICSO gives a minimum intra-cluster value of 4.09E+02 and IWWO attains 1.13E+03 value. PSO-BB-BC gives a minimum intra-cluster of 9.82E+02 whereas improved WWO obtains 1.04E+03 minimum intra-cluster distance value for dataset thyroid. Further, it is perceived that

ICMPKHM gets minimized intra-cluster distance values for glass (1.99E+02) and vowel (1.47E+05) dataset, whereas improved WWO gives 2.37E+02 and 1.53E+05 for glass and vowel dataset respectively.

Table 3.6: Intra-cluster distance results of improved WWO and other hybrid clustering algorithms

Datasets	Hybrid Clustering Algorithms									
	MBOA	ICSO	Chaotic TLBO	H-KHA	MEBBC	IKH	ICMPKHM	PSO-BB-BC	CBPSO	IWWO
CMC	**5.21E+03**	5.32E+03	5.53E+03	5.60E+03	5.53E+03	5.69E+03	5.70E+03	5.57E+03	5.54E+03	5.75E+03
LD	1.32E+03	**4.09E+02**	4.98E+02	3.14E+03	1.36E+03	3.11E+03	3.09E+03	9.98E+03	1.00E+04	1.13E+03
Thyroid	2.16E+03	9.90E+02	1.08E+03	1.80E+03	1.26E+03	1.77E+03	1.39E+03	**9.82E+02**	1.86E+03	1.04E+03
Iris	9.83E+01	9.57E+01	9.69E+01	9.65E+01	9.68E+01	9.71E+01	9.58E+01	9.60E+01	9.69E+01	**9.30E+01**
Glass	2.31E+02	2.26E+02	2.38E+02	2.16E+02	2.27E+02	2.23E+02	**1.99E+02**	2.19E+02	2.13E+02	2.37E+02
Wine	1.71E+04	1.69E+04	1.68E+04	1.66E+04	1.68E+04	1.65E+04	1.67E+04	1.63E+04	1.64E+04	**1.63E+04**
Vowel	1.61E+05	1.59E+05	1.55E+05	3.52E+05	1.57E+05	1.56E+05	**1.47E+05**	1.55E+05	1.51E+05	1.53E+05
Balance	5.96E+04	5.39E+04	5.36E+04	6.78E+04	5.83E+04	6.01E+04	6.26E+04	6.19E+04	6.20E+04	**5.22E+04**

The experimental results of the proposed IWWO and other hybrid clustering algorithms using f-measure are illustrated in Table 3.7. The IWWO attains higher f-measure values for CMC, LD, Thyroid, Iris, and Glass datasets than other standard clustering algorithms. Whereas it also competes with the rest of the datasets. The ICMPKHM gives a higher f-measure value (0.558) for the wine dataset, whereas the f-measure rate of the IWWO algorithm is 0.531. H-KHA give the higher f-measure value (0.671) vowel dataset; for the same, IWWO achieves 0.655 f-measure rates. For the balance dataset, chaotic TLBO gives a higher f-measure rate (0.792), while IWWO obtains 0.744 f-measure rates. The experimental results show that IWWO is a competitive and efficient algorithm for clustering tasks.

Table 3.7: Illustrates F-measure results of proposed improved WWO algorithm and hybrid clustering algorithms

Dataset	Hybrid Clustering Algorithms									
	MBOA	ICSO	Chaotic TLBO	IKH	H-KHA	PSO-BB-BC	MEBBC	CBPSO	ICMPKHM	IWWO
CMC	0.435	0.339	0.345	0.457	0.443	0.461	0.438	0.389	0.456	**0.509**
LD	0.498	0.528	0.517	0.519	0.512	0.506	0.483	0.534	0.521	**0.585**
Thyroid	0.576	0.668	0.698	0.658	0.634	0.549	0.649	0.704	0.641	**0.812**
Iris	0.79	0.784	0.786	0.783	0.787	0.784	0.782	0.787	0.791	**0.793**
Glass	0.574	0.427	0.434	0.459	0.462	0.471	0.476	0.421	0.466	**0.611**
Wine	0.524	0.526	0.528	0.543	0.546	0.528	0.532	0.526	**0.558**	0.531
Vowel	0.634	0.646	0.635	0.663	**0.671**	0.652	0.648	0.651	0.649	0.655
Balance	0.704	0.767	**0.792**	0.736	0.739	0.764	0.687	0.734	0.741	0.744

3.4.3.3 CONVERGENCE BEHAVIOUR OF IMPROVED WWO

The convergence behaviour of the proposed IWWO and other standard algorithms like K-means, GA, BB-BC, BAT, ACO, PSO, ABC, and DE clustering algorithms is shown in Figure 3.2 (a-h). The X-axis denotes the number of iterations, and Y-axis represents the intra-cluster distance. It is concluded that the proposed IWWO algorithm gives a better convergence rate for most datasets. It is also stated that the proposed IWWO algorithm competes with well-known clustering algorithms and yields better clustering results.

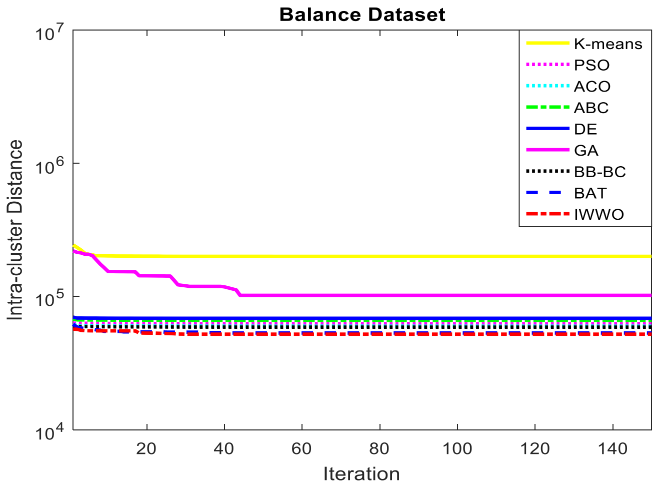


Figure 3.2 (a): Balance Dataset

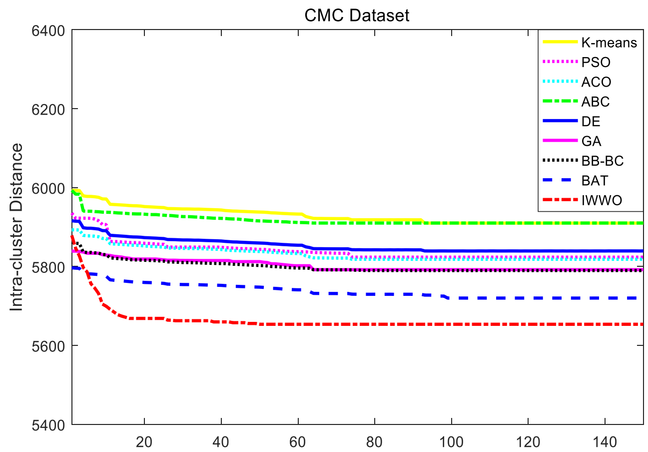


Figure 3.2 (b): CMC Dataset

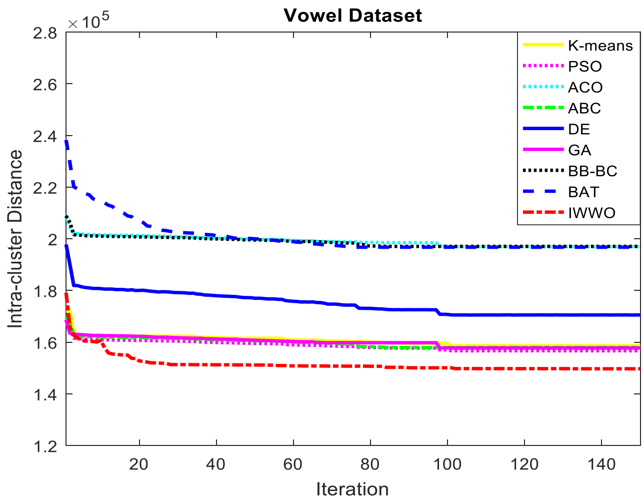


Figure 3.2 (c): Vowel Dataset

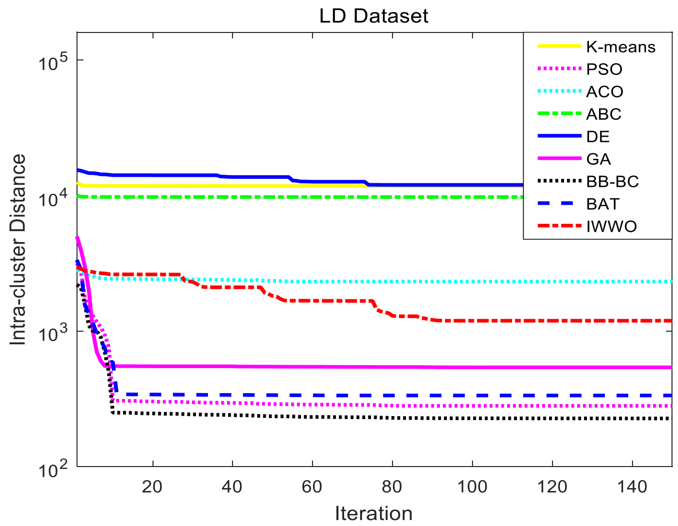


Figure 3.2 (d): LD Dataset

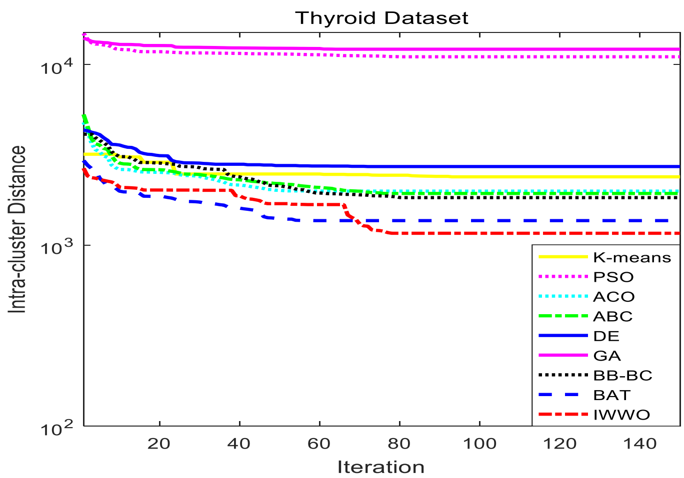


Figure 3.2 (e): Thyroid Dataset

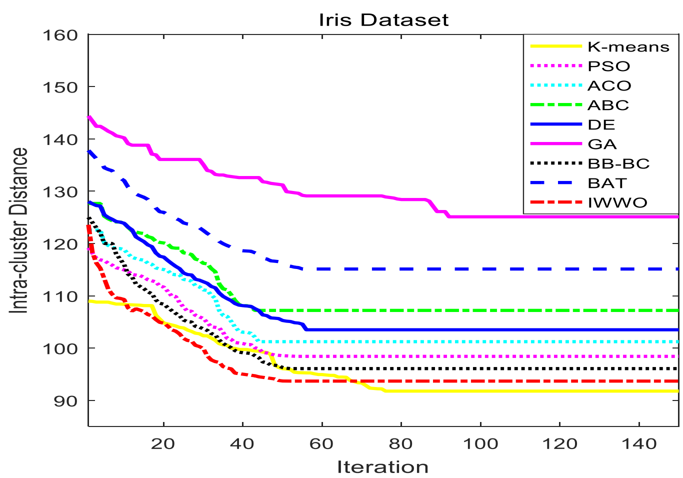


Figure 3.2 (f): Iris Dataset

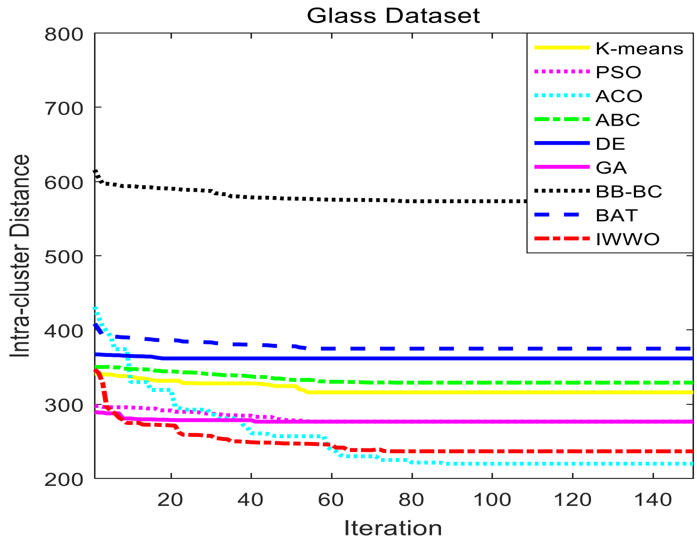


Figure 3.2 (g): Glass Dataset

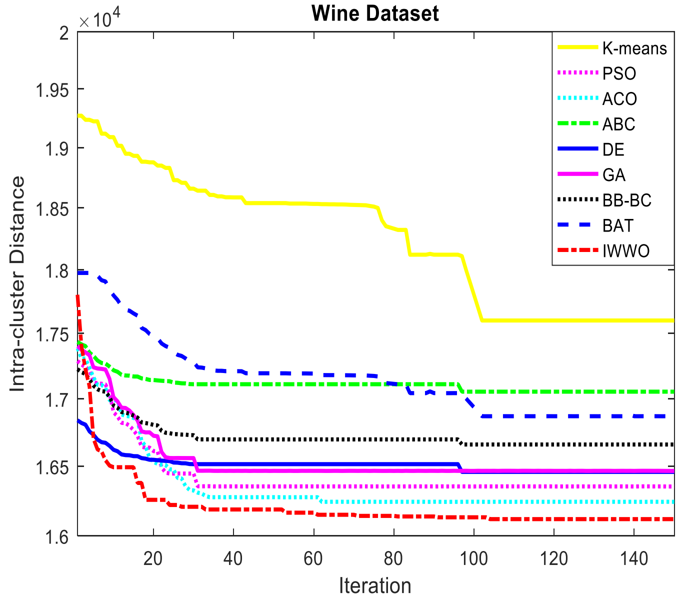


Figure 3.2 (h): Wine Dataset

Figure 3.2 (a-h): Convergence behavior of IWWO and standard clustering algorithms based on intra-cluster distance

3.5 SUMMARY

This chapter presents an IWWO clustering algorithm for effective cluster analysis. In the proposed IWWO algorithm, two amendments are recommended to improve the performance of the WWO algorithm. These amendments are (i) an updated global search mechanism and (ii) a decay operator. The first amendment is an updated global search equation based on PSO, and this amendment aims to improve the global search mechanism of the WWO algorithm. The second amendment corresponds to the inclusion of a decay operator, and this amendment seeks to handle the premature convergence issue of the WWO algorithm. The well-known eight benchmark datasets are considered for evaluating the performance of IWWO and tested over the accuracy, intra-cluster distance, and f-measure. Further, the results of the proposed IWWO clustering algorithm are compared with several popular standards and hybrid clustering algorithms. The results showed that the proposed IWWO clustering algorithm achieves an average higher accuracy of 12.90 % and 12.96 % for standard and hybrid clustering algorithms, respectively. The improvement in F-measure is an average of 1.7% and 5.2% in contrast to standard and hybrid clustering algorithms. On the analysis of the intra-cluster parameter, it is found that IWWO obtains minimum intra-cluster distance with most of the datasets in contrast to other algorithms. Hence, it is said that the IWWO algorithm is capable of performing clustering effectively.

CHAPTER 4
IMPROVED BAT ALGORITHM FOR PARTITIONAL CLUSTERING

This chapter presents an improved bat (IBAT) algorithm for solving partitional clustering problems. In literature, the bat algorithm (BA) is adopted for solving various optimization problems and provides competitive results compared to the same class of algorithms. But, several issues have affected the performance of the bat algorithm, such as lack of population initialization/selection methods, convergence rate, and trapped in local optima. This chapter addresses the issues mentioned above about the bat algorithm and provides viable solutions for improving the performance of the bat algorithm. An improved variant of the bat algorithm is developed after fixing all these issues called the IBAT algorithm. The performance of the proposed IBAT algorithm is tested on several clustering problems, and results are compared with popular standard and hybrid clustering algorithms.

4.1 CONTRIBUTION

This section highlights the contribution of this chapter. In the literature, it is found that several issues affect the outcome of the bat algorithm [116-118]. These issues can be summarized as a lack of population initialization concepts [118-119], trapped in local-optima in the last iterations [120-121], imbalance search mechanisms [121-122] and convergence speed [122-123]. The bat algorithm converges on near-optimal solutions rather than optimal solutions sometimes. The main contribution of this chapter is given below.

- To develop an enhanced co-operative co-evolution method for handling the initialization issue of the bat algorithm.
- To design an elitist strategy for improving the convergence rate.
- To develop a neighbourhood search mechanism for discovering optimal candidate solutions and also handling local optima.

4.2 BAT ALGORITHM

This section discusses the basic bat algorithm. The bat algorithm is inspired by the echolocation behaviour of micro-bats, especially prey detection and avoiding obstacles [125]. The micro-bat emits a short pulse in search of prey and also considers the echo of objects near to find the shape and size of prey. These characteristics of micro-bats can be defined as loudness and emission rate and can be computed using equations 4.1-4.2.

$$A_i^{t+1} = \alpha(A_i^t) \tag{4.1}$$

$$r_i^{t+1} = [1 - \exp(-\alpha)] \tag{4.2}$$

Here, A_i^t denotes loudness, r_i^t indicates pulse-emission rate, and α is the user-defined variable whose value is in the range of 0 and 1. Further, another parameter, called frequency, is also described for the bat algorithm and computed using equation 4.3.

$$f_i^t = f_{min}^t + (f_{max}^t - f_{min}^t)\text{rand}() \tag{4.3}$$

Where, f_{min}^t is the lowest frequency and f_{max}^t denotes the highest frequency at the time stamp t, and rand() generates a random number in between [0-1]. The velocity of bats is computed through equation 4.4.

$$v_i^t = v_i^{t-1} + (x_i^t - x_*)f_i^t \tag{4.4}$$

Where, v_i^{t-1} denotes the initial velocity, x_i^t represents the current position. x_* corresponds to the current best position. Thus, updated positions of bats are given through Equation nos. 4.5-4.6.

$$x_i^{t+1} = X_{new} + v_i^t \tag{4.5}$$

$$X_{new} = x_i^t + \text{randi}[-1,1]A_i^t \tag{4.6}$$

Here, X_{new} corresponds to the new position, and the final updated position is denoted by x_i^{t+1}. The algorithmic steps of the bat algorithm are given in algorithm 4.1.

