Design New Meta Heuristic Algorithms for Partitional Clustering Problems

Thesis submitted in fulfillment of the requirements for the Degree of

DOCTOR OF PHILOSOPHY

By

HAKAM SINGH

Department of Computer Science Engineering and Information Technology

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY

Waknaghat, Solan-173234, Himachal Pradesh, INDIA

November, 2020

TABLE OF CONTENTS

SUPERVISOR’S CERTIFICATE

This is to certify that the work in the thesis entitled **“Design New Meta Heuristic Algorithms for Partitional Clustering Problems”** submitted by **Hakam Singh** is a record of an original research work carried out by him under my supervision and guidance in partial fulfillment of the requirements for the award of the degree of Doctor of Philosophy in Computer Science and Engineering in the Department of Computer Science and Engineering, **Jaypee University of Information Technology, Waknaghat, INDIA.** Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

Date:10/11/2020

Dr. Yugal Kumar

Assistant Professor (Senior Grade)

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology,

Waknaghat -173234, INDIA.

DECLARATION OF SCHOLAR

I hereby declare that the work reported in the Ph.D. thesis entitled **“Design New Meta Heuristic Algorithms for Partitional Clustering Problems”** submitted at **Jaypee University of Information Technology, Waknaghat, INDIA** is an authentic record of my work carried out under the supervision of **Dr. Yugal Kumar.** I have not submitted this work elsewhere for any other degree or diploma. I am fully responsible for the contents of my Ph.D Theses.

Hakam Singh

Hakam Singh
Department of Computer Science & Engineering and Information Technology
Jaypee University of Information Technology,
Waknaghat -173234, INDIA.
Date: 10/11/2020

ACKNOWLEDGEMENTS

First of all, I would like to express my deep sense of respect and gratitude towards my guide Dr. Yugal Kumar in the Department of Computer Science and Engineering and Information Technology, for their support throughout this research work. I want to thank them for introducing me to the field of Data Mining and giving me the opportunity to work under them. Without their invaluable advice and assistance, it would not have been possible for me to complete this thesis. I am greatly indebted to them for their 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 wonderful persons.

I would like to express my gratitude to our Honorable Vice Chancellor Prof. (Dr.) Vinod Kumar and Director & Academic Head and Head Department of CSE/IT Prof. (Dr.) Samir Dev Gupta to promote the research and facilitate resources in the institution. I would also like to thank my doctoral committee members Dr. Vivek Kumar Sehgal, Dr. Kapil Sharma and Dr. Dheeraj 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. In particular, I thank, Ravi Raina and all staff at the Department of Computer Science and Engineering and Information Technology, JUIT, Waknaghat who has been extremely helpful at numerous occasions. I thank my fellow PhD friends for their consistent help and valuable discussion. Last but not the least, I would like to thank my family my parents, for giving birth to me at the first place and supporting me spiritually throughout my life.

.

Hakam Singh

ABSTRACT

In this competitive business world, data mining is an essential aspect of the knowledge discovery process. Clustering is an important data analysis and well-recognized technique in the field of data mining. In clustering, there is no need for training the data. This technique adopts a distance measure to compute data objects in clusters. The implication of clustering techniques in different disciplines primes momentous research in this field. In the present time, clustering can get wide attention from the research community both of theoretical and practical point. It is seen that much work is reported on clustering methods in literature; still, the clustering is an active area of research. This research work focuses on partitional clustering and its disciplines. The partitional clustering divides the data objects into different groups called clusters. Data objects within a cluster are similar in nature and exhibit heterogeneity with other clusters. There are several shortcomings associated with clustering methods like identification of initial cluster centers, stuck in local optima, lack of population diversification mechanism, imbalance exploration and exploitation processes, convergence rate and accuracy. This work addresses the local optima, diversity, convergence rate and exploration and exploitation issues. In this thesis, three algorithms are designed to address the issues as mentioned. An artificial chemical reaction optimization is explored to solve clustering problems. The artificial chemical reaction algorithm has proved its competency in clustering filed and provided good candidate solutions. To address the diversity and convergence rate issue, an improved version of the big bang big crunch algorithm is also proposed. The chaotic maps and cellular automata-based heat transfer methods are introduced in the big bang and big crunch algorithm. Further, to address the local optima and exploration and issue a neighborhood-based cat swarm optimization algorithm is also developed. The position and velocity search equation of the traditional cat swarm optimization algorithm are updated with global position and levy components. These modifications turn an efficient method for clustering.

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

ABC	Artificial bee colony
ACAKHM	Ant clustering algorithm with K-harmonic means
ACRO	Artificial chemical reaction optimization
ACO	Ant colony optimization
BA	Bat algorithm
BB-BC	Big bang big crunch
BFGSA	Bird flock gravitational search algorithm
BH	Black hole
BNMR	Blind naked mole rats
BPFPA	Flower pollination algorithm with bee pollinator
BS	Binary search
CA	Cellular automata
CC	Cooperative co-evolution
CDC	Counts of dimension to change
C-FPA	Chaos and flower pollination algorithm
COM	Center of mass
CSA	Crow search algorithm
CS-KM	Cuckoo search
CSO	Cat swarm optimization
CSS	Charged system search
DE	Differential evolution
ESA	Elephant search clustering algorithm
FA	Firefly algorithm

GA	Genetic algorithm
GAMS	GA with message-based similarity
GCC	Gravitational center clustering
GOA	Grasshopper optimization algorithm
GSA	Gravitational search algorithm
GTCSA	Gene transposon based clone selection algorithm
HACO-PSO	Hybridized artificial bee colony particle swarm optimization
HCSDE	Hybrid cuckoo search and differential evolution algorithm
HGCA	Harmonious genetic clustering algorithm
HKA-K	Heuristic Kalman algorithm and K-means
H-KHA	Krill herd with harmony search algorithm
HS	Harmony search
IBB-BC	Improved big bang big crunch
ICMPKHM	Improved cuckoo search and particle swarm optimization K-harmonic means
ICS	Improved cuckoo search
ICSO	Improved cat swarm optimization
ICSOKHM	Improved cat swarm optimization and K-harmonic means
IFPA	Improved flower pollination algorithm
IoT	Internet of Things
KDD	Knowledge discovery from data
KHA	Heuristic Kalman filtering algorithm
KHM	K-harmonic means
KICS	K-means and improved cuckoo search algorithm
KMQGA	K-means quantum genetic algorithm

K-MICA	K-means with modify imperialist competitive algorithm
K-NM-PSO	K-means Nelder–Mead simplex search and particle swarm optimization
LR	Letter recognition
MCSS	Magnetic charged system search
ME-BB–BC	Memory-enriched big bang–big crunch optimization algorithm
METACOC	Medoid-based clustering using ant colony optimization
OLAP	Online analytical processing
OLTP	Online transactional process
PSO	Particle swarm optimization
PSOHS	Particle swarm optimization with heuristic search
QCCS	Quantum chaotic cuckoo search
QPSO	Quantum particle swarm optimization
SA	Simulated annealing
SGKC	Standard genetic K-means clustering algorithm
SMP	Seeking memory pool
SMSSO	Simplex method-based social spider optimization algorithm
SRD	Seeking range of selected dimension
SRPSO	Self-regulating particle swarm optimization
SSO	Social spider optimization
TGCA	Two-stage genetic clustering algorithm
TLBO	Teaching learning-based optimization
TS	Tabu search
TSGA	Two stage genetic algorithm
WOA	Whale optimization algorithm

CHAPTER 1

INTRODUCTION

1. Introduction

Present time, large amount of data is gathered due digitization and technological revolution. These data are generated through online banking transactions, scientific experiments, satellite imagining, telecom, medical imagining, biology and space experiments, etc. The various web applications are also responsible for generation the digital data such amazon, twitter and so on. The data consists of hidden pattern and unseen information that can be used for taking various managerial and non-managerial decisions. Sometime, the growth of an organization also depends on the extractive information. On the other side, large amount of data are stored on relational database servers. It is also observed that online transactional process (OLTP) systems and online analytical processing (OLAP) systems are also designed for storing and processing of the collected data. Apart from these, various data repository are available on web for storing the data for analysis purpose. Hence, it is stated that some new algorithms can be developed for the analysis of data to extract unseen information and provide faster execution. The extracted information will be adopted for decision making and improve the human life. Knowledge discovery refers to the "nontrivial extraction of implicit, previously unknown and potentially useful information from databases" [1]. Data mining can also be described as KD process. Figure 1.1 demonstrates the KD process steps.