Algorithm 4.1: Bat algorithm

Step 1: Specify the objective function f(x).

Step 2: Initialize bat population (x_i) and velocity (v_i) (i= 1 to n).

Step 3: Initialize loudness (A_i), pulse rates (r_i), and initial frequency (f_i) using equations 4.1 to 4.3.

Step 4: While($t < t_{max}$), do the following:

Step 5: Vary r_i and A_i.

Step 6: New solutions are generated by adjusting frequencies.

Step 7: Update (v_i) and position using equations 4.4 to 4.6.

Step 8: if (rand >r_i)

Accept solution from the best solutions and generate local solution from selected best solutions.

Step 9: End if

Step 10: Generate new solutions randomly

Step 11: If (rand >A_i) and $f(x_i) > f(x_j)$, then

Step 12: Accept the new solution.

Step 13: Rank the bats and find the current best solution.

Step 14: Check for termination conditions.

Step 15: Obtain optimal location of bats.

4.3 IMPROVED BAT ALGORITHM FOR CLUSTERING

The three improvements are recommended in the bat algorithm for improving its performance. These improvements are summarized as

(i) Population initialization issues are handled through enhanced cooperative co-evaluation.

(ii) An elitist strategy is designed for better trade-offs between search mechanisms and convergence rate.

(iii) Local optima issues and good candidate solutions are handled through a neighborhood strategy.

4.3.1 POPULATION INITIALIZATION

This subsection describes the population initialization issue. It is studied that the choice of initial cluster points greatly influences the efficiency of the clustering algorithm [118, 121]. Many studies are reported on initialization strategies and provided good solutions for the initial cluster selection problem [126-129]. An enhanced cooperative co-evolution strategy is developed to select initial cluster centers to improve the bat algorithm's performance. The proposed method follows the divide and conquer paradigm. It divides the dataset into sub-datasets. Further, sub-datasets are considered one by one for resolving the cluster center selection issue. The final solution is the outcome of each sub-dataset solution. Thus, the proposed cooperative co-evolution method is based on the centroid-selection mechanism. It considers three norms: (i) the number of partitions, (ii) partition size, and (iii) the choice of criteria for population initialization. The population can be described in terms of data instances, and it is divided among pre-defined partitions. Partitions for a given dataset are set to K number of clusters as shown in equation 4.7.

$$p_n \propto K \tag{4.7}$$

Where p_n indicates the number of partitions. K represents the number of clusters. The subpopulation size is computed using equation 4.8.

$$p_s = TP/K \tag{4.8}$$

Where TP denotes the total size of the population (i.e., data instances in a dataset), K corresponds to the number of clusters and p_s denotes subpopulation size. The number of subpopulations is calculated via equation 4.9.

$$\begin{cases} p_{s1} = 1 \text{ to } \lceil p_s \rceil \\ p_{s2} = \lceil UL(p_{s1}) + 1 \rceil \ldots \ldots \text{to} \ldots \ldots \ldots \lceil p_{s1} + p_s \rceil \\ \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \\ p_{s(n-1)} = \lceil UL(p_{s(n-2)}) + 1 \rceil \text{ to} \ldots + \lceil p_{s2} \rceil + \lceil p_{s1} \rceil + \lceil p_s \rceil n = \{1, 2, 3 \ldots \ldots K\} \\ p_{sn} = \lceil UL(p_{s(n-1)}) + 1 \rceil + \sum_{n=0}^{K-1} \lceil p_s \rceil \end{cases} \tag{4.9}$$

Where, $p_{s1}, p_{s2}, \ldots p_{sn}$ denotes subpopulations, and UL denotes the upper limit in a sub-population. Further, a suitable centroid is chosen for each sub-population using equation 4.10.

$$C_{kn} = \min(p_{sn}) + (\max(p_{sn}) - \min(p_{sn})) * \text{rand}(0,1); \text{ Where } n = 1 \text{ to } K \tag{4.10}$$

Here, C_k specifies k^{th} cluster-center, $\min(p_n)$ is minimum and $\max(p_n)$ is the maximum value of each n^{th} subpopulation (p_{sn}). rand () is a random number generator.

4.3.2 ELITIST STRATEGY

The convergence speed is another essential element for achieving good performance, especially for effective cluster analysis. The convergence rate depends on the search pattern of an algorithm to determine the optimal solutions. In this work, an advanced elitist strategy is designed to improve the convergence rate of the bat algorithm. This strategy considers the best position of the previous iteration, and this best position passes to subsequent iterations until it cannot become the second best. The elitist approach comprises two phases- (i) the evaluation phase and (ii) updating phase. The evaluation phase determines the personal best (X_{Pbest}) and global-best positions (X_{Gbest}). These positions i.e. X_{Gbest} and X_{Pbest} are computed using the comparison operator described in equations 4.11-4.12.

$$X_{Pbest} = \min(\text{fitness value}) \tag{4.11}$$

$$X_{Gbest} = \min\,(\text{distance value}) \tag{4.12}$$

Further, a fitness function is used for obtaining the personal best (X_{Pbest}) and it is mentioned in equation 4.13.

$$F(C_K) = \sum_{K \in 1}^{K} \frac{SSE(C_K)}{\sum_{K=1}^{K} SSE(C_K)} \tag{4.13}$$

Here, SSE represents the sum-of-square error. C_K denotes to K^{th}centroid object, global best position (X_{Gbest}) is described using the minimum value of the distance function or objective function, and the personal best position (X_{Pbest}) is described using the minimum value of the fitness function.

Updating phase considers the personal and global best positions of the evaluation phase and updates the frequency, velocity, and position of the bat using the current best positions. A comparison between the current and previous best positions is done, and choose the best one. The process of comparison is summarized in equations 4.14-4.15.

$$X_{Pbest} = \begin{cases} X_{Pbest}^{t-1} = X_{pbest}^{t} & \text{fit(t)} <= \text{fit}(t-1) \\ X_{Pbest}^{t} = X_{pbest}^{t-1} & \text{fit(t)} >= \text{fit}(t-1) \end{cases} \tag{4.14}$$

$$X_{Gbest} = \begin{cases} X_{Gbest}^{t-1} = X_{Gbest}^{t} & s(t) \le s(t-1) \\ X_{Gbest}^{t} = X_{Gbest}^{t-1} & s(t) \ge s(t-1) \end{cases} \tag{4.15}$$

Further, the bat algorithm's frequency, velocity, and search equations are updated using equations 4.16-4.19.

$$f_i^t = \frac{\min(X_{Gbest}^t) + (\max(X_{Gbest}^t) - \min\,(X_{Gbest}^t))\beta}{\max\,(X_{Pbest}^t)} \tag{4.16}$$

$$v_i^t = v_i^{t-1} + (X_{Gbest}^t - X_{Pbest}^t) f_i^t \tag{4.17}$$

$$X_{new} = X_{Gbest}^t + \text{randi}[-1,1] \tag{4.18}$$

$$X_i^t = \begin{cases} X_{Gbest} & \text{if rand} > r_i \\ X_{new} + v_i^t & \text{otherwise} \end{cases} \tag{4.19}$$

where, f_i^t, v_i^t and X_i^t denotes the frequency, velocity and position of i^{th} bat, $\min(X_{Gbest}^t)$ denote minimum and $\max(X_{Gbest}^t)$ represents the maximum value of sum-function linked with X_{Gbest}^t position. $\max(X_{Pbest}^t)$ represents the maximum value of the fitness function associated with the personal best position. β denotes random value within 0 and 1.

4.3.3 NEIGHBORHOOD SEARCH MECHANISM

This subsection describes the Q-learning-based approach as a neighbourhood search mechanism for avoiding the local optima problem of the bat algorithm. It is seen that the performance of the clustering algorithm is degraded when an algorithm is trapped in local optima [130]. Researchers have developed different strategies to deal with local optima issues [131-132]. It is also handled through a Q-learning-based neighbourhood-search mechanism [133]. It consists of two steps – (i) identification and (ii) evaluation. Identification corresponds to determining neighbouring data objects and neighbourhood boundaries. Further, the Q-learning-based concept is applied to update the positions of initial cluster points. The Q-learning-based neighbourhood search mechanism is illustrated in Figure 4.1(a-c).

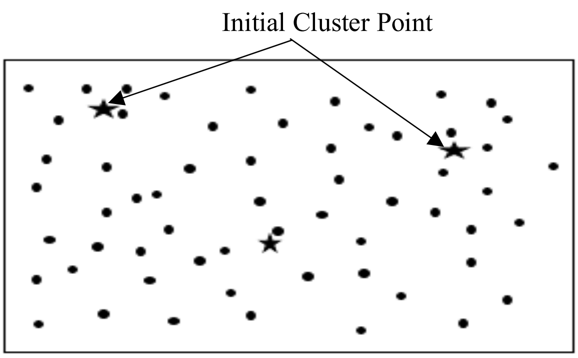


Figure 4.1(a): Data Objects and Initial cluster

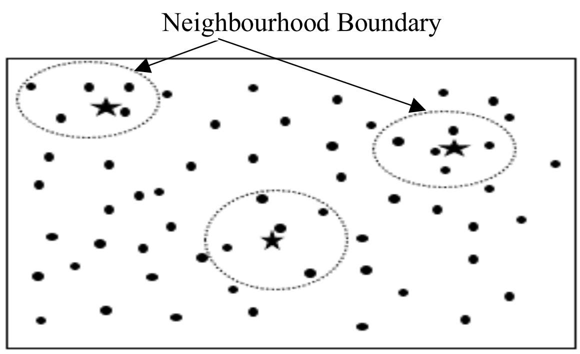


Figure 4.1(b): Neighbourhood Boundary and neighbouring

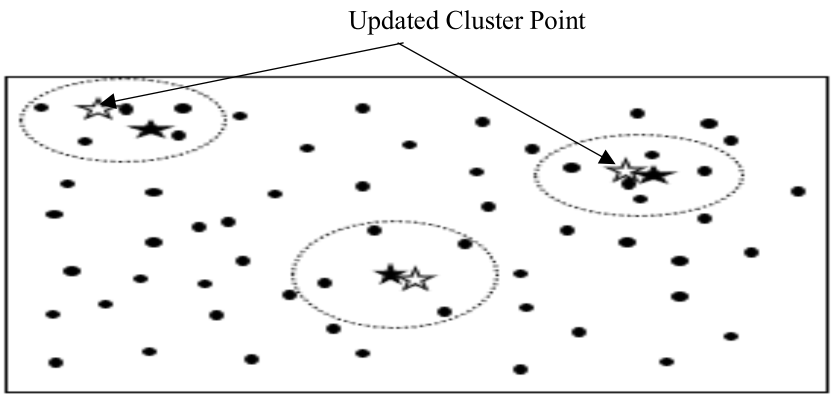


Figure 4.1(c): Evaluate new cluster point using Q-Learning

Figure 4.1 (a-c): Illustrate Neighbourhood Search Mechanism based on Q-learning

In evaluating phase, neighbouring data objects are determined using Euclidean distance measure. In this work, the neighbouring boundary is restricted to five data objects (minimum Euclidean distance from cluster points), and this process is illustrated in Figure 4.1(b). Let X_i corresponds to i^{th} cluster center and $X_{i,neigh}$ represents a set of neighbouring data objects of i^{th}cluster center. $X_{i,neigh}$ is described as $X_{i,neigh} = \{X_{i,1}, X_{i,2}, \dots \dots X_{i,5}\}$ where neigh=1 to 5.

The updated position of initial cluster points is also computed in the evaluation step shown in Figure 4.1(c). A Q-learning mechanism is applied to calculate the positions of neighbourhood data objects and update the cluster points. So, firstly, the Q-table is calculated, followed by action. The rewards are measured and based on the reward Q-table are updated. This process is summarized using equation 4.20.

$$Q'(s, a) \;=\; Q(s, a) \;+ \alpha * [R(s, a) + \gamma * \max Q'(s' + a') - Q(s, a)] \quad (4.20)$$

In equation 4.20, the new Q-value for the state (s) and action (a) is represented by $Q'(s, a)$, the current value is given by $Q(s, a)$, α denotes the learning rate. The reward for action for given state a is represented as $R(s, a)$, γ signifies the discount rate. The maximum expected future reward is denoted as the $\max Q'(s' + a')$.

4.3.4 STEPS OF IMPROVED BAT ALGORITHM FOR CLUSTERING

Algorithm 4.2 explains the step-by-step procedure of an improved bat algorithm for effective clustering.

Algorithm 4.2: Improved Bat algorithm for clustering

Step 1: Upload the dataset, and specify the number of clusters

$$K_i \in (i = 1, 2, \dots, n)$$

Step 2: Choose initial cluster locations using enhanced cooperative co-evolution strategy equations (4.7 to 4.10)

Step 3: Initialize pulse rates (r_i), loudness (A_i), and initial velocity (v_i) using equations 4.1-4.2

Step 4: Evaluate the objective function using equation 1.1 and assign objects to clusters with minimized objective function values

Step 5: Apply neighbourhood strategy to determine (X_{neigh}) position using equation 4.20

Step 6: Apply elitist strategy to calculate X_{Pbest} and X_{Gbest} positions using equations 4.11-4.12

Step 7: While (t < max_number_iterations), do following

Step 8: Calculate pulse frequency (f_i^t) and velocity (v_i^t) using equations 4.16-4.17

Step 9: If (rand >r_i)

Accept X_n position as solution

Else

Update the position using Equations (4.18-4.19)

Increase r_i and reduce A_i

Accept the new solution

Step 10: Recalculate the objective function and assign items with the minimum objective function values

Step 11: Update X_{Pbest} and X_{Gbest} using Equations 4.14 and 4.15, respectively

Step 12: Apply neighbourhood strategy

Step 13: Memorize the result and store it in the memory pool t=t+1;

Step 14: Check termination Condition

Step 15: Obtain a global optimal solution

Figure 4.2 illustrates the flow diagram of the proposed IBAT algorithm, and its working is divided into three phases: initialization, evaluation and updating. In the initialization phase, user-defined parameters are initialized. This phase also contains the enhanced cooperative co-evolution strategy for the selection of initial centroids. The evaluation phase computes the objective function described in terms of Euclidean distance. Further, the data objects are assigned to clusters with minimum objective function values. This phase also includes a Q-learning-based neighbourhood strategy to prevent local optima. If the candidate solution is not improved, it invokes the neighbourhood search mechanism. In the updating phase, the bats' position is updated through local and global search mechanisms. Further, the emission rates of the bat algorithm are compared with a random number [0, 1]. The neighbouring solution is accepted if the emission rate is higher than the random number. Otherwise, some variations are made in emission rate and loudness to compute a new value for the solution. The evaluation and updating phases are repeated until the criteria are not met.

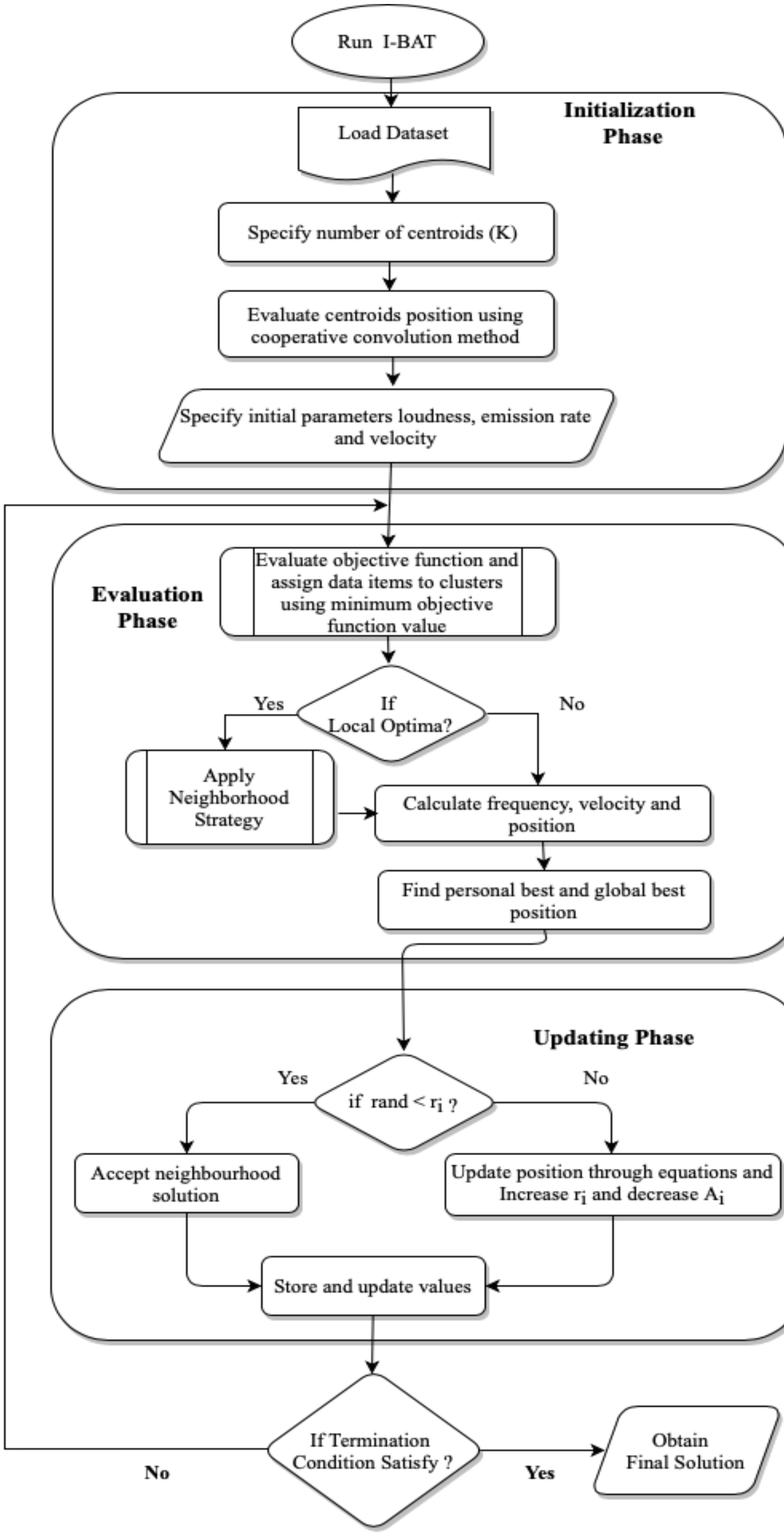


Figure 4.2: Flow diagram of improved Bat algorithm for clustering

4.4 EXPERIMENTAL RESULTS

This section presents the simulation results of the IBAT algorithm and other popular clustering algorithms. The performance of IBAT is assessed over eight well-known clustering datasets, and details of these datasets are mentioned in Table 3.1 of chapter 3. The proposed IBAT algorithm is implemented in MATLAB using Window OS, a corei5 processor with 8GB RAM. These well-known performance measures (accuracy, intra-cluster distance and f-measure) are considered for evaluating the performance of the IBAT algorithm. The IBAT algorithm simulation results are compared with several standards and hybrid metaheuristic clustering algorithms. The standard benchmark clustering algorithms are K-Means, GA, BAT, DE, ABC, PSO, BB-BC, and ACO, while the hybrid clustering algorithms are MBOA, Chaotic TLBO, ICSO, H-KHA, MEBB-BC, IKH, ICMPKHM, PSO-BB-BC and CBPSO.

4.4.1 PARAMETER SETTING

The parameter setting of the proposed IBAT algorithm is given below: population= K * d, loudness$(A_i) \in (0.1, 0.9)$, the initial velocity is set to 0.1, the value of α is 0.5, max_iterations are set to 100, and the number of partitions is considered as K.

4.4.2 RESULTS AND DISCUSSION

This subsection discusses the simulation results of the proposed IBAT and other popular clustering algorithms. The efficacy of the proposed IBAT algorithm is tested over eight clustering datasets and evaluated using accuracy, f-measure, and intra-cluster distance measures.

4.4.2.1 COMPARISON WITH STANDARD CLUSTERING ALGORITHMS

This subsection presents the simulation results of the proposed IBAT algorithm and standard clustering algorithms. The experimental results of the proposed IBAT and other standard clustering algorithms using accuracy are illustrated in Table 4.1. It is seen that the proposed IBAT algorithm obtains more accurate results for LD, thyroid, glass, iris, wine and CMC datasets. The accuracy rates of the IBAT algorithm for these datasets are 54.02%, 79.98%, 69.17%, 93%, 76.01%, and 48.21, respectively. For the Vowel dataset, GA achieves more accuracy, i.e. 84.7 %, while for the Balance dataset, PSO provides more accurate results, i.e. 89.76%. Table 4.2 presents the simulation results of the proposed IBAT and standard clustering algorithms using intra-cluster distance. The simulation results showed that the proposed IBAT algorithm achieves minimum intra-cluster distance with most datasets except for LD. It is seen that the BB-BC algorithm obtains minimum intra-cluster distance (2.32E+02) for the LD dataset, while the IBAT algorithm gets the intra-cluster distance (1.23E+03). Overall, it is said that IBAT is an efficient algorithm for data clustering.

Table 4.1: Accuracy results of proposed IBAT and standard clustering algorithms

Dataset	Simple Clustering Algorithms								
	K-means	PSO	ACO	ABC	DE	GA	BB-BC	BAT	IBAT
CMC	39.69	44.1	36.89	40.06	39.58	43.3	44.67	42.62	**48.21**
LD	52.16	54.05	52.89	49.89	52.01	49.28	50.2	53.07	**54.02**
Thyroid	63.76	68.93	64.87	64.39	65.76	63.2	63.86	63.82	**79.98**
Iris	82.33	84.13	72.87	89.03	88.37	78.34	83.25	90.5	**93**
Glass	51.87	53.73	37.36	48.43	48.48	48.97	55.53	48.76	**69.17**
Wine	67.53	67.94	59.21	70.34	71.1	65.73	66.43	65.48	**76.01**
Vowel	51.16	84.04	51.69	56.31	53.41	**84.7**	84.32	57.21	67.11
Balance	84.99	**89.76**	74.28	76.67	74.96	78.01	79.69	86.75	88.92

Table 4.2: Results of intra-cluster distance measure of proposed IBAT and standard clustering

Datasets	Simple Clustering Algorithms								
	K-means	PSO	ACO	ABC	DE	GA	BB-BC	BAT	IBAT
CMC	5.59E+03	5.85E+03	5.83E+03	5.94E+03	5.95E+03	5.76E+03	5.71E+03	5.79E+03	**5.52E+03**
LD	1.17E+04	2.39E+02	2.41E+03	9.85E+03	1.15E+04	5.44E+03	**2.32E+02**	2.36E+02	1.23E+03
Thyroid	2.39E+03	1.11E+04	1.99E+03	1.98E+03	2.96E+03	1.22E+04	1.94E+03	1.39E+03	**1.25E+03**
Iris	9.20E+01	9.86E+01	1.01E+02	1.08E+02	1.21E+02	1.25E+02	9.68E+01	1.15E+02	**9.16E+01**

Glass	3.79E+02	2.76E+02	2.19E+02	3.29E+02	3.62E+02	2.82E+02	6.64E+02	3.75E+02	**1.96E+02**
Wine	1.81E+04	1.64E+04	1.64E+04	1.69E+04	1.68E+04	1.65E+04	1.67E+04	1.71E+04	**1.61E+04**
Vowel	1.60E+05	1.58E+05	1.89E+05	1.70E+05	1.81E+05	1.59E+05	1.94E+05	1.96E+05	**1.51E+05**
Balance	1.20E+05	6.20E+04	5.94E+04	6.61E+04	6.78E+04	6.91E+04	5.96E+04	6.02E+04	**5.01E+04**

Further, the simulation results of the proposed IBAT and other standard clustering algorithms using F-measure are presented in Table 4.3. The proposed IBAT algorithm gives a higher f-measure rate among all datasets except the wine dataset. For the wine dataset, BB-BC shows a higher f-measure rate (0.566), while IBAT has an f-measure rate of 0.564. Hence, it is concluded that the IBAT algorithm is more competitive and proficient for solving clustering problems and provides state-of-art clustering results.

Table 4.3: Experimental results of proposed IBAT and standard clustering algorithms using f-measure.

Dataset	Simple Clustering Algorithms								
	PSO	GA	K-means	ACO	ABC	DE	BB-BC	BAT	IBAT
CMC	0.331	0.324	0.334	0.328	0.428	0.343	0.446	0.462	**0.501**
LD	0.493	0.482	0.467	0.487	0.508	0.485	0.524	0.536	**0.529**
Thyroid	0.778	0.763	0.731	0.783	0.796	0.768	0.784	0.789	**0.79**
Iris	0.782	0.778	0.78	0.779	0.783	0.773	0.781	0.782	**0.788**
Glass	0.412	0.561	0.426	0.402	0.411	0.406	0.462	0.431	**0.635**
Wine	0.518	0.515	0.521	0.522	0.519	0.518	**0.566**	0.529	0.564
Vowel	0.648	0.647	0.652	0.649	0.638	0.645	0.641	0.645	**0.653**
Balance	0.726	0.716	0.724	0.741	0.743	0.730	0.739	0.740	**0.794**

4.4.2.2 COMPARISON WITH HYBRID CLUSTERING ALGORITHMS

The performance of the proposed IBAT algorithm is also compared with hybrid clustering algorithms. The simulation results of IBAT and other hybrid clustering algorithms using accuracy are presented in Table 4.4. It is analysed that the proposed IBAT algorithm has more accurate results in the context of hybrid clustering algorithms. The accuracy rate of the proposed IBAT for LD, CMC, thyroid, wine and vowel datasets are 54.02%, 48.21%, 79.98%, 76.01%, and 67.11%, respectively. It is seen that MBOA gives higher accuracy (95.43%) for the iris dataset, while IBAT obtains a 93% accuracy rate for the same. In the context of glass and vowel datasets, the PSO-BB-BC algorithm achieves a higher accuracy rate of 69.52% and 89.21%, respectively, while IBAT obtains 69.17% and 88.92%.

Table 4.4: Accuracy results of proposed IBAT and hybrid clustering algorithms

Dataset	Hybrid Clustering Algorithms									
	MBOA	ICSO	Chaotic TLBO	H-KHA	MEBBC	IKH	ICMPKHM	PSO-BB-BC	CBPSO	IBAT
CMC	44.23	46.78	46.54	47.45	46.58	46.63	46.69	47.61	39.58	**48.21**
LD	50.67	53.02	53.12	51.91	49.86	52.96	52.15	52.17	53.65	**54.02**
Thyroid	59.36	68.24	67.38	65.4	65.22	66.91	66.82	56.83	72.21	**79.98**
Iris	**95.43**	91.35	91.19	89.24	90.02	89.87	92.44	90.52	90.79	93
Glass	58.73	69.06	69.52	58.89	58.73	68.39	69.02	**69.52**	51.92	69.17
Wine	70.31	73.24	72.53	75	72.63	72.37	72.88	73.58	71.31	**76.01**
Vowel	56.92	65.28	64.91	66.98	59.21	61.76	59.65	60.18	51.72	**67.11**
Balance	71.95	78.61	81.04	75.42	68.42	76.42	77.42	**89.21**	76.62	88.92

The simulation results of IBAT and other hybrid clustering algorithms are also evaluated using intra-cluster distance. Table 4.5 presents the simulation results of intra-cluster distance. It is observed that the proposed IBAT algorithm gives minimum intra-cluster distance for glass

(1.96E+02), balance (5.01E+04), wine (1.61E+04), and iris (9.16E+01) datasets. In contrast, the MBOA algorithm has a minimum intra-cluster distance (5.21E+03) for the CMC dataset. But, IBAT is also a competitive algorithm as it obtains the second minimum intra-cluster distance (5.52E+03) for the CMC dataset. For the LD dataset, ICSO achieves a minimum value of intra-cluster distance (4.09E+02), while IBAT obtains 1.23E+03 for the same. In the context of the thyroid dataset, the PSO-BB-BC algorithm achieves minimum intra-cluster distance, i.e. 9.82E+02, and IBAT has 1.25E+03 as minimum intra-cluster distance. In the case of the vowel dataset, ICMPKHM gives a minimum intra-cluster distance (1.47E+05), while for the same, IBAT achieves 1.51E+05 as the minimum intra-cluster distance. Overall, it is analysed that the proposed IBAT achieves minimum intra-cluster distance with most datasets, but for few datasets, it achieves second and third minimum intra-cluster distance.

Table 4.5: Intra-cluster distance results of proposed IBAT and hybrid clustering algorithms

Datasets	Hybrid Clustering Algorithms									
	MBOA	ICSO	Chaotic TLBO	H-KHA	MEBBC	IKH	ICMPKHM	PSO-BB-BC	CBPSO	IBAT
CMC	**5.21E+03**	5.32E+03	5.53E+03	5.60E+03	5.53E+03	5.69E+03	5.70E+03	5.57E+03	5.54E+03	5.52E+03
LD	1.32E+03	**4.09E+02**	4.98E+02	3.14E+03	1.36E+03	3.11E+03	3.09E+03	9.98E+03	1.00E+04	1.23E+03
Thyroid	2.16E+03	9.90E+02	1.08E+03	1.80E+03	1.26E+03	1.77E+03	1.39E+03	**9.82E+02**	1.86E+03	1.25E+03
Iris	9.83E+01	9.57E+01	9.69E+01	9.65E+01	9.68E+01	9.71E+01	9.58E+01	9.60E+01	9.69E+01	**9.16E+01**
Glass	2.31E+02	2.26E+02	2.38E+02	2.16E+02	2.27E+02	2.23E+02	1.99E+02	2.19E+02	2.13E+02	**1.96E+02**
Wine	1.71E+04	1.69E+04	1.68E+04	1.66E+04	1.68E+04	1.65E+04	1.67E+04	1.63E+04	1.64E+04	**1.61E+04**
Vowel	1.61E+05	1.59E+05	1.55E+05	3.52E+05	1.57E+05	1.56E+05	**1.47E+05**	1.55E+05	1.51E+05	1.51E+05
Balance	5.96E+04	5.39E+04	5.36E+04	6.78E+04	5.83E+04	6.01E+04	6.26E+04	6.19E+04	6.20E+04	**5.01E+04**

The simulation results of the proposed IBAT algorithm and other hybrid clustering algorithms are also compared using f-measure. Table 4.6 illustrates the simulation results of all clustering algorithms using the f-measure parameter. It is analysed that the proposed IBAT algorithm has higher f-measure rates with most datasets. It is also revealed that CBPSO achieves a higher f-measure rate (0.534) for the LD dataset, while IBAT obtains a 0.529 f-measure rate for the same. Moreover, the ICMPKHM algorithm gets a higher f-measure rate (0.791) among all algorithms for the iris dataset, while IBAT achieves 0.788 as the f-measure rate. The H-KHA algorithm achieves a higher f-measure rate (0.671) for the vowel dataset, and the IBAT algorithm obtains 0.653 as the f-measure rate. Hence, it is perceived that the IBAT algorithm provides a competitive f-measure rate compared to ICMPKHM, H-KHA, and CBPSO algorithms for a few datasets. Thus, it stated that the proposed IBAT algorithm is a competitive algorithm for solving clustering problems and provides superior results with most datasets in terms of intra-cluster distance, accuracy and f-measure.

Table 4.6: F-measure results of proposed IBAT and hybrid clustering algorithms

Dataset	Hybrid Clustering Algorithms									
	MBOA	ICSO	Chaotic TLBO	IKH	H-KHA	PSO-BB-BC	MEBBC	CBPSO	ICMPKHM	IBAT
CMC	0.435	0.339	0.345	0.457	0.443	0.461	0.438	0.389	0.456	**0.501**
LD	0.498	0.528	0.517	0.519	0.512	0.506	0.483	**0.534**	0.521	0.529
Thyroid	0.576	0.668	0.698	0.658	0.634	0.549	0.649	0.704	0.641	**0.79**
Iris	0.79	0.784	0.786	0.783	0.787	0.784	0.782	0.787	**0.791**	0.788
Glass	0.574	0.427	0.434	0.459	0.462	0.471	0.476	0.421	0.466	**0.635**
Wine	0.524	0.526	0.528	0.543	0.546	0.528	0.532	0.526	0.558	**0.564**
Vowel	0.634	0.646	0.635	0.663	**0.671**	0.652	0.648	0.651	0.649	0.653
Balance	0.704	0.767	0.792	0.736	0.739	0.764	0.687	0.734	0.741	**0.794**

4.4.2.3 CONVERGENCE BEHAVIOUR OF IBAT

The convergence behaviour of the proposed IBAT, GA, BB-BC, BAT, K-means, ACO, ABC, PSO and DE clustering algorithm is shown in Figure 4.3(a-h). The X-axis denotes the number of iterations, while Y-axis denotes intra cluster distance achieved by each algorithm. It is analysed that the proposed IBAT algorithm converges on minimum values except the LD dataset. The proposed IBAT algorithm gives most datasets a better convergence rate than compared algorithms. Therefore, it is said that the proposed IBAT algorithm is one effective and efficient clustering algorithm and provides better clustering results.