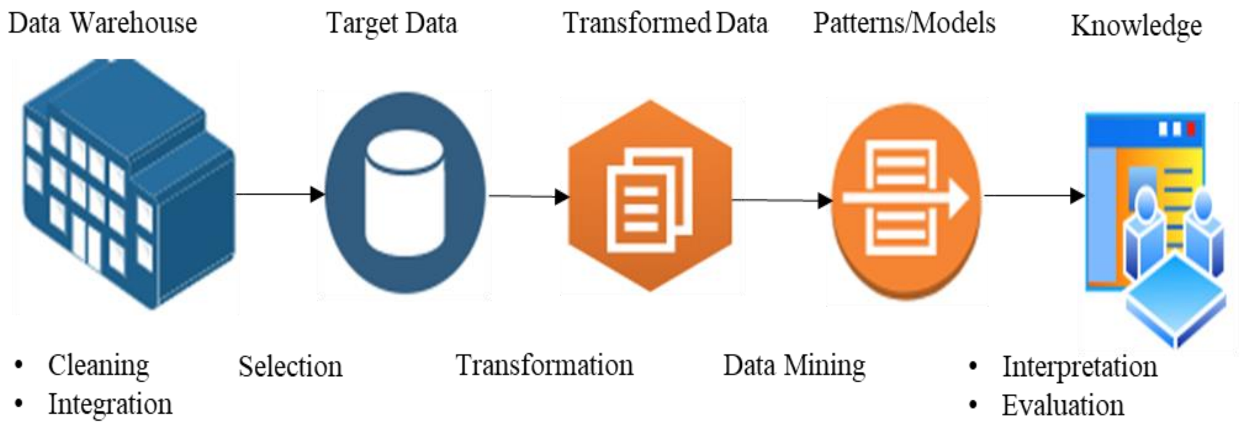

Figure 1.1: Knowledge discovery from data (KDD) process

1.2 Data Mining

It is the process for analyzing data and discovering the unseen pattern and information from the databases. It can be interpreted as a common platform of computer science, machine learning and statistics fields and aim of these fields is to work together for achieving common objective i.e. data analysis. The various techniques are associated with data mining task. These are described as feature selection, feature extraction, classification, prediction, clustering and association rules. The working of data mining task is described into three phases- Data Pre-processing, Pattern Recognition and Visualization.

The Data pre-processing responsible for converting the raw information into preprocessed information using various preprocessing techniques like missing value interpretation, data distribution, normalization etc. It is observed that sometimes raw information is incomplete, having errors and also inconsistent. The preprocessing methods are capable to deal such issues of raw information and transferred the raw information into piece of useful information from which undiscovered pattern will be determined.

The pattern recognition phase deals with the pre-processed data and responsible for identifying the previously undiscovered information. This phase also arrange the information in meaningful way. This phase consists of various techniques to determine unseen information such as prediction, rule mining, classification and clustering. It is also observed that the patterns are also depend on the problems being solved.

The Visualization phase represents the results of pattern recognition phase. The results are visualized either in tabular form or graphical manner. This phase also validates the results of aforementioned in pattern recognition phase by comparing some other techniques.

Furthermore, data mining is defined as either descriptive analysis or predictive analysis [2]. Predictive analysis can be interpreted as measuring the value of future variable using historical data, called as supervised learning/classification. The descriptive analysis corresponds for determining the unseen pattern based on dissimilarity criteria, called as unsupervised learning/clustering.

1.3 Clustering

Clustering is an unsupervised machine learning approach that divides data objects into different groups or clusters [3]. Clustering techniques can be classified as partitional, hierarchical, grid and density clustering [4-8]. In partitional clustering, dataset divides into disjoint clusters which are optimal in nature based on distance function [5]. A tree structured is designed in hierarchical clustering and clusters are described in term of tree nodes. It consists of two sub method i.e. agglomerative and divisive [6]. In grid clustering, the entire space divides into number of cells, called grid and subsequent grids are merged using dissimilarity measure to from clusters [7]. In density clustering, the clusters are described in terms of dense region separated through lower dense region [8]. It is seen that clustering algorithms had proven its efficacy in different research fields like stock market, medical analysis, identification of pattern, bio engineering etc., [9-13]. In recent time, numbers of meta-heuristic algorithms inspired from natural phenomenon have been reported for clustering. These natural phenomena's can be described as swarm intelligence, insect's behavior, well-defined laws of physics, and other natural processes of living beings. Some of these optimization techniques are ant colony optimization [14], particle swarm optimization [15], charged system search approach [16], artificial bee colony algorithm [17], magnetic optimization algorithm [18], black hole [19] and big bang big crunch algorithm [20]. These algorithms consist of some approximation function to determine the optimal solution. Further, it is noticed that local and global search mechanisms of meta-heuristic algorithms should be balanced for obtaining a good candidate solution. This research work focuses on partitional clustering and its disciplines. The partitional clustering divides the data objects into different groups called clusters. Euclidean distance is widely adopted similarity measure in partitional clustering [21-23]. This can be described as the sum of square root of difference between data objects and the cluster centers. This can be computed using equation 1.1.

$$D(X_i, C_j) = \sqrt{\sum_{k=1}^{d} (X_{ik}, C_{jk})^2} \tag{1.1}$$

Where the X_{ik} represents the data objects and C_{jk} represents the cluster centers.