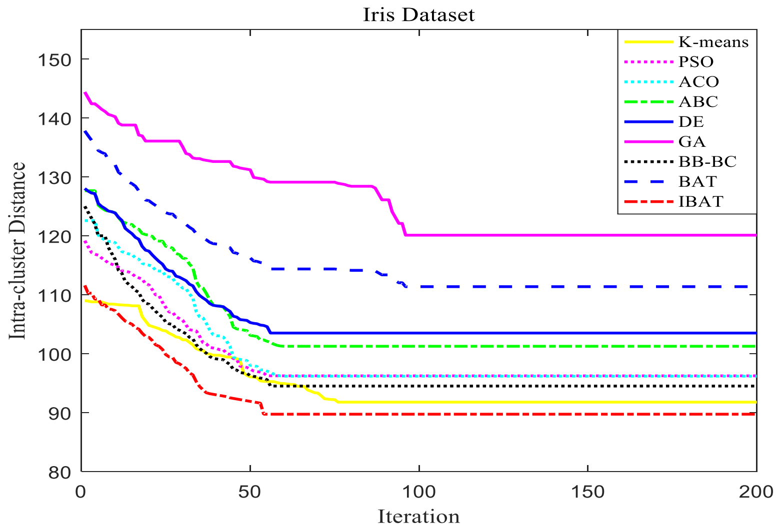


Figure 4.3 (a): Iris dataset

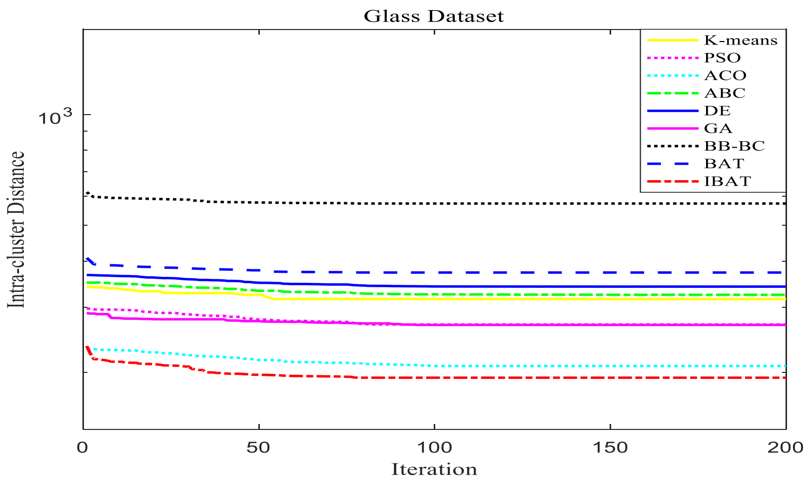


Figure 4.3 (b): Glass dataset

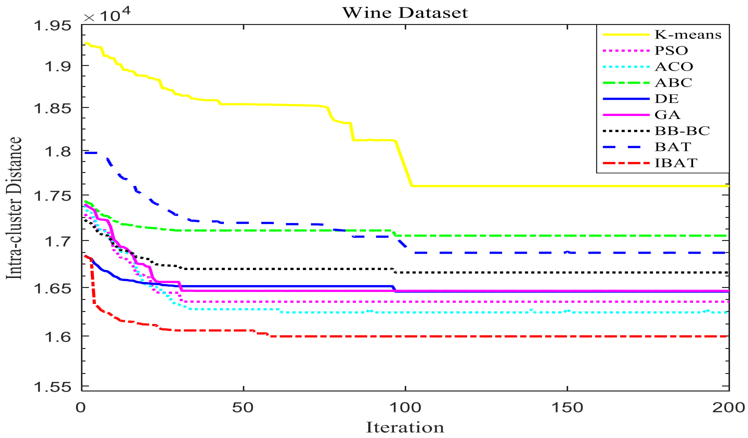


Figure 4.3 (c): Wine dataset

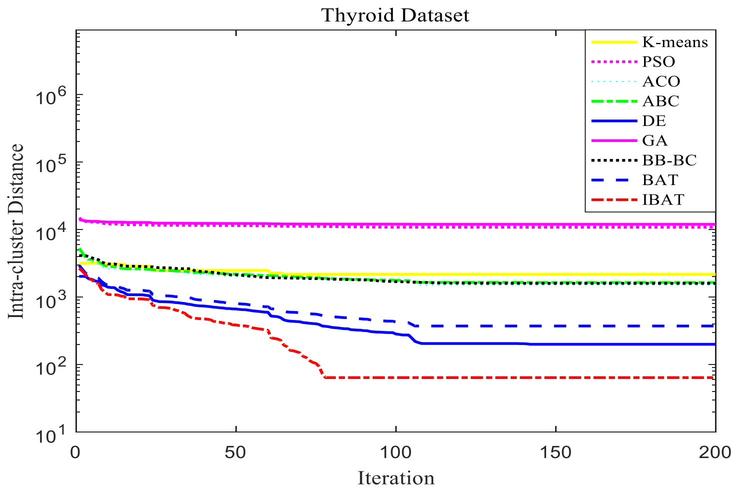


Figure 4.3 (d): Thyroid dataset

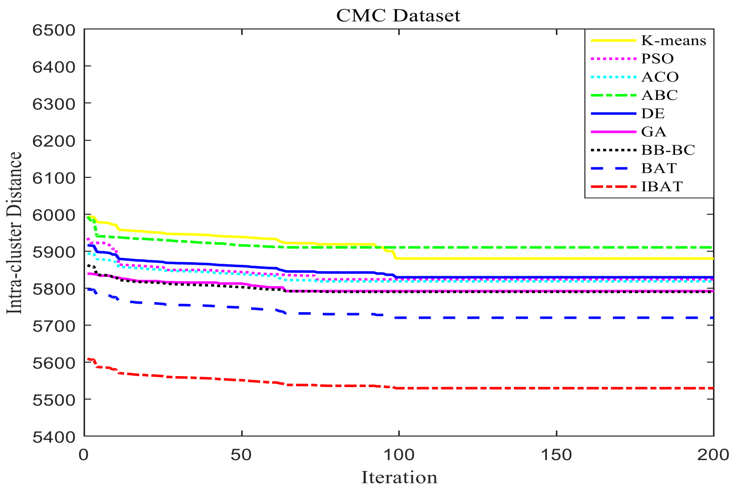


Figure 4.3 (e): CMC dataset

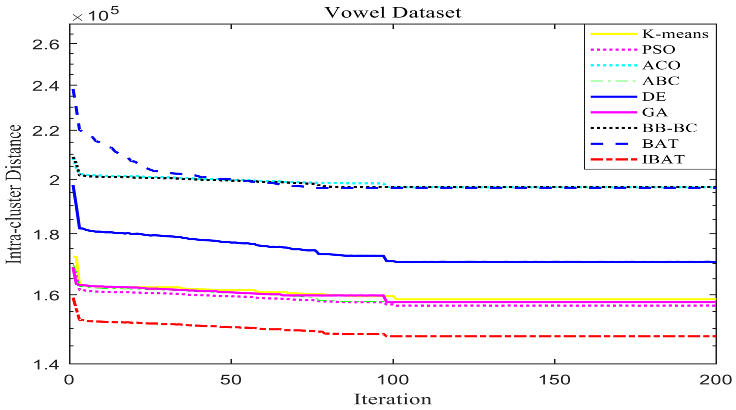


Figure 4.3 (f): Vowel dataset

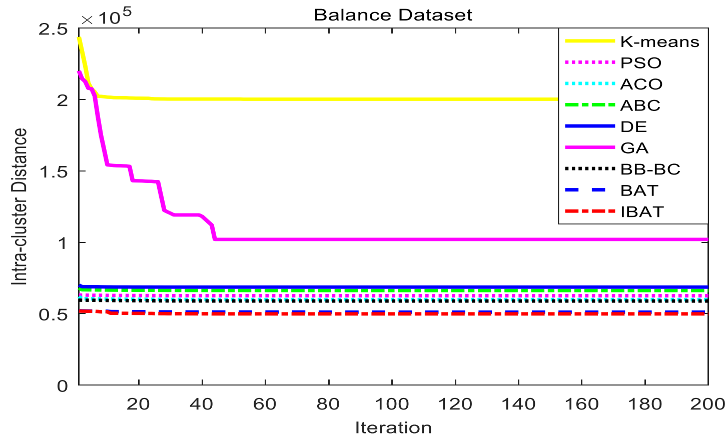


Figure 4.3 (g): Balance dataset

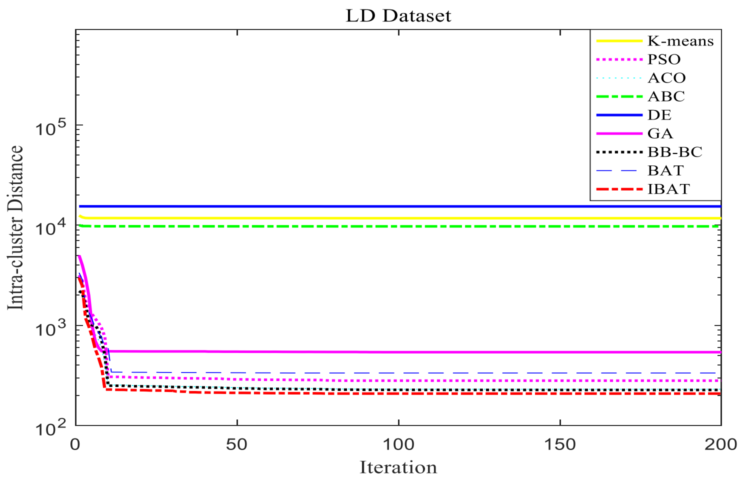


Figure 4.3 (h): LD dataset

Figure 4.3 (a-h): Convergence behavior of IBAT and standard clustering algorithms based on intra-cluster distance

4.5 SUMMARY

This chapter discusses an improved bat algorithm for partitional data clustering problems. The proposed IBAT algorithm addresses three well-known issues- initial population selection, local optima, and unbalanced search mechanism. The choice of initial population issue is resolved through an enhanced cooperative co-evolution method. The unstable search mechanism (local and global searches) issue is addressed through an elitist strategy. Further, the local optima issue is handled through a neighbourhood search mechanism. The performance of the proposed IBAT algorithm is tested over a well-known clustering dataset and compared with several popular clustering algorithms. The simulation results showed that the proposed IBAT algorithm achieves state-of-art clustering results in terms of accuracy, intra-cluster distance, and f-measure. The proposed IBAT algorithm achieves higher accuracy, an average of 3.73% and 3.77%, and improvement in F-measure results in an average of 1.9% and 5.4% compared to standard and hybrid clustering metaheuristic algorithms. Hence, it is stated that the proposed IBAT algorithm is an efficient algorithm for performing data clustering tasks.

CHAPTER 5
MULTIOBJECTIVE VIBRATING PARTICLE SYSTEM FOR CLUSTERING

This chapter presents a multi-objective vibrating particle system (VPS) algorithm for solving partitional data clustering problems. In literature, it is found that sometimes single objective clustering algorithms produce biased solutions because a single objective function is applied to solve the clustering problems. This problem can be resolved using more than one objective function for solving clustering problems. But, another issue regarding the selection of objective function is came into existence. However, it is found that the selecting objective functions are conflicting in nature. Hence, this chapter addresses the biasing issue of single-objective clustering through a multi-objective approach. A vibrating particle system algorithm with two objective functions is proposed for effective data clustering called the MOVPS algorithm. In the proposed MOVPS algorithm, intra-cluster distance and connectedness are two conflicting objective functions. Further, the efficacy of the proposed MOVPS is evaluated using eight standard clustering datasets, and simulation results are compared to several single and multi-objective clustering algorithms, including standard and hybrid clustering algorithms.

5.1 CONTRIBUTION

The section presents the contribution of the work. This chapter extends the vibrating particle system as multi-objective algorithms. Further, two objective functions (compactness and connectedness) are integrated into the VPS algorithm for solving the clustering problems effectively. MOVPS aims to optimize the two contradicting objective functions and provides the optimum clustering results. The intra-cluster variance is described as compactness. It is calculated as the distance between data objects to respective cluster centres. In contrast, connectedness investigates the neighbouring data elements and their connectivity to the cluster centres. The major contributions of the work are given as follows:

- A multi-objective vibrating particle system (MOVPS) algorithm is proposed for handling the biasing issue of single-objective clustering.
- Selection of objective functions to get more optimal clustering results. In turn, intra-cluster distance and connectedness functions are chosen in this work.

- Eight well-known clustering datasets are selected to evaluate the proposed MOVPS algorithm's performance.
- Simulation results are compared with several multi-objective and single-objective clustering algorithms.

5.2 ORIGINAL VIBRATING PARTICLE SYSTEM

This section discusses the basic vibrating particle system (VPS) algorithm. Kaveh and Ghazaan developed a metaheuristic algorithm inspired by the concept of free vibrations called the VPS algorithm [134]. Further, vibration can be categorized into free and forced vibration [134-136]. In VPS, the particles are described as candidate solutions, and the population is defined in terms of particles. These particles are randomly initialized in d- dimensional search space, and the aim of these particles is to achieve the equilibrium position gradually. The algorithmic steps of the VPS algorithm are discussed below.

(i) Initialization: This step corresponds to initialising the various user-defined parameters, population initialization, upper bound and lower bound of the optimization problems. The population of the VPS algorithm is described through particles. Hence, the initial position of particles in k-dimensional search space is computed using equation 5.1.

$$\mathrm{X_j} = X_{\min} + \mathrm{rand}() \times (\mathrm{X_{max}} - \mathrm{X_{min}}) \tag{5.1}$$

Where, $\mathrm{X_j}$ corresponds to j[th] particle, $\mathrm{X_{min}}$ represents the minimum and $\mathrm{X_{max}}$ the maximum value of each dimension and random numbers are generated via the rand () function.

(ii) Evaluate the candidate solutions: A problem-dependent objective function is defined, and the aim of particles compute the objective function for getting optimal results.

(iii) Update the particle positions: The position of particles are updated by considering the three position of particles in the previous iteration. These are described as (a) historical best (HB), the best position of a particle in the entire population, (b) Good Particle (GP), and (c) Bad Particle (BP). For selecting the position of GP and BP, first arrange the current population in ascending order based on objective function values, then select the GP from the first half and BP from the rest of the parts in random order. The updated position of particles is computed using equations 5.2-5.5.

$$\mathrm{X_{j,new}} = \mathrm{w_1} \times [\mathrm{D} \times \mathrm{A} \times \mathrm{rand1} + \mathrm{HB}] + \mathrm{w_2} \times [\mathrm{D} \times \mathrm{A} \times \mathrm{rand2} + \mathrm{GP}] + \mathrm{w_2} \times [\mathrm{D} \times \mathrm{A} \times \mathrm{rand3} + \mathrm{BP}] \tag{5.2}$$

$$A = [w_1 \times (HB - X_{j,old})] + [w_2 \times (GP - X_{j,old})] + [w_3 \times (BP - X_{j,old})] \tag{5.3}$$

$$D = \left(\frac{iter}{iter_{max}}\right)^{-\alpha} \tag{5.4}$$

$$w_1 + w_2 + w_3 = 1 \tag{5.5}$$

Where the new position of j^{th} particle is denoted by $X_{j,new}$; w_1, w_2 and w_3 assess the relative significance of HB, GP, and BP, respectively; rand1, rand2, and rand3 are uniformly distributed random numbers in the range of [0,1]. The current position of j^{th} particle is represented using $X_{j,old}$. D is descending function. It is defined as proportional to the number of iterations. iter denotes the current iteration, $iter_{max}$ signifies the maximum number of iterations, and α symbolizes a constant value.

Further, to examine the impact of BP in the position updating process, another parameter, p is defined in the range of [0, 1]. A comparison between parameter p and rand () function is performed for each particle. If, $(p < rand())$, then w_3 is set to 0 and w_2 is computed using $w_2 = 1 - w_1$. The above process invokes the impact of BP on the position updating whether BP actively participated in updating process or not. The algorithmic steps of VPS are mentioned in algorithm 5.1.

Algorithm 5.1: Vibrating Particle System

Step 1:	Initialize the various parameters of VPS such as population in terms of vibration particles, lower and upper constraints, max_iterations
Step 2:	Select initial position of particles in randomly using equation 5.1
Step 3:	Compute objective function values and store HB
Step 4:	While (max_iterations)
Step 5:	For each vibrating particle, do
Step 6:	Choose good particle and bad particle
Step 7:	If $P < rand$ then
Step 8:	Set $w_3 = 0$ and compute w_2 using equation 5.5
Step 9:	End if
Step 10:	For each vibrating particle,
Step 11:	Obtain new location of particle through equations 5.2 - 5.4

Step 12: End For

Step 13: If (violation of lower and upper constraints?)

Step 14: Apply Harmony search to update vibrating particle position

Step 15: End If

Step 16: End For

Step 17: Compute the objective function and update value of HB

Step 18: End While

Step 19: Obtain the optimal position of vibrating particles

5.3 MULTI-OBJECTIVE VIBRATING PARTICLE SYSTEM ALGORITHM

To handle the biasing issue of single objective clustering algorithm, multi-objective vibrating particle system algorithm is developed for effective cluster analysis. This section explains the multi-objective vibrating particle system (MOVPS) algorithm. In the multi-objective approach, more than one objective function is considered to eliminate the biasing effect of a single objective. It also ensures that both objective functions are conflicting in nature. Thus, the proposed MOVPS algorithm also considers the two conflicting objective functions. A detailed description of the MOVPS algorithm is given below.

5.3.1 POPULATION INITIALIZATION

The starting step of the MOVPS clustering algorithm is to initialize the population. The population of the algorithm is defined in terms of vibrating particles. These particles are defined in terms of the number of clusters present in the given dataset. Further, the initial position of vibrating particles is computed from the dataset in random order. A randomsample () function is applied to calculate the index of data objects for a given dataset. The selected data objects serve as the VPS algorithm's initial cluster centres/ positions.

5.3.2 OBJECTIVE FUNCTIONS

The selection of objective functions, especially in multi-objective approaches, significantly impacts the algorithm's performance. It is analyzed that every dataset is different in terms of distribution, dimension and characteristics. So sometimes, it is a tough task to achieve good quality results using the single objective function. Through literature, it is also found that a single objective function sometimes converges on biased solutions. This problem can also get rid off by using multiple objective functions. But, it is also challenging to choose the appropriate objective functions to obtain the optimal solution. It is also seen that if two or more objective functions are simultaneously optimized, a significant improvement can be achieved. Hence, this work considers two contradicting objective functions for achieving optimal clustering results. The objective functions are intra-cluster variance and connectedness. Intra-cluster variance is applied for allocating the data objects to respective cluster centers, and it is computed using equation 1.1 mentioned in Chapter 1. While compactness measures the quality of cluster. It gives information about the neighbourhood structure. For each data object, a penalty function is computed. The penalty is set to zero if the neighbouring data object stays in same cluster. Otherwise, it is computed using j^{th} nearest neighbour of m^{th} data object, which is provided as $\left(1/_{j}\right)$. Connectedness is computed using equation number 5.7.

$$\text{connectedness (M)} = \sum_{m=1}^{N}\left(\sum_{j=1}^{P} X_{m,\text{neig}_m(j)}/p\right) \text{ such that } \begin{cases} 1 \exists\, C_k: m, \text{neig} \notin C_k \\ 0, \quad \text{otherwise} \end{cases} \tag{5.7}$$

where M denotes the connectedness function, j^{th} nearest neighbour of data object m, is represented by $\text{neig}_m(j)$ for a given dataset. The number of neighbours is denoted by P, and N denotes the total number of data objects. The value of connectedness (M) lies between 0 and 1.

5.3.3 FITNESS FUNCTION

This subsection describes the significance of fitness function. Every metaheuristic algorithm initialises a population before starting the algorithmic procedure. It is also seen that a good population can be selected for computing the optimal solution to each metaheuristic algorithm. So, a fitness function is designed to choose a good population, and it can be described as a heuristic function that can evaluate the goodness of the population. Further, this function also determines the local best population, global best population and bad population for the metaheuristic algorithms. Hence to pick the GP, BP and HB particles, a fitness function is designed based on density (ψ), and distance (Δ) [137]. It associates a probability to each particle, whether it may act as cluster centres or not. The fitness of a particle is computed using equation 5.8-5.9.

$$F(d_{jk}) = \frac{1}{\psi \times \Delta} \tag{5.8}$$

$$\psi = \sum_{k=1}^{K} e^{\left(\frac{d_{jk}}{d_c}\right)} \tag{5.9}$$

Where in equation 5.9, d_{jk} is the Euclidean distance between j^{th} data object and k^{th} particle; d_c is the minimum significant distance and Δ represents the $\min\left(d_{jk}\right)$. The smaller the value of $F\left(d_{jk}\right)$, having more chances to be chosen as a cluster center.

5.3.4 PROPOSED MULTI-OBJECTIVE VPS CLUSTERING ALGORITHM

This subsection discusses the proposed multi-objective VPS clustering algorithm. The proposed multi-objective VPS algorithm is also considered the Pareto optimality. It is applied to explore the solution space and determine optimal candidate solutions presented in the search space. Further, Pareto optimality generates multiple non-dominating solutions instead of a single solution, as in single objective clustering. In addition, two conflicting objective functions are integrated into MOVPS to find the optimal partitioning for the dataset. The MOVPS algorithm starts with initializing the initial position of particles in a d-dimensional search space. The compactness among data objects is identified using Euclidean distance. Further, the neighbouring structure of data objects is revealed through the connectedness function. This function also computes a penalty function if data objects do not belong to the same cluster. A fitness function is associated with each particle and can measure the particles' goodness. Based on the fitness function, GP, BP and HB particles are selected. Moreover, the position of particles is updated using the position updating mechanism, but it also ensures that the updated position of particles does not violate boundary constraints. A set of non-dominating solutions are also generated, and these solutions are stored in an external archive.

5.3.4.1 STEPS OF MOVPS FOR CLUSTERING

The algorithmic steps of the proposed MOVPS are mentioned in algorithm 5.2, and the flow diagram is shown in Figure 5.1.

Algorithm 5.2: MOVPS Clustering Algorithm

Step 1: Initialize the population of MOVPS algorithm in terms of vibration particles i.e., pop_size, number of clusters (K), max_iter, neigh_particle, lower and upper constraints. Determine the initial position of random particles in random order.

Step 2: Create an external archive i.e., list to store the non-dominating solutions. Initially, list is empty.

Step 3: While (i<max_iter)

Step 4: For each vibrating particle, do

Step 5: Compute the objective function_1 i.e., intra cluster variance using equation 5.6.

Step 6: Allocate the data objects to nearest cluster (particle) using minimum variance value.

Step 7: End For

Step 8: For each data object, do

Step 9: Compute the objective function_2 i.e., connectedness using equation 5.7 and determine the penalty function and associate it with data objects.

Step 10: End For

Step 11: Determine the penalty of clusters (particles) using average penalty of data objects.

Step 12: For each vibrating particle, do

Step 13: Compute the fitness function of particle using equations 5.8-5.9.

Step 14: End For

Step 15: Determine the HB, BP and GP using fitness of particles.

Step 16: For each vibrating particle, do

Step 17: Compute the values of w_2 and w_3 using random function and these values satisfy equation 5.5.

Step 18: Update the position of vibration particles using equations 5.2-5.4.

Step 19: If (lower and upper constraints are violated)

Step 20: Generate new particle with in lower and upper constraints in random order.

Step 21: End If

Step 22: End For

Step 23: Stored the non-dominating solution in external archive (E).

Step 24: Update the list according to best non-domination solutions.

Step 25: i++

Step 26: End While

Step 27: Obtain the appropriate partitioning of data objects

5.4 EXPERIMENTAL RESULTS

This section presents the simulation results of the proposed MOVPS and other clustering algorithms. The efficacy of the proposed MOVPS algorithm is tested over eight well-known clustering datasets, and the simulation results are compared with several single and multi-objective clustering algorithms. The simulation results are evaluated using three well-known performance measures (accuracy, intra-cluster distance and f-measure). The Pareto front is also adopted as a performance measure for multi-objective clustering algorithms. The proposed MOVPS algorithm is implemented in MATLAB using Window OS, 8GB RAM and a corei5 processor.

5.4.1 PARAMETER SETTING

This subsection discusses the parameter settings of the proposed MOVPS algorithm. It is observed that optimal parameter setting of user-defined parameters can improve the final outcome of an algorithm and can be selected. The population of MOVPS is described as vibrating particles equal to the number of clusters present in the given dataset. The value of α = 0.2, neigh_particle is set to 5, and the size of external_archive is chosen to be 50. Further, the algorithm is run thirty independent times, and simulation results have presented an average of thirty separate runs. The parameter setting of other algorithms is similar, as reported in the literature.

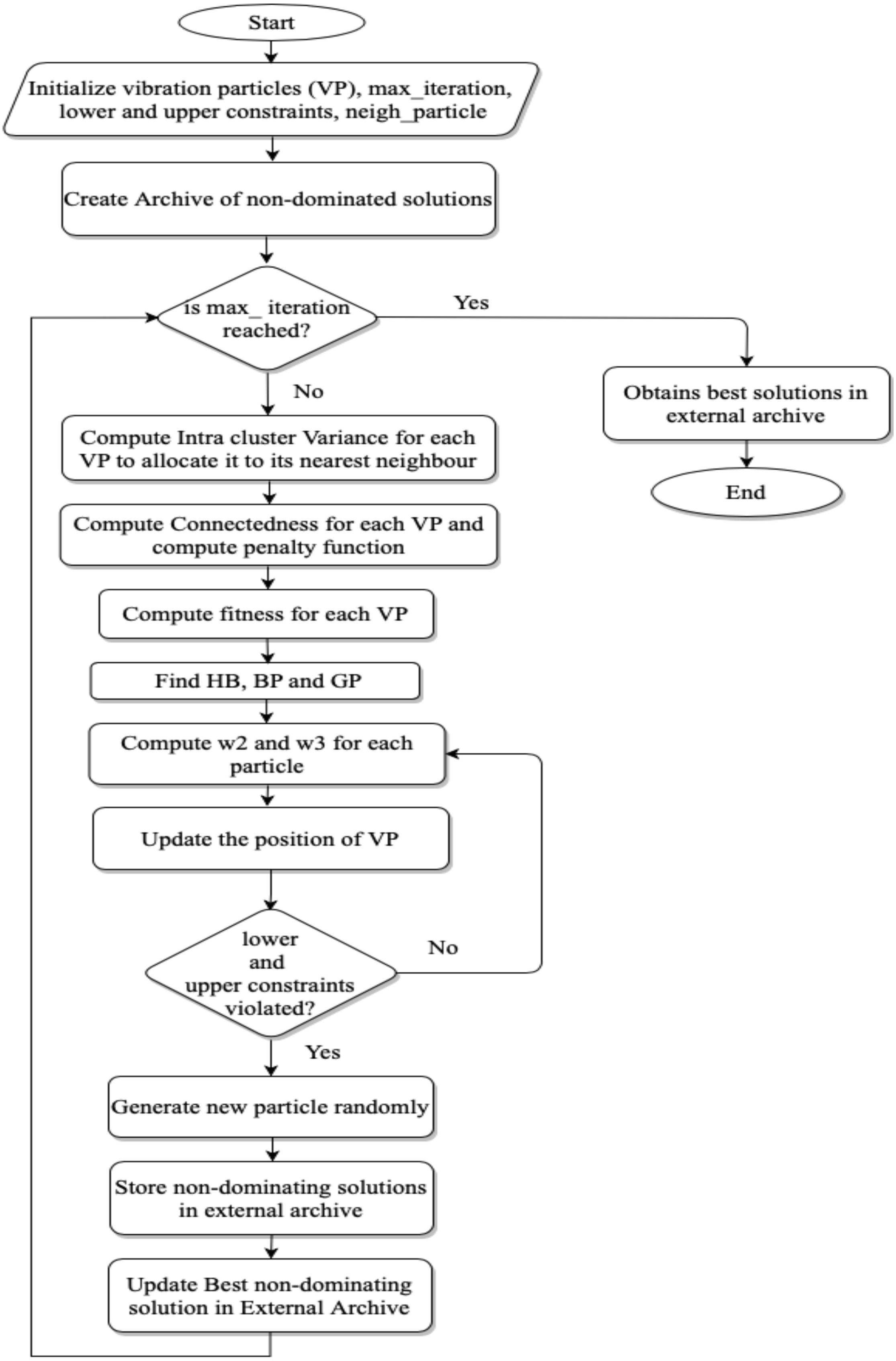


Figure 5.1: Flow diagram of MOVPS algorithm for clustering

5.4.2 RESULT AND DISCUSSION

The capability of the proposed MOVPS is tested on eight benchmark clustering datasets. The results are compared with standard, hybrid, and multi-objective clustering algorithms based on accuracy, intra-cluster distance, f-measure and Pareto fronts. This subsection discusses the simulation results of the proposed MOVPS and other algorithms being compared.

5.4.2.1 COMPARISON WITH STANDARD CLUSTERING ALGORITHMS

The experimental results of the proposed MOVPS and other standard clustering algorithms based on accuracy are illustrated in Table 5.1. It is noticed that the proposed MOVPS achieves a higher accuracy rate for CMC, LD, thyroid, iris, glass, wine, vowel and balance datasets in comparison to other standard clustering algorithms. The accuracy rate of MOVPS for the datasets mentioned above is 64.53%, 91.26%, 93.09%, 96.28%, 73.81%, 78.43%, 92.04% and 92.87% for balance datasets.

Table 5.1: Accuracy results of proposed MOVPS and other standard clustering algorithms

Dataset	Standard Clustering Algorithms								
	K-means	PSO	ACO	ABC	DE	GA	BB-BC	BAT	MOVPS
CMC	39.69	44.1	36.89	40.06	39.58	43.3	44.67	42.62	**64.53**
LD	52.16	54.05	52.89	49.89	52.01	49.28	50.2	53.07	**91.26**
Thyroid	63.76	68.93	64.87	64.39	65.76	63.2	63.86	63.82	**93.09**
Iris	82.33	84.13	72.87	89.03	88.37	78.34	83.25	90.5	**96.28**
Glass	51.87	53.73	37.36	48.43	48.48	48.97	55.53	48.76	**73.81**
Wine	67.53	67.94	59.21	70.34	71.1	65.73	66.43	65.48	**78.43**
Vowel	51.16	84.04	51.69	56.31	53.41	84.7	84.32	57.21	**92.04**
Balance	84.99	89.76	74.28	76.67	74.96	78.01	79.69	86.75	**92.87**

Further, the experimental results of proposed MOVPS and other algorithms using intra-cluster distance are reported in Table 5.2. It is observed that the proposed MOVPS obtains minimum intra-cluster distance for CMC (5.05E+03), thyroid (8.98E+02), glass (2.05E+02), wine (1.25E+04), vowel (1.47E+05), and balance (4.92E + 04) datasets. For the LD dataset, the BB-BC algorithm achieves minimum intra-cluster distance, i.e. 2.32E+02, While MOVPS achieves

9.24E+02 as intra-cluster distance. For the iris dataset, the K-Means algorithm has a minimum intra-cluster distance (9.20E+01), and MOVPS achieve 9.41E+01 as the intra-cluster distance for the same. Hence, simulation results indicate that MOVPS performs significantly superior to other single objective standard clustering algorithms in the intra-cluster distance.

Table 5.2: Intra-cluster distance results of proposed MOVPS and standard clustering algorithms

Datasets	Standard Clustering Algorithms								
	K-means	PSO	ACO	ABC	DE	GA	BB-BC	BAT	MOVPS
CMC	5.59E+03	5.85E+03	5.83E+03	5.94E+03	5.95E+03	5.76E+03	5.71E+03	5.79E+03	**5.05E+03**
LD	1.17E+04	2.39E+02	2.41E+03	9.85E+03	1.15E+04	5.44E+03	**2.32E+02**	2.36E+02	9.24E+02
Thyroid	2.39E+03	1.11E+04	1.99E+03	1.98E+03	2.96E+03	1.22E+04	1.94E+03	1.39E+03	**8.98E+02**
Iris	**9.20E+01**	9.86E+01	1.01E+02	1.08E+02	1.21E+02	1.25E+02	9.68E+01	1.15E+02	9.41E+01
Glass	3.79E+02	2.76E+02	2.19E+02	3.29E+02	3.62E+02	2.82E+02	6.64E+02	3.75E+02	**2.05E+02**
Wine	1.81E+04	1.64E+04	1.64E+04	1.69E+04	1.68E+04	1.65E+04	1.67E+04	1.71E+04	**1.25E+04**
Vowel	1.60E+05	1.58E+05	1.89E+05	1.70E+05	1.81E+05	1.59E+05	1.94E+05	1.96E+05	**1.47E+05**
Balance	1.20E+05	6.20E+04	5.94E+04	6.61E+04	6.78E+04	6.91E+04	5.96E+04	6.02E+04	**4.92E+04**

The simulation results using the f-measure rate of proposed MOVPS and other standard clustering algorithms are presented in Table 5.3. It is observed that the proposed MOVPS achieves f-measure rates 0.614, 0.693, 0.885, 0.947, 0.683, 0.739, 0.692 and 0.816 for CMC, LD, thyroid, iris, glass, wine, vowel, and balance dataset. So, it is stated that MOVPS provides superior clustering results than standard clustering algorithms in terms of accuracy, intra-cluster distance and f-measure rate.

Table 5.3: F-measure results of proposed MOVPS and other standard clustering algorithms

Dataset	Standard Clustering Algorithms								
	PSO	GA	K-means	ACO	ABC	DE	BB-BC	BAT	MOVPS
CMC	0.331	0.324	0.334	0.328	0.428	0.343	0.446	0.462	**0.614**
LD	0.493	0.482	0.467	0.487	0.508	0.485	0.524	0.536	**0.693**
Thyroid	0.778	0.763	0.731	0.783	0.796	0.768	0.784	0.789	**0.885**
Iris	0.782	0.778	0.78	0.779	0.783	0.773	0.781	0.782	**0.947**
Glass	0.412	0.561	0.426	0.402	0.411	0.406	0.462	0.431	**0.683**
Wine	0.518	0.515	0.521	0.522	0.519	0.518	0.566	0.529	**0.739**
Vowel	0.648	0.647	0.652	0.649	0.638	0.645	0.641	0.645	**0.692**
Balance	0.726	0.716	0.724	0.741	0.743	0.730	0.739	0.740	**0.816**

5.4.2.2 COMPARISON WITH HYBRID CLUSTERING ALGORITHMS

This subsection presents the simulation results of the proposed MOVPS and hybrid clustering algorithm. Table 5.5 shows the simulation results of MOVPS and hybrid single-objective clustering algorithms using accuracy measures. It is analyzed that the proposed MOVPS achieves more accurate results for all datasets as compared to other algorithms. The simulation results using intra-cluster distance are presented in Table 5.5. It is revealed that MOVPS obtains minimum intra-cluster distance for CMC, thyroid, iris, wine, vowel, and balance datasets. While, for LD and glass datasets, ICSO and ICMPKHM obtain minimum intra-cluster distances (4.09E+02, 1.99E+02), respectively, the proposed MOVPS achieves 9.24E+02 and 2.05E+02 as intra-cluster distances for the same. Further, the simulation results are also evaluated using the f-measure parameter. These results are reported in Table 5.6. It is analyzed that the proposed MOVPS achieves high f-measure rate than hybrid clustering algorithms for all datasets.

Table 5.4: Accuracy results of proposed MOVPS and other hybrid clustering algorithms

Dataset	Hybrid Clustering Algorithms									
	MBOA	ICSO	Chaotic TLBO	H-KHA	MEBBC	IKH	ICMPKHM	PSO-BB-BC	CBPSO	MOVPS
CMC	44.23	46.78	46.54	47.45	46.58	46.63	46.69	47.61	39.58	**64.53**
LD	50.67	53.02	53.12	51.91	49.86	52.96	52.15	52.17	53.65	**91.26**
Thyroid	59.36	68.24	67.38	65.4	65.22	66.91	66.82	56.83	72.21	**93.09**
Iris	95.43	91.35	91.19	89.24	90.02	89.87	92.44	90.52	90.79	**96.28**
Glass	58.73	69.06	69.52	58.89	58.73	68.39	69.02	69.52	51.92	**73.81**
Wine	70.31	73.24	72.53	75	72.63	72.37	72.88	73.58	71.31	**78.43**
Vowel	56.92	65.28	64.91	66.98	59.21	61.76	59.65	60.18	51.72	**92.04**
Balance	71.95	78.61	81.04	75.42	68.42	76.42	77.42	89.21	76.62	**92.87**

Table 5.5: Intra-cluster distance results of proposed MOVPS and other hybrid clustering algorithms

Datasets	Hybrid Clustering Algorithms									
	MBOA	ICSO	Chaotic TLBO	H-KHA	MEBBC	IKH	ICMPKHM	PSO-BB-BC	CBPSO	MOVPS
CMC	5.21E+03	5.32E+03	5.53E+03	5.60E+03	5.53E+03	5.69E+03	5.70E+03	5.57E+03	5.54E+03	**5.05E+03**
LD	1.32E+03	**4.09E+02**	4.98E+02	3.14E+03	1.36E+03	3.11E+03	3.09E+03	9.98E+03	1.00E+04	9.24E+02
Thyroid	2.16E+03	9.90E+02	1.08E+03	1.80E+03	1.26E+03	1.77E+03	1.39E+03	9.82E+02	1.86E+03	**8.98E+02**
Iris	9.83E+01	9.57E+01	9.69E+01	9.65E+01	9.68E+01	9.71E+01	9.58E+01	9.60E+01	9.69E+01	**9.41E+01**

Glass	2.31E+02	2.26E+02	2.38E+02	2.16E+02	2.27E+02	2.23E+02	**1.99E+02**	2.19E+02	2.13E+02	2.05E+02
Wine	1.71E+04	1.69E+04	1.68E+04	1.66E+04	1.68E+04	1.65E+04	1.67E+04	1.63E+04	1.64E+04	**1.25E+04**
Vowel	1.61E+05	1.59E+05	1.55E+05	3.52E+05	1.57E+05	1.56E+05	**1.47E+05**	1.55E+05	1.51E+05	**1.47E+05**
Balance	5.96E+04	5.39E+04	5.36E+04	6.78E+04	5.83E+04	6.01E+04	6.26E+04	6.19E+04	6.20E+04	**4.92E+04**

Table 5.6: F-measure results of proposed MOVPS and other hybrid clustering algorithms

Dataset	Hybrid Clustering Algorithms								
	MBOA	ICSO	Chaotic TLBO	IKH	H-KHA	PSO-BB-BC	MEBBC	CBPSO	ICMPKHM	MOVPS
CMC	0.435	0.339	0.345	0.457	0.443	0.461	0.438	0.389	0.456	**0.614**
LD	0.498	0.528	0.517	0.519	0.512	0.506	0.483	0.534	0.521	**0.693**
Thyroid	0.576	0.668	0.698	0.658	0.634	0.549	0.649	0.704	0.641	**0.885**
Iris	0.79	0.784	0.786	0.783	0.787	0.784	0.782	0.787	0.791	**0.947**
Glass	0.574	0.427	0.434	0.459	0.462	0.471	0.476	0.421	0.466	**0.683**
Wine	0.524	0.526	0.528	0.543	0.546	0.528	0.532	0.526	0.558	**0.739**
Vowel	0.634	0.646	0.635	0.663	0.671	0.652	0.648	0.651	0.649	**0.692**
Balance	0.704	0.767	0.792	0.736	0.739	0.764	0.687	0.734	0.741	**0.816**

5.4.2.3 CONVERGENCE BEHAVIOUR OF MOVPS

The convergence behaviour of the proposed MOVPS algorithm and standard clustering algorithm is reported in Figure 5.2 (a-h). The X-axis denotes the number of iterations in each run, while Y-axis represents the intra-cluster distance. It is noticed that the proposed MOVPS algorithm converges on minimum intra-cluster distance with most datasets. For a few datasets, it is not converged on minimum intra-cluster distance, but it provides more stable results in early iterations. Hence, it is worth mentioning that the proposed MOVPS is a more competitive clustering algorithm than other clustering algorithms.