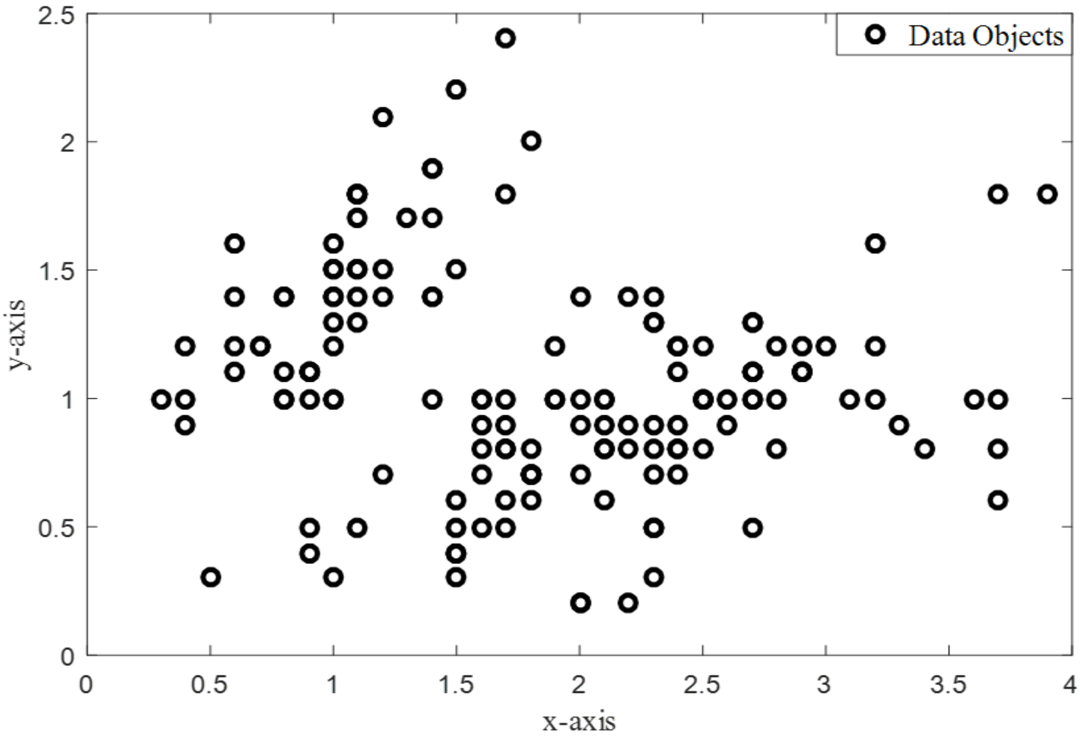

Figure 1.2: Distribution of data

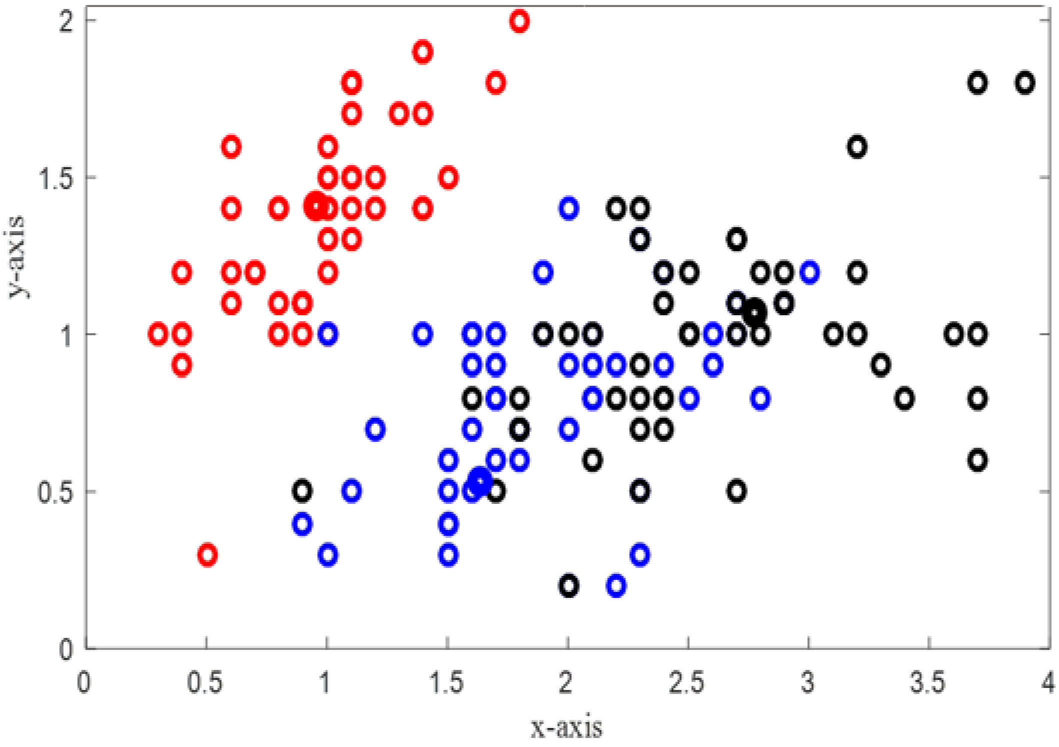

Figure 1.3: Partitional clustering

Figure 1.2 shows the distribution of data objects, whereas figure 1.3 illustrates the clustering of data object into clusters. The dataset is divided into three different clusters as cluster1 (red color), cluster2 (blue color) and cluster3 (black color). The data objects of cluster1 are linearly separable, and rest of are non-linear in nature.

1.4 Meta-Heuristic Techniques

The meta-heuristic algorithms are widely adopted for solving diverse optimization problems. These algorithms consist of more than one heuristic function for determining the optimal solutions. Furthermore, these algorithms having unique characteristics which are listed as

- To obtain near optimal solution instead of exact solution.
- Convergence of algorithm on optimal solution is not described using mathematical proof.
- Computationally less extensive as compared to exhaustive search.

The search mechanisms of meta-heuristic algorithms are modified initial candidate solution using various operations. Further, these algorithms are iterative in nature. But, these algorithm cannot generate optimal solution for every optimization problems. But, these algorithms are capable to address the complexities of optimization problems in efficient manner. It is also seen that these algorithms can be worked with different objective functions. These are main reasons behind the popularity of meta-heuristic algorithm in the research community. These algorithms are derived on the basis of biological, physical and natural principles and also consists of various operators and mechanism for generating the optimal solutions. The local search and global search methods are defined for exploring the search space to compute candidate solutions. The local search and global search can be interpreted as exploration and exploitation. It is also observed that some meta-heuristic algorithms are more explorative than exploitive. These algorithms can generate diverse population and guided the search towards optimal solutions. Whereas, rest of algorithms having opposite behavior. Such algorithms can have faster convergence. But, it is noticed that the balance between these searches is one of the important aspect for generation of optimal solutions. Several popular meta-heuristic algorithms are ACO, PSO, CSS, MCSS, BB-BC, ABC, BH, etc. The wide range of problem is being solved using these meta-heuristic algorithms.

1.5 Motivation

Clustering problem get wide attention by research community in recent time. This problem is addressed by researcher's practical as well as theoretical point of view. It can be described as unsupervised learning for arranging the data objects into distinct clusters. This arrangement can be done with the help of dissimilarity measures. The objects within cluster are more similar than others. The objective of clustering is to arrange n data objects into k clusters for computing unseen information [24-27]. Clustering techniques had proven its significance in diverse area such as biology, market research, image processing, process monitoring, bioinformatics and pattern recognition [9-13]. Many clustering algorithms are reported in recent years for addressing clustering problems by research community [5]. It is also seen that evolutionary and meta-heuristic algorithms are also applied for improving the clustering results. These algorithms consist of some approximation functions to handle complex problem and provide near to optimal solution. The pros these algorithms are listed as.

- Having variety of strategies for optimum solution.
- Independent of the size of problem.
- Having problem specific objective functions, constraints and variables.
- Adaptable in nature i.e. having capability to react according to problem definition.
- Require less mathematical function.
- Optimal for solving for combinatorial and nonlinear problems.
- Having low computation power.
- Intelligent mechanism to avoid local optima.
- Problem can be solved with different initial points.

Apart from aforementioned advantages, it is also observed that several shortcomings are associated with these algorithms [28-30].

- Identification of initial cluster centers.
- Stuck in local optima.
- Lack of population diversification mechanism.
- Imbalance exploration and exploitation processes.
- Convergence rate.
- Absence of universal method for accurate clustering.

1.6 Objective

On the basis of motivation, it is highlighted that large number of meta-heuristic algorithm are adopted for addressing clustering task. It is also pinpoint that meta-heuristic algorithms are also suffered with several shortcoming like solution quality, convergence and trap in local optima. The objectives of this thesis work based on the motivation are highlighted as

- To develop new algorithm for improving the accuracy of partitional clustering.
- To hybridize the existing algorithm for addressing diversity and convergence rate issues.
- To handle the local optima problem of partitional clustering using neighborhood search-based concept.

1.7 Thesis Organization

The thesis can be organized as

Chapter 1: This chapter gives the brief description regarding the introduction, motivation and objective of research works.

Chapter 2: This chapter presents a review on partitional clustering algorithms. The literature is categorized into meta-heuristic algorithms, automatic clustering algorithm and improved/ hybridized clustering algorithm.

Chapter 3: This chapter is based on the first objective of our study i.e. design a new algorithm for partitional clustering. This work explores the capabilities of ACRO algorithm for solving clustering problems.