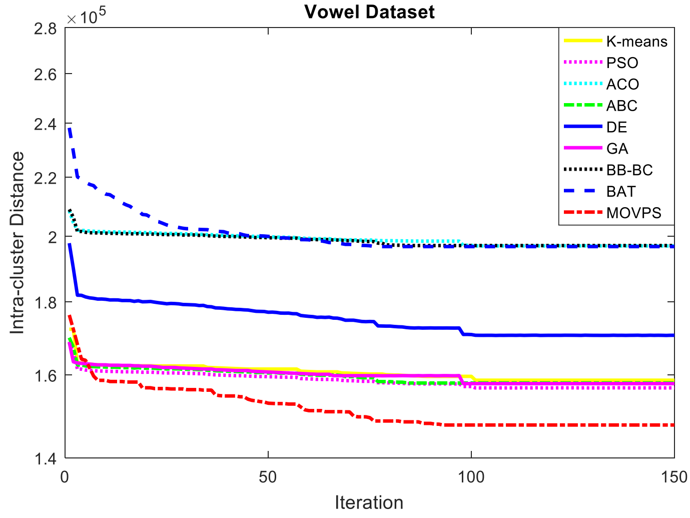


Figure 5.2 (a): Vowel dataset

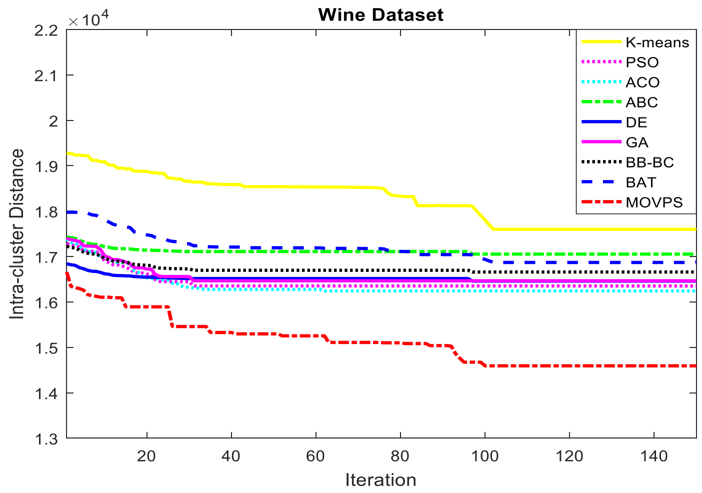


Figure 5.2 (b): Wine dataset

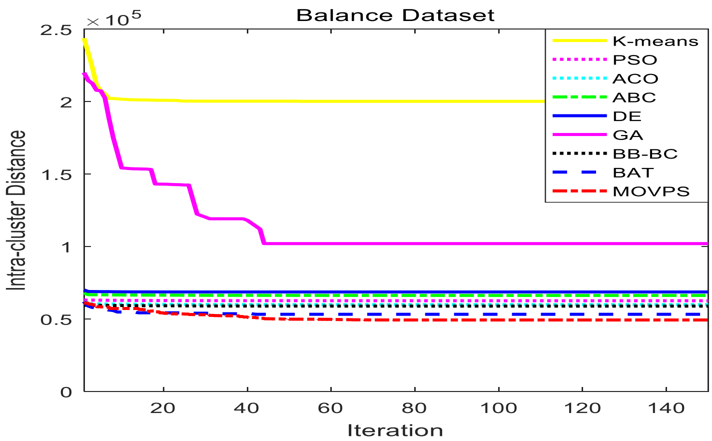


Figure 5.2 (c): Balance dataset

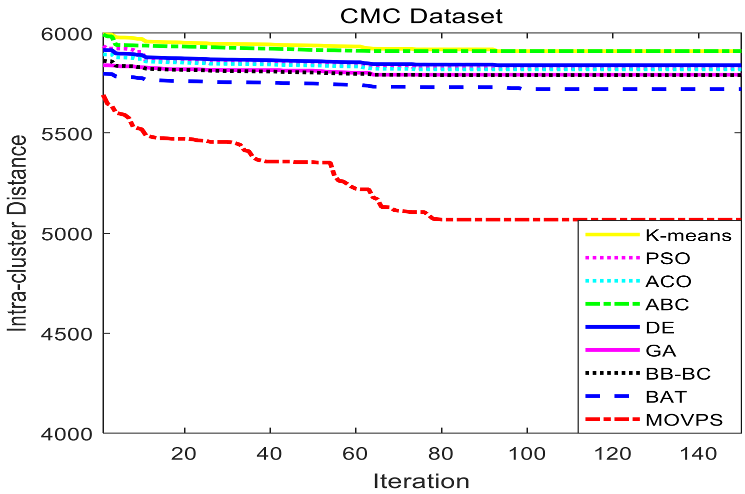


Figure 5.2 (d): CMC dataset

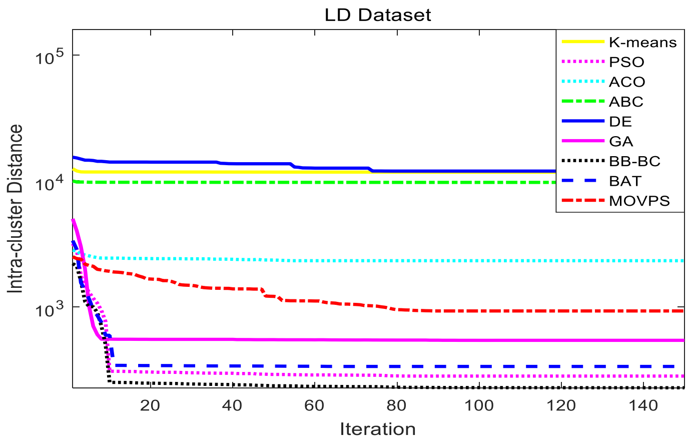


Figure 5.2 (e): LD dataset

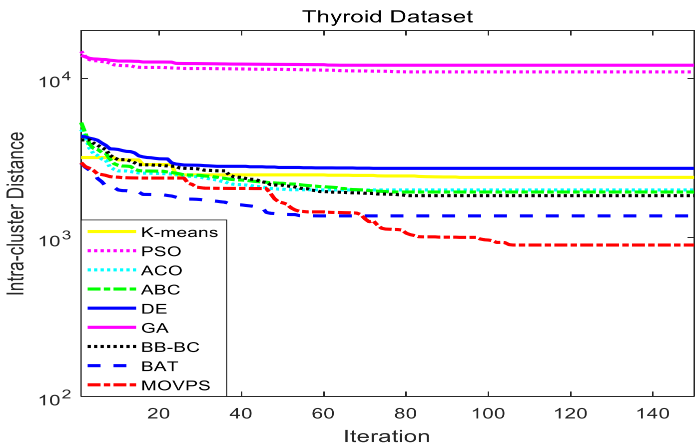


Figure 5.2 (f): Thyroid dataset

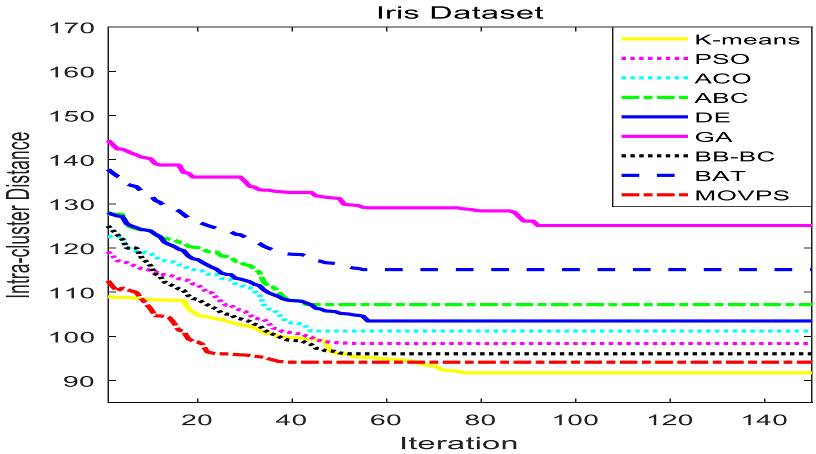


Figure 5.2 (g): Iris dataset

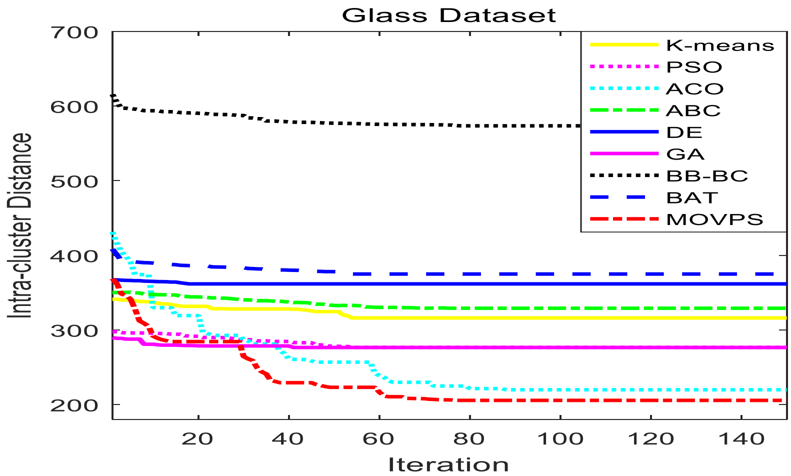


Figure 5.2 (h): Glass dataset

Figure 5.2 (a-h): Convergence behaviour of MOVPS and standard clustering algorithms based on intra-cluster distance

5.4.2.4 COMPARISON WITH PARETO FRONT RESULTS

This subsection presents the Pareto front results of the proposed MOVPS algorithm and other multi-objective clustering algorithms. The Pareto Front (PF) can be described as a set of all optimal solutions achieved by an algorithm, especially in multi-objective optimization. It was also stated earlier that the proposed MOVPS algorithm simultaneously optimizes two conflicting objectives, i.e. Connectedness and Intra-cluster distance. Hence, the PF is considered an effective performance measure to examine the efficacy of the proposed MOVPS compared to other multi-objective clustering algorithms. The results of Pareto-fronts of proposed MOVPS and different multi-objective clustering algorithms like TSMPSO, NSGA-II, MOPSO and MABC are illustrated in Figure 5.3 (a-h) using eight benchmark datasets. The PF results show that MOVPS is an efficient algorithm for finding optimal solutions compared to other multiobjective algorithms. The PFs is also validated the effectiveness and competence of the proposed MOVPS algorithm.

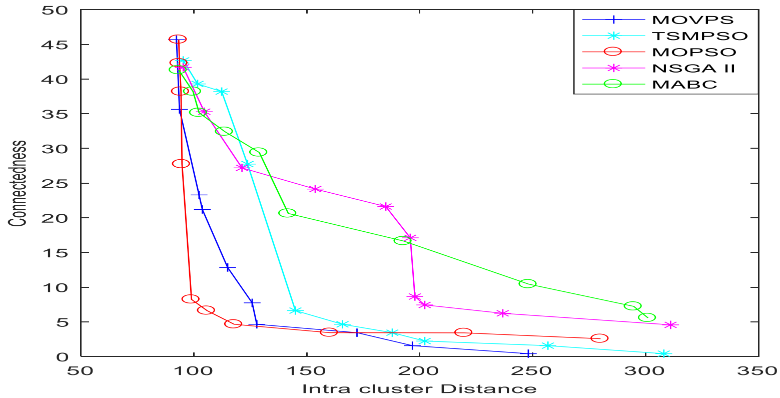


Figure 5.3 (a): Pareto front using Iris dataset

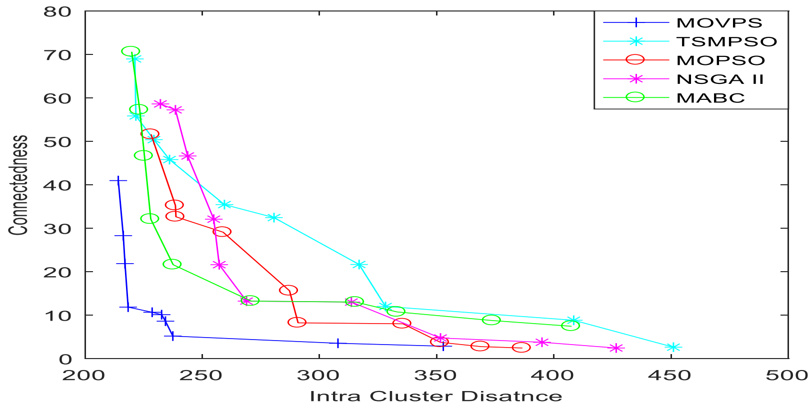


Figure 5.3 (b): Pareto front using Glass dataset

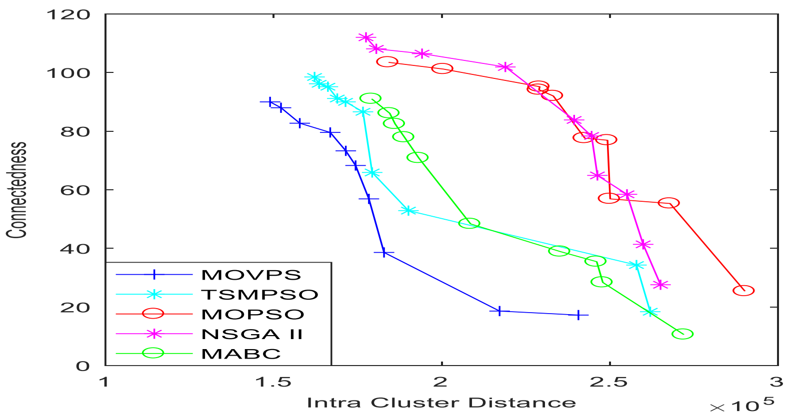


Figure 5.3 (c): Pareto front using Vowel dataset

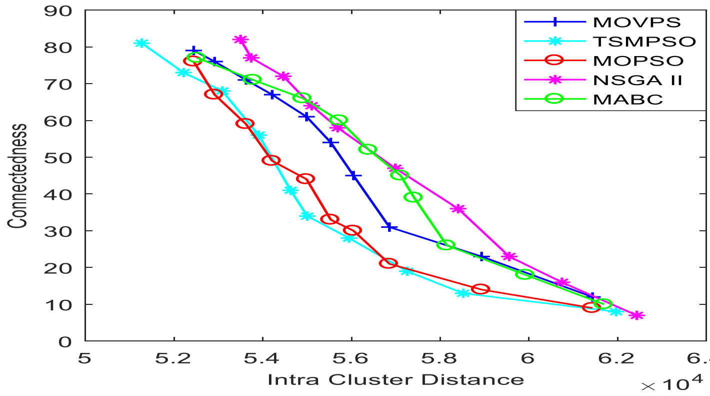


Figure 5.3 (d): Pareto front using Balance dataset

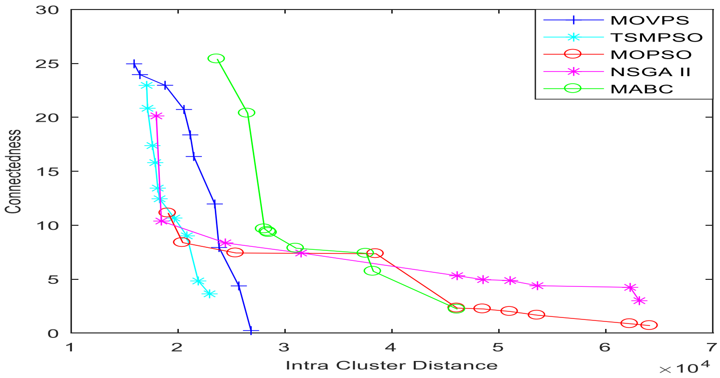


Figure 5.3(e): Pareto front using Wine dataset

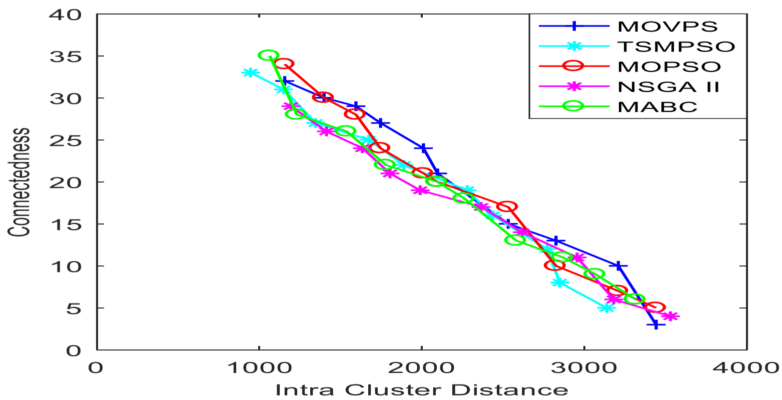


Figure 5.3 (f): Pareto front using Thyroid dataset

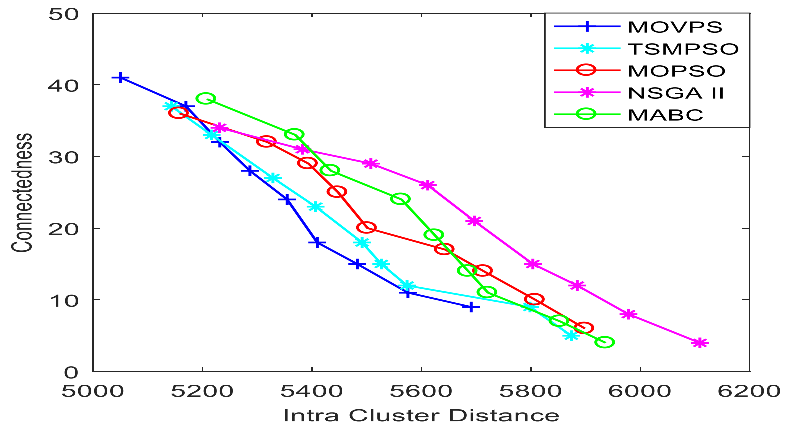


Figure 5.3(g): Pareto front using CMC dataset

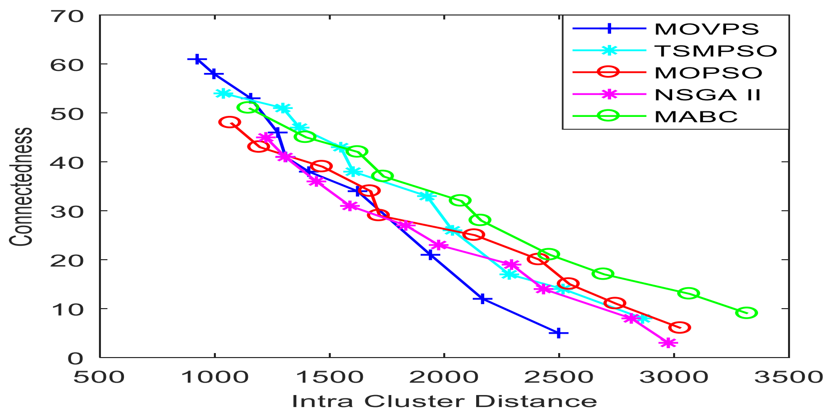


Figure 5.3(h): Pareto front using Liver Disorder dataset

Figure 5.3(a-h): Demonstrates Pareto-front of the non-dominating solution obtained through proposed MOVPS and other multiobjective algorithms

5.4 SUMMARY

This chapter presents a multiobjective vibrating particle system algorithm (MOVPS) for effective clustering. The MOVPS algorithm optimized two objective functions, namely, intra-cluster distance and connectedness, for optimal solutions. Also, the Pareto Front is considered for improving clustering results through MOVPS. A fitness function is also designed to assist in each particle's fitness evaluation. The fitness function helps in the assessment of the historical best position, bad particles, and good particles. Further, the best particle is considered in the particle's position updating. Eight standard datasets are used for the performance evaluation of MOVPS. The various performance measures for assessing results are accuracy, intra-cluster distance and f-measure. The experimental results of MOVPS are contrasted to single objective clustering algorithms consisting of standard and hybrid clustering algorithms. From the results, it is stated that MOVPS achieves high-quality results in comparison to single objective algorithms (standard and hybrid). For each multiobjective clustering algorithm, the Pareto solutions are also computed. These solutions also prove the capability of the proposed MOVPS algorithm for solving clustering problems. The MOVPS clustering algorithm achieves higher accuracy, an average of 16.95% and 17.01%. In contrast to standard and hybrid clustering algorithms, the results of the f-measure parameter have improved by an average of 12.1% and 15.6%. Also, minimum intra-cluster distance is achieved for most of the datasets. Therefore, it is stated that the proposed MOVPS algorithm is a robust and effective multiobjective clustering algorithm.

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.1 CONCLUSION

This thesis work presents three metaheuristic algorithms for solving the partitional clustering problems. This thesis work discusses the capabilities of single and multiobjective clustering algorithms for handling partitional clustering problems. In this thesis, two single-objective and one multiobjective clustering algorithm are developed for effective cluster analysis. Furthermore, this thesis is organized into six chapters. Chapter 1 gives a brief introduction to clustering problems. It also includes single-objective and multiobjective optimization problems. Further, the applicability of metaheuristic algorithms is also discussed in this chapter. It also highlights the motivation for the work and research objectives. Chapter 2 presents in-depth details on partitional clustering algorithms. It is divided into two sections- the first section describes the details of recent and popular single objective metaheuristic algorithms for cluster analysis, including its pros and cons. In contrast, the second section presents the multiobjective partitional clustering algorithm. It also discussed its capability, need, and reason for moving from single-objective to multi-objective clustering. Chapter 3 discusses an improved water wave optimization algorithm for data clustering. Before implementing the WWO algorithm, two modifications are inculcated into WWO to make it more capable and enhance the search in the optimal direction. These modifications are summarized as PSO-based solution search equation and decay operator. The first modification aims to improve the tracking of WWO in the optimal direction. In contrast, the second modification handles the premature convergence issue of WWO. A well-known benchmark clustering dataset is considered for evaluating the proposed IWWO algorithm performance. The results are compared with existing popular single-objective clustering algorithms regarding intra-cluster distance, accuracy and f-measure. The results showed that the proposed IWWO algorithm significantly improves the quality of clustering results.

Chapter 4 proposes an improved bat (IBAT) algorithm for effective data analysis. It is noticed that several performance issues are associated with the bat algorithm, such as population initialization, convergence rate and local optima. These issues significantly impact the outcome of the bat algorithm, especially with clustering problems. Hence, to get rid of the issues mentioned above in the bat algorithm, three improvements are proposed. These improvements are summarized as an enhanced cooperative co-evolution method, an elitist strategy and a Q-learning-based neighbourhood mechanism, respectively. The IBAT algorithm's performance is tested over a well-known clustering dataset and compared with several popular clustering algorithms. It is seen that the IBAT algorithm achieves more accurate results in terms of intra-cluster distance, accuracy and F-measure.

This thesis also highlights the efficacy of multiobjective clustering algorithms as it is noticed that sometimes single-objective clustering algorithms provide a biased solution because of one objective function. Hence, chapter 5 of this thesis presents a multiobjective vibrating particle system algorithm (MOVPS) for effectively clustering data. The problem of bias is handled through two objective functions. In multiobjective optimization, more than one objective function is designed to get the optimal solution for problems. It is also remembered that selected objective functions conflict with each other. Hence, for MOVPS, two conflicting objectives are designed-Euclidean distance and connectedness. Further, the performance of MOVPS is tested over a set of well-known clustering datasets. Simulation results are compared with several single and multi-objective clustering algorithms with many performance parameters. The simulation results stated that the proposed MOVPS provides state-of-the-art clustering results compared to single and multiobjective clustering algorithms. In short, this thesis work presents two single-objective clustering algorithms (IWWO and IBAT) and one multiobjective clustering algorithm (MOVPS) for effective data clustering. The simulation results analysis found that the MOVPS clustering algorithm provides better clustering results than IWWO and IBAT because of two objective functions. These functions are also problem depended and can be optimized simultaneously. It is also seen that single-objective algorithms exhibit biased solutions for a few datasets and can get rid of multiobjective optimization. In the context of single objective algorithms, it is stated that the overall performance of the IBAT algorithm is superior to the IWWO algorithm. The performance of the IBAT algorithm is also improved due to the inclusion of a neighbourhood search mechanism. It is also concluded that neighbourhood strategy can significantly improve the clustering results.

6.2 FUTURE SCOPE

This research work focused on partitional clustering problems. In future, the work will be extended to solving other clustering problems like model-based clustering, grid-based clustering etc. In future, the proposed clustering techniques will be implemented on the real-life datasets. The sensitivity analysis is also an important measure for evaluating the performance of the meta-heuristic algorithm. It will be utilized for the evaluation of meta-heuristic algorithms in future work. It is also observed that multiobjective clustering algorithms provide superior results than single-objective clustering algorithms. So, in future, more metaheuristic algorithms will be explored for multiobjective clustering. One of the challenging issues in the case of a partitional clustering problem is prior knowledge of the number of clusters. This issue of clustering will be addressed through dynamic clustering in the future, which computes the number of clusters from the dataset automatically. Moreover, effective neighbourhood strategies will be designed to get optimal solutions. Future work can also consider graph clustering and parallel clustering method.

REFERENCES

1. Han J., Kamber M. and Pei J., "*Data mining: concepts and techniques: concepts and techniques"*, Elsevier, 2011.
2. Fayyad U., Piatetsky G. S. and Smyth P., "*From data mining to knowledge discovery in databases"*, AI magazine, Vol. 17, No. 3, pp. 37-54, 1996.
3. Dasgupta S., *"The hardness of k-means clustering".* Technical Report CS2007-0890, University of California, San Diego, 2007.
4. Jain A. K. and Dubes R. C., *"Algorithms for Clustering Data",* Prentice-Hall, Englewood cliffs, 1988.
5. Jain A. K., *"Data clustering: 50 years beyond K-means",* Pattern Recogn. Letter, Vol. 31, pp. 651-66, 2010.
6. Hruschka E. R., Campello RJGB, Freitas A. A. and De Carvalho ACPLF, *"A survey of evolutionary algorithms for clustering*", IEEE Trans. Syst. Man Cybern. Part C Appl. Rev., Vol. 39, No. 2, pp. 133–55, 2009.
7. Das S., Abraham A. and Konar A. *"Meta-heuristic Clustering",* Springer, 2009.
8. Basu S., Davidson I. and Wagstaff K., *"Constrained Clustering: Advances in Algorithms, Theory and Applications"*, Data Mining and Knowledge Discovery, Chapman and Hall/CRC, 2008.
9. Abraham A., Das S. and Roy S., "*Swarm intelligence algorithms for data clustering",* Soft Computing for Knowledge Discovery and Data Mining, Part IV, Springer, pp. 79–313, 2007.
10. Roux, M. *"A comparative study of divisive and agglomerative hierarchical clustering algorithms".* Journal of Classification, Vol. 35, No. 2, pp. 345-366, 2018.
11. Seifoddini, H. K., *"Single linkage versus average linkage clustering in machine cells formation applications"*, Computers & Industrial Engineering, Vol. 16, No. 3, 419-426, 1989.
12. Frades I., and Matthiesen R. *"Overview on techniques in cluster analysis".* Bioinformatics methods in clinical research, pp. 81-107,2010.
13. Guha S., Rastogi R., and Shim K., *"Cure: an efficient clustering algorithm for large databases"*, Information systems, Vol. *26*, No. 1, pp. 35-58, 2001.
14. Zhang T., Ramakrishnan R., and Livny M. *"BIRCH: A new data clustering algorithm and its applications*". Data mining and knowledge discovery, Vol. 1, No. 2, pp. 141-182, 1997.

15. Ester M ., Kriegel H. P., Sander J. and Xu X., *"A density based algorithm for discovering clusters in large spatial databases",* Proceedings of International Conference on Knowledge Discovery and Data Mining (KDD'96), Portland, OR, pp. 226–231, 1996.
16. Ankerst M., Breunig M., Kriegel H. P. and Sander J., *"OPTICS: ordering points to identify the clustering structure"*, Proceedings of the ACM-SIGMOD International Conference on Management of Data, Philadelphia, PA, pp. 49–60, 1999.
17. McLachlan G. and Krishnan T., *"The EM Algorithm and Extensions"*. New York: Wiley, 1997.
18. Wang W., Yang J. and Muntz R., "STING: a statistical information grid approach to spatial data mining", Proceedings of the International Conference on Very Large Data Base (VLDB'97), Athens, Greek, pp. 186–195, 1997.
19. Glover F., *"Future paths for integer programming and links to artificial intelligence"*, Computers & operations research, Vol. 13, No. 5, pp. 533-549, 1986.
20. Das S., Abraham A., and Konar A., "*Metaheuristic clustering"*, Vol. 178, Springer, 2009.
21. Gunantara N., *"A review of multi-objective optimization: Methods and its applications"*, Cogent Engineering, Vol. *5*, No. 1, 1502242, 2018.
22. Thiele L., Miettinen K., Korhonen P. J., and Molina, J., *"A preference-based evolutionary algorithm for multi-objective optimization",* Evolutionary computation, Vol. 17, No. 3, pp. 411-436, 2009.
23. Deb K.,"Multi-objective optimization", In Search methodologies, pp. 403-449, 2014, Springer, Boston, MA.
24. Gong S., Hu W., Li H., and Qu Y., *"Property Clustering in Linked Data: An Empirical Study and Its Application to Entity Browsing",* International Journal on Semantic Web and Information Systems (IJSWIS), Vol. 14, No. 1, pp. 31-70, 2018.
25. Saini S. K., Dhull V., Singh S., and Sharma A., "*Visual exploratory data analysis of covid-19 pandemic*", In 2020 5th IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE), pp. 1-6, IEEE, December 2020.
26. Holý V., Sokol O., and Černý M., *"Clustering retail products based on customer behaviour"*, Applied Soft Computing, Vol. 60, pp. 752-762, 2017.
27. Javed M. A., Zeadally S., and Hamida E. B., *"Data analytics for cooperative intelligent transport systems"*, Vehicular communications, Vol. 15, pp. 63-72, 2019.

28. Chen Y., Wang L., Li F., Du B., Choo K. K. R., Hassan H., and Qin W, "Air quality data clustering using EPLS method", Information Fusion, Vol. 36, pp. 225-232, 2017.
29. Gupta M. K., Govil M. C., and Singh G., *"Text-mining and pattern-matching based prediction models for detecting vulnerable files in web applications"*, Journal of Web Engineering, pp. 028-044, 2018.
30. Keserwani P. K., Govil M. C., and Pilli E. S., *"The web ad-click fraud detection approach for supporting to the online advertising system"*, International Journal of Swarm Intelligence, Vol. 7, No. 1, pp. 3-24, 2022.
31. Navarro Á. A. M., and Ger P. M., *"Comparison of clustering algorithms for learning analytics with educational datasets"*, IJIMAI, Vol. 5, No. 2, pp. 9-16, 2018.
32. Mekhmoukh A., and Mokrani K., *"Improved Fuzzy C-Means based Particle Swarm Optimization (PSO) initialization and outlier rejection with level set methods for MR brain image segmentation"*, Computer methods and programs in biomedicine, Vol. 122, No. 2, pp. 266-281, 2015.
33. Triguero I., del Río S., López V., Bacardit J., Benítez J. M., and Herrera F., *"ROSEFW-RF: the winner algorithm for the ECBDL'14 big data competition: an extremely imbalanced big data bioinformatics problem"*, Knowledge-Based Systems, Vol. 87, pp. 69-79, 2015.
34. Abualigah L. M., Khader A. T., Al-Betar M. A., and Alomari O. A., *"Text feature selection with a robust weight scheme and dynamic dimension reduction to text document clustering"*, Expert Systems with Applications, Vol. 84, pp. 24-36, 2017.
35. Marinakis Y., Marinaki M., Doumpos M., and Zopounidis C., *"Ant colony and particle swarm optimization for financial classification problems"*, Expert Systems with Applications, Vol. 36, No. 7, pp.10604-10611, 2009.
36. Chou C. H., Hsieh S. C., and Qiu C. J., *"Hybrid genetic algorithm and fuzzy clustering for bankruptcy prediction",* Applied Soft Computing, Vol. 56, pp. 298-316, 2017.
37. Kuo R. J. and Lin L. M., *"Application of a hybrid of genetic algorithm and particle swarm optimization algorithm for order clustering"*, Decision Support Systems, Vol. 49, pp. 451–62, 2010.
38. Gupta S., and Deep K., "*A memory-based grey wolf optimizer for global optimization tasks*". Applied Soft Computing, *93*, 106367, 2020.
39. Shelokar P. S., Jayaraman V. K. and Kulkarni B. D., *"An ant colony approach for clustering"*, Analytica Chimica Acta, Vol. 509, No. 2, pp. 187-95, 2004.

40. Chen C. Y. and Ye F., *"Particle swarm optimization algorithm and its application to clustering analysis"*. IEEE International Conference on Networking, Sensing and Control, Vol. 2, pp. 789-794, 2004.
41. Satapathy S. C. and Naik A., *"Data clustering based on teaching-learning-based optimization"*, In Swarm, Evolutionary, and Memetic Computing, Springer Berlin Heidelberg, pp 148-56, 2011.
42. Karaboga D. and Ozturk C., *"A novel clustering approach: Artificial Bee Colony (ABC) algorithm"*. Applied Soft Computing, Vol. 11, pp. 652–57, 2011.
43. Singh A., and Deep K., "Artificial Bee Colony algorithm with improved search mechanism. *Soft Computing"*, Vol. *23*, No. 23, pp. 12437-12460, 2019.
44. Santosa B. and Ningrum M. K., *"Cat swarm optimization for clustering"*. In International Conference on Soft Computing and Pattern Recognition (SOCPAR'09), pp. 54-59, 2009.
45. Mullen R. J., Dorothy M., Sarah B. and Paolo R., *"A review of ant algorithms"*, Expert Systems with Applications, Vol. 36, No. 6, pp. 9608-9617, 2009.
46. Karaboga D. and Basturk B., *"A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm"*, J. Global Optim., Vol. 39, No. 3, pp. 459–471, 2007.
47. Sarafrazi S., Nezamabadi H. P. and Saryazdi S., *"Disruption: A new operator in gravitational search algorithm"*, Scien-tia Iranica D, Vol. 18, No. 3, pp. 539–548, 2011.
48. Mukhopadhyay A., Maulik U., and Bandyopadhyay S., A survey of multiobjective evolutionary clustering. *ACM Computing Surveys (CSUR)*, Vol. 47, No. 4, pp. 1-46, 2015.
49. Savsani V., and Tawhid M. A., "*Non-dominated sorting moth flame optimization (NS-MFO) for multi-objective problems"*, Engineering Applications of Artificial Intelligence, Vol. 63, pp. 20-32, 2017.
50. Alok A. K., Saha S., and Ekbal A., *"A new semi-supervised clustering technique using multi-objective optimization"*. Applied Intelligence, Vol. *43*, No. 3, pp. 633-661, 2015.
51. Kumar Y., and Singh, P. K., *"A chaotic teaching learning based optimization algorithm for clustering problems".* Applied Intelligence, Vol 49, No. 3, pp. 1036-1062, 2019.
52. Bouyer, A., and Hatamlou, A. "*An efficient hybrid clustering method based on improved cuckoo optimization and modified particle swarm optimization algorithms"*, Applied Soft Computing, Vol. 67, pp. 172-182, 2018.

53. Hatamlou A., *"A hybrid bio-inspired algorithm and its application"*. Applied Intelligence, Vol. 47, No. 4, pp. 1059-1067, 2017.
54. Hatamlou A., *"Black hole: A new heuristic optimization approach for data clustering"*, Information Sciences, Vol. 222, pp. 175-184, 2013.
55. Kaveh A. and Talatahari S., *"A novel heuristic optimization method: charged system search"*. Acta Mechanica, Vol. 213, No. 3-4, pp. 267-289, 2010.
56. Erol O. K. and Eksin I., *"A new optimization method: big bang–big crunch"*. Advances in Engineering Software, Vol. 37, No. 2, pp. 106-111, 2006.
57. Kaveh A., Share AMAM. and Moslehi M., "*Magnetic charged system search: a new meta-heuristic algorithm for optimization"*, Acta Mechanica, Vol. 224, No. 1, pp. 85-07, 2013.
58. Wang R., Zhou Y., Qiao S., and Huang, K., *"Flower pollination algorithm with bee pollinator for cluster analysis"*. Information Processing Letters, Vol. *116*, No. 1, 1-14, 2016.
59. Pakrashi A., and Chaudhuri B. B., *"A Kalman filtering induced heuristic optimization based partitional data clustering"*. Information Sciences, Vol. 369, pp. 704-717, 2016.
60. Kang Q., Liu S., Zhou M., and Li, S.,*"A weight-incorporated similarity-based clustering ensemble method based on swarm intelligence"*. Knowledge-Based Systems, Vol. 104, pp. 156-164, 2016.
61. Abualigah L. M., Khader A. T., Hanandeh E. S., and Gandomi, A. H., *"A novel hybridization strategy for krill herd algorithm applied to clustering techniques"*. Applied Soft Computing, Vol. 60, pp. 423-435, 2017.
62. Zhou Y., Zhou Y., Luo Q., and Abdel-Basset M., *"A simplex method-based social spider optimization algorithm for clustering analysis"*. Engineering Applications of Artificial Intelligence, Vol 64, pp. 67-82, 2017.
63. Nayak J., Naik B., Behera H. S., and Abraham A., *"Hybrid chemical reaction based metaheuristic with fuzzy c-means algorithm for optimal cluster analysis"*. Expert Systems with Applications, Vol. 79, pp. 282-295, 2017.
64. Han X., Quan L., Xiong X., Almeter M., Xiang J., and Lan Y., *"A novel data clustering algorithm based on modified gravitational search algorithm"*. Engineering Applications of Artificial Intelligence, Vol. *61*, pp. 1-7, 2017.