Chapter 4: This chapter discusses the second objective of the study i.e. hybridization of existing algorithms. In this chapter, an improved version of big bang-big crunch algorithm (IBB-BC) is developed to address convergence rate and diversification issues. Several amendments are proposed in terms of chaotic maps and cellular automata-based concepts.

Chapter 5: In this chapter, a neighbourhood search-based cat swarm optimization algorithm is proposed to optimize the clustering problems. Additionally, a neighbourhood search strategy-based concept is also developed to handle local optima problem.

Chapter 6: This chapter contains the conclusion of the entire work and future work direction.

CHAPTER 2
REVIEW OF LIERATURE

2.1 Introduction

Clustering is a data analysis technique adopted in different research domains such as medical informatics, image processing, cloud computing, IoT, etc. The objective of clustering is to determine clusters of similar objects. The objects within a cluster exhibits more similarity than others. The similarity between objects are computed through an objective function. The objective function for clustering is defined in terms of distance function. It is also described as O = [o_1, o_2, o_3… o_n] is a dataset contains "n" number of data objects. C = [c_1, c_2 … c_k] is a set of clusters contains k clusters. Clustering arranges data objects of a dataset O into distinct set of clusters i.e. C with minimized value of distance measure. Large number of distance measures are reported for clustering task, most commonly used distance function is Euclidean distance [21-23].

2.2 Review of Clustering Algorithms

The entire literature review section is divided into three subsections. These are Meta-heuristic algorithm, Automated clustering algorithm and Improved /Hybridized clustering algorithm. Meta-heuristic can be described as algorithms with heuristic search mechanisms like natural phenomenon etc. Automated clustering algorithm are defined as algorithm with automatic selection of clusters and initial population. Improved or hybridized clustering algorithms are algorithms with improved search mechanism, hybridized existing clustering algorithms, etc. Figure 2.1 demonstrates the categorization of partitional algorithms.

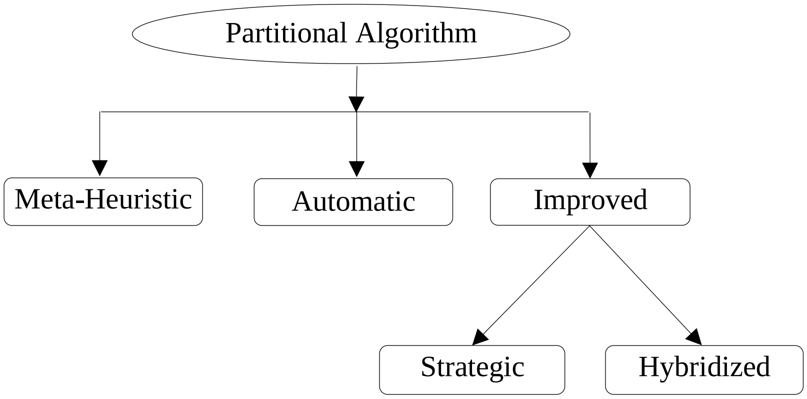

Figure 2.1: Partitional clustering algorithms

2.2.1 Meta-Heuristic Partitional Clustering Methods

This subsection demonstrates the work reported on meta-heuristic algorithms for solving partitional clustering Table 2.1. Large numbers of meta-heuristic algorithms inspired from natural phenomenon are reported for clustering problem [14-20]. Natural phenomena's include swarm intelligence, insect's behavior, natural process of living beings etc. Several algorithms are designed from natural phenomena's and these algorithms consist of some approximation function to determine the optimal solution. It is also noticed that these algorithms consist of local search and global search mechanisms for obtaining good candidate solutions. Bezdek et al. [31] developed genetic algorithm to address partitional clustering problems. This algorithm utilizes crossover and mutation operators for optimizing initial population. Moreover, the optimization process is directed through a fitness function. Iris dataset is considered for computing the simulation results of proposed algorithm and simulation results are compared with hard c-means algorithm. It is noticed that clustering results are significantly improved using GA. But, this algorithm is sensitive to initial population.

To tackle partitional clustering problem, Shelokar et al. [32] designed a new clustering algorithm based on the behavior of ants, called ant colony optimization algorithm. This algorithm consists of several key features of ant system like distributed computation, constructive greedy heuristic, positive feedback, and pheromone matrix. Further, five datasets are adopted for evaluating the efficiency of proposed ACO algorithm. The results are compared with SA, TS, and GA based clustering algorithms. It is observed that ACO algorithm converges on optimal solutions in comparison to existing algorithms. A tabu search (TS) based algorithm is reported for handling clustering problems in [33]. The working of this algorithm is descried in terms of five operations and three neighborhoods. The aim of these operations is to enhance the convergence speed. Whereas, neighborhoods are included for improving the population diversity. Simulation results showed that proposed tabu search based clustering algorithm obtains higher accuracy results than other algorithms.

Mahdavi et al. [34] developed a musical harmony based algorithm for solving partitional clustering problem in effective manner. In this algorithm, a new vector solution is derived through combination of all vectors in solution space. Moreover, K-means algorithm is integrated into harmony based clustering algorithm. The aim of hybridization is to enhance the convergence speed of harmony based clustering algorithm. Several well-known clustering datasets are adopted for evaluating the effectiveness of the proposed clustering algorithm. Its revealed that minimum intra cluster distance is achieved using the harmonic search based clustering algorithm. It is also noticed that this algorithm also improves the accuracy rate of clustering. Santosa and Ningrum [35] applied CSO for obtaining optimal clusters. It consists of two modes- tracing and seeking. Seeking mode corresponds to resting behavior of cats and responsible for local search. In contrast, tracing mode describes the haunting skills of cats and responsible for global search. The iris, soybean-small, glass, and balance scale datasets are taken for computing the effectiveness of CSO algorithm and simulation results are compared with K-means and PSO algorithms. It is noticed that proposed CSO algorithm works better for small scale dataset and extensive in terms of computational time.

The capabilities of artificial bee colony (ABC) also explored to obtain optimal solution for partitional clustering problem [36]. In the proposed clustering algorithm, Deb rule is considered to guide the search in optimal direction. Three datasets are taken for evaluating the simulation results of ABC based clustering algorithm. The simulation results are compared with GA, SA, TS, ACO and K-NM-PSO algorithms. Simulation results showed that ABC based clustering algorithm achieves minimum intra cluster distance and higher f-measure rate. A big bang-big crunch algorithm (BB-BC) presented for obtaining optimal cluster centers [20]. This algorithm consists of two phases- big bang and big crunch. BB phase determines random points in search space and acts as exploration phase. Whereas, BC phase optimized the random points via minimum cost approach and responsible for exploitation process. Four real life datasets are adopted to check the performance of BB-BC algorithm and simulation results are compared with existing algorithms. The proposed BB-BC algorithm obtains optimal cluster centers for all datasets in contrast to other algorithms.

Satapathy and Naik [37] applied TLBO algorithm for cluster analysis. It is based on classroom teaching methodology and having two phases-teacher and learner. Teacher is responsible for disseminating the knowledge among learners, whereas, learner improves its learning capability from teacher and other learners. Both artificial and real life datasets are adopted for assessing the simulation results of TLBO based clustering algorithm. The results showed that TLBO achieves better results in terms of accuracy and intra cluster distance, but suffers with slow convergence. Senthilnath et al. [38] presented firefly based clustering algorithm (FA) for addressing the optimal cluster center problem, especially for partitional clustering. The flash pattern of fireflies is considered to design the algorithm. The light intensity and movements of fireflies are the main component of FA. On the basis of light intensity and movement, two phases are designed. In first phase, variation in light intensity is measured for determining the position of fireflies. In second phase, movement of fireflies is computed. The efficiency of FA is assessed over thirteen datasets. Several clustering algorithms are considered for comparing the simulation results of FA algorithm. The FA achieves better quality results with most of clustering datasets.

A PSO based algorithm is designed for effective cluster analysis [15]. In this algorithm, populations are described through particles and these particles are subjected to move in entire search space. The simulation results are taken on seven datasets. The popular existing clustering algorithms are selected for comparing the simulation results of PSO algorithm. Simulation results showed that PSO algorithm is capable to perform clustering without knowing the cluster numbers. It is also noticed that PSO achieves better accuracy rate as compared to rest of algorithms. Hatamlou [39] developed a novel binary search algorithm to obtain high-quality clusters with better convergence speed. In this work, initial seed points are generated from different location and data instances are allocated to the nearest ones. Further, it is stated that if, current objective function is less than previous objective function. The search will proceed in same direction; otherwise, it will proceed in opposite direction. The six benchmark datasets are adopted for evaluating the efficacy of binary search algorithm. It is stated that binary search algorithm effectively solves the clustering problems.

Taherdangkoo et al. [40] designed a blind naked mole-rats (BNMR) algorithm for clustering. The inspiration behind this algorithm is blind naked mole-rat colonies which effectively search food sources and protect the colony from attackers. The six datasets are selected for evaluating the simulation results of BNMR algorithm. Further, several existing algorithms are considered for comparing the simulation results of BNMR. Authors claimed that blind naked mole-rat algorithm provides excellent quality results with faster convergence speed. A new clustering algorithm inspired through “black hole phenomenon”, is developed for partitional clustering [19]. BH algorithm describes the population in terms of stars and for each star, an objective function is computed. The best star can be acted as black hole, whereas remaining stars represented as normal stars. The six datasets are adopted for evaluating the efficacy of BH algorithm. Some popular clustering algorithm are considered for comparing the simulation results of BH algorithm. It is concluded that BH algorithm is one of robust and effective partitional clustering algorithm.

A new clustering algorithm based on inverse transformation is reported for effectively handling the clustering task [41]. This algorithm works on fits the data in clusters approach rather than fitting the clustering model into data approach. It starts with artificial clustering structure selection and then determine the data points into these structure. For initialization process, seven structures such as line, diagonal, random, random with optimal partition, K-means++ initialization, line with uneven clusters and points structures are introduced. It is noticed that K-means* is also recognized as a post-processing algorithm similar to K-means algorithm. Kumar et al. [42] presented the MCSS algorithm for cluster analysis. In MCSS algorithm, local search is carried out through Coulomb law (electrical) and Gauss laws (magnetic). Whereas, global search is performed through "Newton second law of motion". Nine datasets are selected to check the effectiveness of MCSS algorithm. Simulation results stated that more accurate and effective clustering results are obtained by MCSS algorithm.

Łukasik et al. [43] reported grasshopper optimization algorithm (GOA) for obtaining optimal cluster centers. This algorithm formulates the characteristic of social interaction among grasshoppers as mathematical function. In turn, devise movement strategies of grasshopper through two components. First component describes the interaction of grasshoppers in the larvae stage and insect form. While, the second component describes the tendency of grasshopper in the direction of food source. The performance of GOA is assessed on five benchmark datasets. GOA obtains higher f-measure rate and is more stable algorithm for solving clustering problems. Deb et al. [44] designed an ESA for determine the optimal cluster centers. Further, ESA is integrated with C-means algorithm. The performance of proposed ESA is tested over four datasets and compared with classical clustering algorithms. The proposed ESA outperforms than classical algorithms in terms of accuracy. Moreover, the efficacy of proposed ESA is investigated over time series data. The K-means, PSO and Fuzzy C-mean algorithms are selected for comparing the simulation results of proposed ESA algorithm. It is noted that C-ESA also outperforms than other clustering algorithms for time series data clustering.

Nasiri and Khiyabani [45] formulated a new clustering algorithm based on swarm foraging behavior of humpback whales. Moreover, a hyper-cube mechanism is integrated in the proposed algorithm to explore search space. Eight clustering datasets are considered for evaluating the performance of whale optimization algorithm (WOA). It is noticed that WOA attains good quality solution than others clustering algorithms. To accelerate the clustering process, a gravitational algorithm is reported for solving clustering problem effectively in [46]. In this algorithm, the movement of excessive centroid is considered for better tradeoff between local and global searches. The   random initialization of cluster centers is avoided through variance and median methods. Thirteen well known clustering datasets are adopted to assess the gravitational algorithm. The proposed gravitational algorithm achieves better performance on seven datasets in terms of purity, MCR and F-score. A novel clustering algorithm based on gravity center methodology, called GCC is reported in [47]. This algorithm utilizes connectivity and cohesion principles for determining the similarity among data points. In GCC algorithm, critical distance is used to restrict the search, whereas Euclidean distance is used for similarity. Some real and synthetic datasets are selected for computing the simulation results of GCC algorithm and these results are compared with different variants of K-means. It is noticed that GCC algorithm performs well with small scale datasets.

Table 2.1: Demonstrates the meta-heuristic partitional clustering methods

Authors	Approach/ Method	Adoption Criteria	Dataset	Performance parameter	Advantage
Bezdek et al. [31]	Genetic algorithm	To optimize clustering problems	Iris	Accuracy (error)	Viable and efficient method for solving clustering problems
Shelokar et al. [32]	Ant colony optimization algorithm	To design robust method for clustering	Simulated and Chemical	Function and CPU time	Viable method for clustering
Liu et al. [33]	Tabu search algorithm	To optimize clustering problems	Data-52, Data-62, British Towns, Iris, Vowel, Serum, Sub cell cycle, German Towns and Crude Oil	Intra-cluster distance, standard deviation and error rate	Viable method for clustering

Mahdavi et al. [34]	Harmony search algorithm	To design a heuristic optimization algorithm for clustering	Dataset 1, Dataset 2 and Dataset 3	F-measure	Good quality solutions
Santosa and Ningrum [35]	Cat swarm optimization algorithm	To solve clustering problems	Iris, Soybean-small, Glass, Balance scale	Execution time and error rate	Work accurately on small datsets
Zhang et al [36]	Artificial bee colony algorithm	To explore the capabilities of ABC algorithm in clustering field	Thyroid, Iris, and wine	Intra-cluster distance	Qualitative solution
Hatamlou et al. [20]	BB-BC algorithm	To obtain optimal solution	Iris, CMC, Cancer and Wine	Intra-cluster distance	Viable method for clustering
Satapathy Naik [37]	TLBO algorithm	To design robust method for clustering	Iris, wine, breast cancer, Haberman's Survival, glass and artificial datasets	Intra and inter cluster distance	Optimization method for clustering problems
Senthilnath et al. [38]	Fire Fly algorithm	To design an efficient method for clustering	Glass, Heart, Cancer, Cancer-Int, Diabetes, E. Coli, Thyroid, Wine, Credit, Iris Dermatology, Horse and Balance.	Error percentage and ranking	Efficient method for clustering
Cura [15]	Particle swarm optimization	To design robust method for clustering	Art1, Art2, Iris, Thyroid, Wine, CMC and Glass.	Intra cluster distance, error rate, computation time and number of clusters	Work even without prior knowledge of number of clusters

Hatamlou [39]	Binary search algorithm	To discover new clustering algorithm	Iris, Wine, Cancer Glass, CMC, Glass and Vowel	Intra-cluster distances, standard deviation and F-measure	High quality optimal cluster centers
Taherdangkoo et al. [40]	Blind naked mole rats	To solve clustering problems	Vowel, Crude oil, Wood defects, Wine, Control chart and Iris	Intra-cluster distances, Error, DB measure and Time	Better convergence rate and accuracy
Hatamlou [19]	Black hole algorithm	To solve clustering problems.	Iris, CMC, Cancer Vowel, Wine and Glass	Intra-cluster distances, standard deviation, error rate and ranking.	Viable method in clustering field
Malinen et al. [41]	K-means*	To accelerate the clustering process.	Thyroid, House, Wdbc, Breast, Glass, Bridge, Missa, Iris, Wine, Yeast, S(1,2,3,4) A1 and DIM (32,64,128,256)	MSE and NMI	Inverse transform step and random swap strategy
Kumar et al. [42]	MCSS algorithm	To solve clustering problems	Art1, Art2, Iris, Glass, LD, Thyroid Cancer, CMC, Vowel and Wine.	Intra-cluster distance, standard deviation and F-measure	Quality results
Łukasik et al. [43]	Grasshopper optimization algorithm	To optimize clustering problems	Glass, Wine, Iris, Seeds and Heart	Rand Index and standard deviation	Accurate clustering results