65. Queiroga E., Subramanian A., and Lucídio dos Anjos F. C., *"Continuous greedy randomized adaptive search procedure for data clustering",* Applied soft computing, Vol. 72, pp. 43-55, 2018.
66. Bijari K., Zare H., Veisi H., & Bobarshad H., *"Memory-enriched big bang–big crunch optimization algorithm for data clustering".* Neural Computing and Applications, *29*(6), 111-121, 2018.
67. Kumar Y., and Singh P. K. *"Improved cat swarm optimization algorithm for solving global optimization problems and its application to clustering".* Applied Intelligence, Vol. 48, No. 9, pp. 2681-2697, 2018.
68. Das P., Das D. K., and Dey S., *"A new class topper optimization algorithm with an application to data clustering".* IEEE Transactions on Emerging Topics in Computing, Vol. 8, No. 4, pp. 948-959, 2018.
69. Deb S., Tian Z., Fong S., Wong R., Millham R., and Wong K. K., *"Elephant search algorithm applied to data clustering"*, Soft Computing, Vol. 22, No. 18, pp. 6035-6046, 2018.
70. Tsai C. W., Chang W. Y., Wang Y. C., and Chen H. *"A high-performance parallel coral reef optimization for data clustering".* Soft Computing, Vol. 23, No. 19, pp. 9327-9340, 2019.
71. Sharma M., and Chhabra J. K., *"An efficient hybrid PSO polygamous crossover-based clustering algorithm"*. Evolutionary Intelligence, 1-19, 2019.
72. Abdulwahab H. A., Noraziah A., Alsewari A. A., and Salih S. Q., *"An enhanced version of black hole algorithm via levy flight for optimization and data clustering problems"*, IEEE Access, Vol. 7, pp. 142085-142096, 2019.
73. Mustafa H. M., Ayob M., Nazri M. Z. A., and Kendall G., *"An improved adaptive memetic differential evolution optimization algorithm for data clustering problems"*, PloS one, Vol. 14, No. 5, e0216906, 2019.
74. Tarkhaneh O., and Moser I. *"An improved differential evolution algorithm using Archimedean spiral and neighborhood search-based mutation approach for cluster analysis"*. Future Generation Computer Systems, Vol. 101, pp. 921-939, 2019.
75. Agbaje M. B., Ezugwu A. E., and Els, R., *"Automatic data clustering using hybrid firefly particle swarm optimization algorithm"*, IEEE Access, 7, 184963-184984, 2p19.
76. Kushwaha N., Pant M., and Sharma S., *"Electromagnetic optimization-based clustering algorithm"*, Expert Systems, e12491, 2019.

77. Senthilnath J., Kulkarni S., Suresh S., Yang X. S., and Benediktsson J. A., *"FPA clust: evaluation of the flower pollination algorithm for data clustering"*. Evolutionary Intelligence, 1-11, 2019.
78. Mageshkumar C., Karthik S., and Arunachalam V. P., *"Hybrid metaheuristic algorithm for improving the efficiency of data clustering"*. Cluster Computing, Vol. 22, No. 1, pp. 435-442, 2019.
79. Xie H., Zhang L., Lim C. P., Yu Y., Liu C., Liu H., and Walters J., *"Improving K-means clustering with enhanced Firefly Algorithms"*, Applied Soft Computing, Vol. 84, 105763, 2019.
80. Huang K. W., Wu Z. X., Peng H. W., Tsai M. C., Hung Y. C., and Lu Y. C. *"Memetic particle gravitation optimization algorithm for solving clustering problems"*. IEEE Access, Vol. 7, pp. 80950-80968, 2019.
81. Dinkar S. K., and Deep K., *"Opposition-based antlion optimizer using Cauchy distribution and its application to data clustering problem"*, Neural Computing and Applications, 1-29, 2019.
82. Kuwil F. H., Shaar F., Topcu A. E., and Murtagh F., *"A new data clustering algorithm based on critical distance methodology",* Expert Systems with Applications, Vol. 129, pp. 296-310, 2019.
83. Singh H., Kumar Y., and Kumar S., *"A new meta-heuristic algorithm based on chemical reactions for partitional clustering problems"*, Evolutionary Intelligence, Vol. 12, No. 2, pp. 241-252, 2019.
84. Baalamurugan K. M., and Bhanu S. V., *"An efficient clustering scheme for cloud computing problems using metaheuristic algorithms",* Cluster Computing, Vol. 22, No. 5, pp. 12917-12927, 2019.
85. Singh T., *"A chaotic sequence-guided Harris hawks optimizer for data clustering"*, Neural Computing and Applications, Vol. 32, No. 23, pp. 17789-17803, 2020.
86. Aljarah I., Mafarja M., Heidari A. A., Faris H., and Mirjalili S., *"Clustering analysis using a novel locality-informed grey wolf-inspired clustering approach*", Knowledge and Information Systems, Vol. 62, No. 2, pp. 507-539, 2020.
87. Zhu L. F., Wang J. S., Wang H. Y., Guo S. S., Guo M. W., and Xie W. *"Data Clustering Method Based on Improved Bat Algorithm with Six Convergence Factors and Local Search Operators"*, IEEE Access, Vol. 8, pp. 80536-80560, 2020.

88. Kaur A., Pal S. K., and Singh A. P., *"Hybridization of Chaos and Flower Pollination Algorithm over K-Means for data clustering*", Applied Soft Computing, Vol. 97, 105523, 2020.
89. İnkaya T., Kayalıgil S., and Özdemirel N. E. "*Ant colony optimization based clustering methodology"*, Applied Soft Computing, Vol. 28, pp. 301-311, 2015.
90. Prakash J., and Singh P. K., *"An effective multiobjective approach for hard partitional clustering"*, Memetic Computing, Vol. 7, No. 2, pp. 93-104, 2015.
91. Kishor A., Singh P. K., and Prakash, J., *"NSABC: Non-dominated sorting based multi-objective artificial bee colony algorithm and its application in data clustering"*, Neurocomputing, Vol. *216*, pp. 514-533, 2016.
92. Armano, G., and Farmani, M. R., *"Multiobjective clustering analysis using particle swarm optimization"*, Expert Systems with Applications, Vol. 55, pp. 184-193, 2016.
93. Garza-Fabre M., Handl J., and Knowles J., *"An improved and more scalable evolutionary approach to multiobjective clustering"*, IEEE Transactions on Evolutionary Computation, Vol. 22, No. 4, pp. 515-535, 2017.
94. Martínez-Peñaloza M. G., Mezura-Montes E., Cruz-Ramírez N., Acosta-Mesa H. G., and Ríos-Figueroa H. V., *"Improved multi-objective clustering with automatic determination of the number of clusters"*, Neural Computing and Applications, Vol. 28, No. 8, pp. 2255-2275, 2017.
95. Shahsamandi Esfahani P., and Saghaei A., *"A multi-objective approach to fuzzy clustering using ITLBO algorithm"*, Journal of AI and Data Mining, Vol.5, No.2, pp. 307-317, 2017.
96. Zhou Z., and Zhu S., *"Kernel-based multiobjective clustering algorithm with automatic attribute weighting"*, Soft Computing, Vol. 22, No.11, pp.3685-3709, 2018.
97. Paul A. K., and Shill P. C., *"New automatic fuzzy relational clustering algorithms using multi-objective NSGA-II",* Information Sciences, Vol. 448, pp. 112-133, 2018.
98. Liu C., Liu J., Peng D., and Wu C., *"A general multiobjective clustering approach based on multiple distance measures"*. IEEE Access, Vol. 6, pp. 41706-41719, 2018.
99. Wang R., Lai S., Wu G., Xing L., Wang L., and Ishibuchi H. *"Multi-clustering via evolutionary multi-objective optimization"*, Information Sciences, Vol. 450, pp.128-140, 2018.
100. Liu C., Li Y., Zhao Q., and Liu C. *"Reference vector-based multi-objective clustering for high-dimensional data"*, Applied Soft Computing, Vol. 78, pp. 614-629, 2019.

101. Prakash J., and Singh P. K., *"Gravitational search algorithm and K-means for simultaneous feature selection and data clustering: a multi-objective approach"*, Soft Computing, Vol. 23, No. 6, pp. 2083-2100, 2019.

102. Zang W., Wang Z., Jiang D., and Liu X., *"A kernel-based intuitionistic fuzzy C-means clustering using improved multi-objective immune algorithm",* IEEE Access, Vol. 7, pp. 84565-84579, 2019.

103. Prakash J., Singh P. K., and Kishor A., *"Integrating fitness predator optimizer with multi-objective PSO for dynamic partitional clustering"*, Progress in Artificial Intelligence, Vol. 8, No.1, pp. 83-99, 2019.

104. Antunes V., Sakata T. C., Faceli K., and de Souto M. C., *"Hybrid strategy for selecting compact set of clustering partitions"*, Applied Soft Computing, Vol. 87, 105971, 2020.

105. Kuo R. J., and Zulvia F. E., *"Multi-objective cluster analysis using a gradient evolution algorithm"*, Soft Computing, pp.1-15, 2020.

106. Liu R., Ren R., Liu J., and Liu J., *"A clustering and dimensionality reduction based evolutionary algorithm for large-scale multi-objective problems"*, Applied Soft Computing, Vol. 89, 106120, 2020.

107. Hancer E., *"A new multi-objective differential evolution approach for simultaneous clustering and feature selection",* Engineering Applications of Artificial Intelligence, Vol. 87, 103307, 2020.

108. Yan D., Cao H., Yu Y., Wang Y., and Yu X., *"Single-Objective/Multiobjective Cat Swarm Optimization Clustering Analysis for Data Partition"*, IEEE Transactions on Automation Science and Engineering, 2020.

109. Chen L., Duan H., Fan Y., and Wei C., *"Multi-objective clustering analysis via combinatorial pigeon inspired optimization"*. Science China Technological Sciences, 1-12, 2020.

110. Zhu S., Xu L., and Goodman E. D., *"Evolutionary multi-objective automatic clustering enhanced with quality metrics and ensemble strategy"*, Knowledge-Based Systems, 188, 105018, 2020.

111. Zheng Y. J., *"Water wave optimization: a new nature-inspired metaheuristic"*, Computers & Operations Research, Vol. 55, pp.1-11, 2015.

112. Soltanian A., Derakhshan F., and Soleimanpour-Moghadam M., "MWWO: Modified water wave optimization", 3rd Conference on Swarm Intelligence and Evolutionary Computation (CSIEC), pp. 1-5, IEEE, 2018, March.

113. Zhu G., and Kwong S., *"Gbest-guided artificial bee colony algorithm for numerical function optimization"*. Applied mathematics and computation, Vol. 217, No.7, pp. 3166-3173, 2010.
114. Kumar S., Yadav, D. K., and Khan, D. A., *" A novel approach to automate test data generation for data flow testing based on hybrid adaptive PSO-GA algorithm"* . International Journal of Advanced Intelligence Paradigms, *Vol. 9, No. 2-3,* pp. 278-312, 2017.
115. Hematabadi A. A., and Foroud, A. A., *"Optimizing the multi-objective bidding strategy using min–max technique and modified water wave optimization method"*, Neural Computing and Applications, Vol. 31, No. 9, 5207-5225, 2019.
116. Fister Jr I., Fister D., and Yang X. S., "A hybrid bat algorithm". arXiv preprint arXiv:1303.6310, 2013
117. Yilmaz S., and Kucuksille E. U., "Improved bat algorithm (IBA) on continuous optimization problems". Lecture Notes on Software Engineering, Vol. 1, No. 3, pp 279, 2013.
118. Senthilnath J, Kulkarni S, Benediktsson JA, and Yang XS, *"A novel approach for multispectral satellite image classification based on the bat algorithm",* IEEE Geoscience and Remote Sensing Letters. Vol. 13, No. 4, pp. 599-603, 2016.
119. Neelima S., Satyanarayana N., and Murthy P. K., *"Minimizing Frequent Itemsets Using Hybrid ABCBAT Algorithm".* In Data Engineering and Intelligent Computing, pp. 91-97, Springer, Singapore, 2018.
120. Ashish T., Kapil S., and Manju, B., *"Parallel bat algorithm-based clustering using mapreduce"*, In Networking Communication and Data Knowledge Engineering, pp. 73-82, Springer, Singapore, 2018.
121. Aboubi Y, Drias H, Kamel N, *"BAT-CLARA: BAT-inspired algorithm for Clustering LARge Applications"*, IFAC-PapersOnLine. Vol. 49, No. 12, pp. 243-248, 2016.
122. Fister I., Fong S., and Brest J., "A novel hybrid self-adaptive bat algorithm", The Scientific World Journal, 2014.
123. Zhao D., and He Y., *"Chaotic binary bat algorithm for analog test point selection"*. Analog Integrated Circuits and Signal Processing, Vol. 84, No. 2, pp. 201-214, 2015.
124. Yang Xin-She., *"A new metaheuristic bat-inspired algorithm"*, Nature inspired cooperative strategies for optimization (NICSO 2010), pp. 65-74, 2010, Springer.

125. Jiang B., and Wang N., *“Cooperative bare-bone particle swarm optimization for data clustering”*, Soft Computing, Vol. 18, No. 6, pp. 1079-1091, 2014.

126. Rahman Md Anisur, and Md Zahidul Islam, *"A hybrid clustering technique combining a novel genetic algorithm with K-Means"*, Knowledge-Based Systems, Vol. 71, pp. 345-365, 2014.

127. Liu Ruochen, et al., "*Gene transposon based clone selection algorithm for automatic clustering"*, Information Sciences, Vol. 204, pp.1-22, 2012.

128. Kumar Y. and Sahoo G., *“A two-step artificial bee colony algorithm for clustering”*, Neural Computing and Applications , Vol. 28, No. 3, pp. 537-55, 2017.

129. Cao Fuyuan, Jiye Liang, and Guang Jiang, *“An initialization method for the K-Means algorithm using neighborhood model”*, Computers & Mathematics with Applications Vol. 58, No. 3, pp. 474-483, 2009.

130. Maulik Ujjwal, and Sanghamitra Bandyopadhyay, *“Genetic algorithm-based clustering technique”*, Pattern recognition, Vol. 33, No. 9, pp. 1455-1465, 2000.

131. Kumar Y., and Sahoo G., *“An Improved Cat Swarm Optimization Algorithm Based on Opposition-Based Learning and Cauchy Operator for Clustering”*, JIPS, Vol. 13, No. 4, pp. 1000-1013, 2017.

132. Jensi R., and Wiselin Jiji G., *“An improved krill herd algorithm with global exploration capability for solving numerical function optimization problems and its application to data clustering”,* Applied Soft Computing, Vol. 46, pp. 230-245, 2016.

133. Watkins C. J., and Dayan P., “Q-learning. Machine learning”, Vol. 8, No. 3-4, pp. 279-292, 1992.

134. Kaveh A., and Ghazaan M. I. *“A new meta-heuristic algorithm: vibrating particles system”*, Scientia Iranica. Transaction A, Civil Engineering, Vol. 24, No. 2, pp. 551, 2017.

135. Kaveh A., Hoseini Vaez S. R., and Hosseini P., *“Enhanced vibrating particles system algorithm for damage identification of truss structures”*. Scientia Iranica, Vol 26, No. 1, 246-256, 2019.

136. Kaveh A., and Ghazaan M. I., *“Vibrating particles system algorithm for truss optimization with multiple natural frequency constraints”,* Acta Mechanica, Vol. 228, No. 1, pp. 307-322, 2017.

137. Cai J., Wei H., Yang H., and Zhao X., *“A novel clustering algorithm based on DPC and PSO”*, IEEE Access, Vol. 8, pp. 88200-88214, 2020.

138. Arora S., Singh S., and Yetilmezsoy K., *“A modified butterfly optimization algorithm for mechanical design optimization problems”*, Journal of the Brazilian Society of Mechanical Sciences and Engineering, Vol. 40, No.1, pp. 1-17, 2018.
139. Tripathi A. K., Sharma K., Bala M., Kumar A., Menon, V. G., and Bashir A. K., *“A parallel military-dog-based algorithm for clustering big data in cognitive industrial internet of things”*, IEEE Transactions on Industrial Informatics, Vol. *17*, No. 3, 2134-2142, 2020.
140. Li M. W., Wang Y. T., Geng J., and Hong W. C., *“Chaos cloud quantum bat hybrid optimization algorithm”*, Nonlinear Dynamics, Vol. 103, No. 1, pp. 1167-1193, 2021.

LIST OF PUBLICATIONS

SCI Listed

Published:

1) Kumar Y. and Kaur A., "*Variants of bat algorithm for solving partitional clustering problems*". *Engineering with Computers*, *38*(3), 1973-1999, 2022. Springer *(Impact Factor: 7.963)*
2) Kaur A. and Kumar Y., "*Neighborhood Search Based Improved Bat Algorithm for Data Clustering*". Applied Intelligence, Springer, pp, *1-35,* 2022. *(Impact Factor: 5.086)*
3) Kaur A. and Kumar Y., "*A Multi Objective Vibrating Particle System Algorithm for Data Clustering*". Pattern Analysis and Application, *25*(1), 209-239, 2022. Springer *(Impact Factor: 2.580)*

Communicated:

1) Kaur A. and Kumar Y., "*Evolutionary Approaches for Clustering Problems: A Survey*". Artificial Intelligence Review, Springer *(Paper Id: AIRE-D-21-00589-Under Review) (Impact Factor: 8.139)*

SCOPUS Listed

1) Kaur A. and Kumar Y., "*A new metaheuristic algorithm based on water wave optimization for data clustering*". Evolutionary Intelligence, Springer, Vol. 5, No. 1, pp. 759-783, 2021.
2) Kaur A. and Kumar Y., "*Healthcare Data Analysis Using Water Wave Optimization-Based Diagnostic Model*". Journal of Information and Communication Technology, Vol. 20, No. 4, pp. 457-488, 2021.
3) Kaur A. and Kumar Y., "*Water Wave Optimization Based Data Clustering Model*". In Journal of Physics: Conference Series, Vol. 1950, No. 1, p. 012054. IOP Publishing, August 2021.
4) Kaur A. and Kumar Y., "*Analyzing Healthcare Data Using Water Wave Optimization-Based Clustering Technique*". International Journal of Reliable and Quality E-Healthcare (IJRQEH), IGI Global, USA, Vol. 10, No. 4, pp. 38-57, 2021.

5) Kaur A. and Kumar Y., "*Recent Developments in Bat Algorithm: A Mini Review*". In Journal of Physics: Conference Series, Vol. 1950, No. 1, p. 012055, IOP Publishing, August 2021.

Presented in International Conference

1) Kaur A. and Kumar Y., "*Water Wave Optimization based data clustering*", 1st International Conference on Mechatronics and Artificial Intelligence. SGT University, Gurugram, Haryana. February 26th-27th, 2021.
2) Kaur A. and Kumar Y., "*Recent Developments in Bat algorithm: A Critical Mini Review*", 1st International Conference on Mechatronics and Artificial Intelligence. SGT University, Gurugram, Haryana. February 26th-27th, 2021.