Deb et al. [44]	Elephant search clustering algorithm	To design a robust algorithm for clustering	Gesture, Mice Protein, Iris, Haberman, CBF data, Face all, Swedish leaf, Two pattern, Yoga, Faces UCR, CinC ECG torso, Diatom size reduction, Symbol and Italy power demand	Time and accuracy	Balanced exploration and exploitation search mechanism
Nasiri and Khiyabani [45]	Whale optimization clustering algorithm	To optimize clustering problems	ART, Cancer, Wine, CMC, Balance, Glass, Thyroid and Iris	Intra-cluster distance and standard deviation	Able to handle local optima
Alswaitti et al. [46]	Optimized gravitational based data clustering algorithm	To make a balance among exploration and exploitation processes	Cancer, Cancer-Int, Iris, New Thyroid, Dermatology, Transfusion, Haber man, Wine Landsat, and Balance, Glass, Lung Cancer and Seeds	MCR, F-score and purity	Enhanced convergence rate
Kuwil et al. [47]	Gravity center clustering algorithm	To optimize clustering problems	Salmonellosis to shigellosis, salmonellosis, Babesiosis, health infectious disease, Unplanned hospital visits, diabetes and Medicare National DMEPOS HCPCS aggregate	Time and success rate	Efficient method for small scale dataset

2.2.2 Automatic/Automated Partitional Clustering Methods

This subsection discusses several automatic/automated partitional clustering methods Table 2.2. It is observed that clustering algorithms suffer with several performance issues like lack of prior knowledge of clusters, random initialization, initial seed points, etc. To overcome these issues and automate clustering process, extensive works have been carried in literature. Hartigan and Wong [48] developed an improved version of K-means algorithm to seek the optimal solution. A new method is designed for initial cluster selection which arranges the data points in reference to overall mean of the sample, then elects a point as initial cluster center. Several datasets are selected for evaluating the simulation results of improved K-means. Further, AS-58 algorithm is selected for comparing the results of improved K-means algorithm. and simulation results are compared with AS-58 algorithm. It is observed that proposed algorithm successfully overcomes the initialization issue of K-means algorithm.

To seek the optimal solution for clustering task, Zhang et al. [49] integrated the K-means and HS algorithm. The objective of this integration is to design a new initialization method for K-means algorithm. The simulation results of proposed algorithm are taken over randomly generated datasets. The K-means and EM algorithms are selected for comparing the results of proposed algorithm i.e. K-means-HS. It is stated that K-means-HS efficiently addresses the initialization deficiency of K-means algorithm and significantly improves the clustering results. A variant of K-means i.e. K'-means designed to solve clustering problems [50]. This method is an extended version of K-means algorithm and works without knowing the cluster numbers. It works in two phases. Initially, k numbers of cluster centers are disseminated in such a way that each cluster consists of one or more cluster centers. In second phase, cluster centers are aligned using minimum value of cost function. The simulation results of K'-mean algorithm are assessed over several existing datasets. Authors claimed that K'-mean is capable to perform clustering without knowing cluster numbers.

A new initialization method on the concept of neighborhood rough set theory is presented for effective clustering in [51]. This method is integrated with K-mean algorithm for addressing the performance issues. Three datasets are adopted for computing the simulation results of proposed algorithm. The several existing initialization techniques are selected for comparing the simulation results of proposed algorithm. Authors claimed that neighborhood concept based initialization method provides superior clustering results. Xiao et al. [52] developed a robust clustering approach based on K-means and quantum-inspired genetic algorithm (KMQGA). In this work, the properties of QGA (Q-bit representation) and GA are adopted for alleviating the deficiencies of K-means algorithm. In KMQGA, chromosome is represented through Q-bits and DB index is considered as a criterion function. The simulation results of KMQGA is taken over real life datasets. It is noticed that KMQGA is capable to find optimal cluster centers without prior knowledge of clusters.

A hybrid version of ant clustering algorithm with k-harmonic means is reported in [53]. This algorithm contains merits of ACA and KHM algorithms i.e. initialization ability of K-harmonic means and local optima handling ability of ant algorithm. Five benchmark datasets are adopted to check the efficacy of ACAKHM. It is observed that ACAKHM achieves better-quality results than other algorithms, but computationally extensive. Niknam et al. [54] hybridized the K-means with modify imperialist competitive algorithm (K-MICA). This algorithm works in two steps-expectation and maximization. In expectation step, likelihood function is computed, while in maximization step, the maximum likelihood is estimated. A modified expectation method is also incorporated for determining accurate cluster number. Iris, Wine, CMC, Vowel and Manufactured datasets are adopted for evaluating the results of K-MICA. It is stated that K-MICA obtains better intra cluster distance than compared algorithms. Erisoglu et al. [55] developed a new initialization method for clustering problems and this method is integrated into K-means. In this method, two attributes of the dataset are selected to map the data in bi-dimensional feature space. The first attribute is a variable having maximum value of the variation coefficient, called main axis. Second attribute is determined through the value of correlation between the main axis and attributes. The attribute having minimum value of correlation is selected as second variable. Several benchmark datasets are adopted for evaluating the results of aforementioned algorithm. It is noticed that proposed method significantly improves the performance of K-means.

To automate the clustering process, a two-stage genetic algorithm, called TSGA is reported in [56]. In TSGA, variable length coding scheme is adopted to determine number of clusters. The consistency of clusters is maintained through selection and mutation operations. In addition to it, maximum attribute range partition method is implemented for selecting initial cluster centers. The eleven datasets are adopted for validating the result of TSGA. The NJW, ANJW, HAC, K-means, SGKC and HKM algorithms are selected for comparing the experimental results of TSGA. It is noticed that TGCA performs well with low dimensional data and insensitive to the initial population. Liu et al. [57] presented a GTCSA for automatic clustering. A new genetic operator, called antibody gene transposon is introduced for detecting number of clusters. Twenty-three datasets are selected for computing the experimental results of GTCSA. Further, standard clustering algorithms are adopted for comparing the results of GTCSA. Simulation results stated that GTCSA is capable to perform clustering without prior knowledge of cluster.

A message-based GA algorithm (GAMS) is presented for solving partitional clustering problems [58]. This algorithm contains message based similarity measure. The messages are defined in terms of responsibility and availability, and exchanged over data objects and clusters. The responsibility of data point corresponds to an evidence regarding cluster centers, whereas, availability refers to an evidence regarding appropriateness of data point to cluster centers. Further, GAMS consists of variable-length chromosome representation and problem-specific evolutionary operators. Both real life and artificial datasets are adopted for evaluating the experimental results of GAMS algorithm. Results showed that GAMS obtains significant clustering results than others. Khan and Ahmad [59] addressed randomized cluster center initialization problem of clustering task and designed a new cluster center initialization method. The proposed method consists of two parts-relevant attributes selection, and initial cluster centers computation. In relevant attribute selection, two methods are proposed, the first method selects the promising attribute, while the second method computes the significant attributes. The outcomes of the aforementioned methods correspond to initial population of the K-medoids algorithm. The well-known datasets are selected for computing the simulation results of proposed algorithm. The existing initialization methods are adopted for comparing the results of proposed algorithm. The proposed algorithm provides faster convergence with better accuracy.

Tzortzis and Likas [60] developed MinMax K-means algorithm for cluster center initialization issues. In this method, weights are allocated to clusters according to variance and also restrained large variance of clusters. Further, an iterative procedure is incorporated in algorithm for automation task. The performance of MinMax K-means is assessed over several benchmark datasets. K-means variants are adopted for comparing the experimental results of MinMax K-means. Results confirmed that MinMax K-means is a robust method for clustering task. Peng et al. [61] designed an automatic clustering algorithm based on tissue-like membrane system. In this work, a computing framework based on tissue-like membrane system is developed to limit the number of clusters with optimal partitions. Further, a modified velocity-position model is also proposed for addressing acceleration and diversity issues. Six datasets are selected for measuring the efficiency of aforementioned algorithm. Further, three existing automatic clustering techniques are considered for comparing the experimental results of proposed algorithm. It is observed that membrane clustering algorithm is superior than other algorithms and efficiently works with different dimensions of data.

Menendez et al. [62] presented two algorithms for clustering of objects based on the concept of medoid and ant colony optimization. In first algorithm, ant colony method is applied to determine the optimal solution, called METACOC algorithm. Whereas, in second algorithm an ant colony procedure is applied to automate the clustering task, called METACOC-K algorithm. Some real and synthetic datasets are selected to compute experimental results of above mentioned algorithms. METACOC and METACOC-K algorithms provide significant results in terms of accuracy and f-measure rate. To automate the clustering process, Zhao et al. [63] proposed an improved version of K-means algorithm. Cuckoo search method is applied to generate initial cluster center for K-means. Hence, initial cluster centers are evaluated using cuckoo search algorithm for k-means instead of random initialization. Two-dimensional data points are selected for assessing the experimental results of CS-KM and results are compared with K-means algorithm. The CS-KM algorithm achieves better execution time than K-means.

Huang et al. [64] developed a harmonious genetic clustering algorithm (HGCA) to automate the clustering process. The computing framework of HGCA is based on eugenic theory. Further, in this work, mating strategy is applied for selecting the suitable mate to chromosomes. The efficiency of HGCA is examined using existing datasets. It is stated that HGCA is capable to perform clustering without knowing of clusters numbers and also having higher accuracy rate than others. A hybrid version of black hole algorithm for solving clustering problems is reported in [65]. The aim of hybridization is to deal the initialization issue of black hole algorithm. So, initial population of BH algorithm is computed through K-means algorithm. The six datasets are adopted for evaluating the experimental results of hybrid BH algorithm. Further, GSA, BH and K-means algorithms are selected for comparing the results of hybrid BH algorithm. It is seen that hybrid black hole clustering algorithm gives better quality results for red wine and glass datasets.

Table 2.2: Summarized the automatic/automated partitional clustering methods

Authors	Approach/ Method	Adoption Criteria	Dataset	Performance parameter	Advantage
Hartigan and Wong [48]	KMNS algorithm	To explore the working phenomenon of K-means clustering algorithm	-	Intra cluster distance	A new initialization method
Zhang et al. [49]	K-Harmonic means	To overwhelmed the deficiencies of K-means algorithm	BIRCH and Hier	Initialization and local optima	Insensitive to initial cluster centers
Zalik [50]	K'-means	Number of clusters and initial cluster selection	S1, S2, S1 and Wine	Number of samples, mixing proportion, mean vector and standard variance	Work without prior knowledge of clusters numbers
Cao et al. [51]	Neighborhood based initialization method for K-means algorithm	To solve poor initialization issue	Iris, Wine and Glass	Accuracy, precision and recall	Solved the initialization issue of K-means algorithm
Xiao et al. [52]	Quantum inspired genetic algorithm for K-means clustering	To develop robust method for clustering	Sds1, Sds2, Sds3, Glass, Wine, SPECTF-Heart and Iris	Number of clusters and standard deviation	Work without prior knowledge of number of clusters

Jiang et al. [53]	ACO with K-harmonic means	To design an efficient algorithm for clustering	ArtSet1, ArtSet1, Glass, Iris and Wine	KM, KHM, F-measure and runtime	Insensitivity towards initialization.
Niknam et al. [54]	Hybrid K-MICA	Automatically find the number of clusters.	Iris, Wine, CMC, Vowel and Manufactured datasets	Function values and standard deviations	Automatically determine number of clusters
Erisoglu et al. [55]	Proposed algorithm	Initial cluster selection	Iris, Wine, Letter, Ruspin and Spambase	Error percentage and rand index	Solved the initialization issue of K-means algorithm
He and Tan [56]	Two-stage genetic algorithm	To automate clustering process	Ruspini, Atestdata1, Atestdata2, Atestdata3, WBCancer, Iris, Glass, Connectionist bench, Zoo, Lung cancer and Ionosphere,	Accuracy, rand index and adjusted rand index	Insensitive to initial population
Liu et al. [57]	Gene transposon based clone selection algorithm	Automatically find the number of clusters.	AD(10_2, 15_2), D(14_2, 20_2) Data(1,2,3,5,9,8) Long1, Sizes5, Data3, Synthetic(1,3) Liver disorder, Sticks, Twenty, Glass, Ionosphere, New Thyroid, WBC, Wine, Diabetes and Iris	Minskowski score and standard deviations	Automatically determine number of clusters
Chang et al. [58]	GA with message-based similarity	Number of clusters	Data (1, 2, 3, 4, 5), Iris, Breast and Wine	Number of clusters and Rand Index	Automatically determine number of clusters

Khan and Ahmad [59]	Cluster center initialization algorithm	To solve poor initialization issues	Soybean, Mushroom, Dermatology, Lung-Cancer, Zoo, Vote and Breast-Cancer	Accuracy, precision, recall, time complexity, confusion and match metric	Efficient method for initial cluster center selection is developed
Tzortzis and Likas[60]	MinMax K-means	To solve poor initialization issues	Coil1, Coil2, Coil3, Multiple features-pixel averages, Multiple features profile correlations, Pendigits, Olivetti, Ecoli, and Dermatology	NMI and Time	Efficient method for initial cluster center selection is developed
Peng et al. [61]	Membrane clustering algorithm	Number of clusters	Iris, New thyroid, Vowel, Glass, Wine and Cancer	Numbers of clusters, classification errors and exution time	Determine number of clusters with optimal partitions
Menéndez et al. [62]	METACOC and METACOC-K	Automatically detect the number of clusters	Synthetic 1, Synthetic 2 and Synthetic 3	Rand index, silhouette metric and computational time	Automatically identified the number of clusters
Zhao et al. [63]	Improved K-means clustering algorithm	To solve initial cluster center selection problem	Two-dimensional (100) data points	Execution Time	Better execution time
Huang et al. [64]	Harmonious genetic clustering	To recognize number of clusters	Vowel, Iris, C-Cube, Letter, Animal (PHOG), 20Newsgroups, DS1, DS2, DS3 and DS4.	Fitness index DBI, CS and VRC	Automatically determine the number of clusters

Pal and Pal [65]	Black hole and K-means clustering algorithm	To solve initialization issue of Black hole algorithm	Pima Indians diabetes database, Iris, Lower back pain symptoms, Red wine quality, Wine and Glass	Intra cluster distance	Resolved initialization issue

2.2.3 Improved Partitional Clustering Algorithms

This subsection describes recent works reported in the direction of improved partitional clustering algorithms Table 2.3. Yang et al. [66] combined the PSO and KHM, called PSOKHM for obtaining optimal clusters. It contains the merits of both algorithms i.e. convergence speed of KHM and optimization ability of PSO for solving clustering problems. In turn, hybrid algorithm is also overwhelmed the local optima issue of KHM and enhanced the convergence speed of PSO algorithm. Seven datasets are adopted for evaluating the experimental results of PSOKHM. Further, PSO and KHM algorithms are selected for comparing the results of PSOKHM algorithm. Simulation results stated that PSOKHM achieves better quality clustering results than others. Yin et al. [67] presented a hybrid algorithm based on improved gravitational search algorithm and KHM algorithm for addressing clustering problems. This works explores the advantage of both algorithms i.e. better convergence rate of KHM algorithm and diversity mechanism of GSA. Seven benchmark datasets are selected for evaluating the simulation results of hybrid clustering algorithm. Authors claimed that IGSA-KHM based hybrid clustering algorithm improves the accuracy rate of clustering.

A hybrid version of ABC algorithm with GA is presented in [68]. This work explores the capability of crossover operator for improving information exchange procedure of bees. Six benchmark datasets are selected for evaluating the performance of hybrid ABC algorithm. Several well-known clustering algorithms are adopted for comparing the simulation results of ABC-GA. Simulation results showed that ABC-GA obtains higher f-measure rate. Huang et al. [69] presented a hybrid clustering algorithm based on continuous ACO and PSO techniques. In this work, four search approaches are also designed for obtaining optimal clustering results. These are sequential, parallel, sequence with pheromone-particle solution and global best. In the first (sequential) approach, algorithms share the same pheromone table, whereas in the second approach (parallel), new solutions are generated based on pheromone table. In the third approach, new solutions are created from enlarged pheromone table, and best solutions are stored in the table. In case of global best approach, the best solution is shared from the pheromone table. Several well-known datasets are adopted for assessing the simulation results of proposed hybrid clustering algorithm. Simulation results stated that sequence approaches with enlarged pheromone particle table achieves good clustering results as compared to other approaches.

Hatamlou and Hatamlou [70] designed a two-stage clustering approach by integration of PSO and heuristic search algorithm, called PSO-HS. The proposed clustering algorithm works in two-stages. In first stage, initial candidate solution is generated through PSO algorithm. In second stage, the quality of solution is improved through heuristic search algorithm. The performance of two stage PSO-HS clustering algorithm is evaluated over seven benchmarks and simulation results are compared with K-means, PSO, GSA, BB-BC clustering algorithms. The PSO-HS clustering algorithm delivers optimal clustering results. Jiang and Wang [71] presented a BB-PSO algorithm for solving clustering problems efficiently. A cooperative co-evolution (CC) method is integrated into PSO algorithm for improving convergence rate and population diversity. It is noticed that CC method works as a decomposer, while, PSO algorithm acts as an optimizer. Several real life and synthetic datasets are adopted for evaluating the experimental results of proposed hybrid algorithm. Further, large number of algorithms like PSO, SRPSO, ACO, ABC, DE, K-means are selected for comparing the simulation results of hybrid algorithm. It is claimed that hybrid clustering algorithm obtains minimum intra cluster distance.

Kumar and Sahoo [72] developed a new algorithm using ICSO and KHM, called ICSOKHM for effective cluster analysis. This work addresses two issues of CSO algorithm-diversity and exploitation. To handle diversity and exploitation issues, two improvements are inculcated in CSO algorithm. These improvements are inertia weight and initialization method for KHM algorithm. Seven datasets are selected to examine the efficiency of ICSOKHM clustering algorithm. It is noticed that hybridization of KHM and ICSO improves the convergence speed as well as effectively handles local optima situation. Wang et al. [73] developed an improved version of FPA (IFPA) for clustering task. In this work, for improving the diversity mechanism and search capabilities of flower pollination algorithm, pollen operator of ABC is adopted. Local search is also enhanced through mutation and crossover operators. Several artificial and real datasets are selected for evaluating the simulation results of aforementioned algorithm. The well-known algorithms are adopted for comparing the results of IFPA algorithm. Authors claimed that IFPA obtains higher accuracy rate with small datasets. Pakrashi et al. [74] developed a new clustering algorithm based on heuristic Kalman filtering algorithm (HKA) and K-means. The merits of both algorithms are listed as faster convergence rate of K-means algorithm and global exploration ability of HKA. Both the advantages are incorporated in a single algorithm, called HKA-K clustering algorithm. Furthermore, a new conditional restart mechanism is also introduced for avoiding local optima condition. Seven datasets are adopted for computing the efficiency of HKA-K algorithm. Further, HKA, KGA, GAC, ABCC and PSO algorithms are selected for comparing the results of HKA-K. The simulation results stated that HKA-K achieves optimal results in terms of better convergence rate.

To obtain optimum clustering results, a two-step ABC algorithm is reported in [75]. Authors propose three improvements in ABC algorithm. These improvements are listed as initial cluster center positions, modified search equations and abandoned food source locations. Initial cluster centers for ABC is generated through K-means. The search equations of ABC are improved using PSO algorithm for discovering the promising solutions. Both real and artificial datasets are adopted for evaluating the simulation results of two-step ABC algorithm. Furthermore, standard clustering algorithms are selected for comparing the results of two-step ABC algorithm. Simulation results showed that proposed improvements improves the performance of traditional ABC algorithm in significant manner. Han et al. [76] designed a new diversity mechanism for GSA, especially for handling clustering problems. The diversity mechanism is inspired from the collective response of birds. This method works in three steps i.e. initialization, identification (nearest neighbors) and orientation alteration. In initialization step, the candidate population is generated and forwarded to the second step i.e. nearest neighbor. In second step, nearest neighbors are identified through a neighborhood strategy. While, in third step, the candidate solution is updated according to nearest neighbor. Thirteen datasets are selected or assessing the performance of aforementioned GSA clustering algorithm. The well-known algorithms are adopted for comparing the results of GSA. It is noticed that GSA achieves minimum intra cluster distance.

To design an efficient partitional clustering algorithm, Abualigah et al. [77] combined harmonic search (HS) algorithm with krill herd algorithm (KHA), called KHA-HS. The searching capabilities of KHA algorithm is improved through global search operator of HS algorithm. Some standard clustering datasets are considered for evaluating the performance of KHA-HA clustering algorithm. The simulation results are compared with GA, PSO, H-GA, H-PSO, HS and KHA clustering algorithms. It is seen that KHA-HS conveys good quality results. Zhou et al. [78] presented an improved version of social spider optimization (SSO) algorithm for cluster analysis. To improve the search mechanism and convergence rate, a simple method is integrated into social spider algorithm. The eleven datasets are adopted for assessing the experimental results of improve SSO algorithm. Several standard clustering algorithms are selected for comparing the simulation results of improved SSO algorithm. The improved SSO algorithm outperforms than other algorithms in terms of convergence speed and robustness.

Boushaki et al. [79] reported an improved cuckoo search (ICS) algorithm for effective clustering. In this work, authors introduce a non-homogeneous update mechanism inspired from the quantum theory for improving search mechanism of CS algorithm. In addition to this, the rand () function is replaced with chaotic maps for better convergence rate. The performance of improved CS clustering algorithm is compared with GA, KCPSO, DE, KICS, QPSO, HCSDE, GQCS and traditional CS algorithm. The experimental results showed that ICS algorithm achieves better accuracy results. To obtain optimal solution for clustering problems, Lakshmi et al. [80] hybridized the K-means and crow search algorithm (CSA-K). The shortcomings associated with K-means such as selection of cluster center, local optima are improved through Crow search algorithm. It is noticed that integration of K-means and crow search algorithm, called CSA-K make the balance between diversification and intensification processes. The six datasets are adopted for evaluating the experimental results of CSA-K. Further, K-means++, Genetic K-means, PSOK Means and K-means algorithms are selected for comparing the results of CSA-K. The CSAK provides better performance than other algorithms using average intra cluster distance.

Bouyer and Hatamlou [81] combined KHM, ICS and PSO algorithms for designing the robust clustering algorithm to obtain optimal clusters. The proposed algorithm contains the merits of ICS i.e. global optimal solution; particle swarm optimization i.e. ability to handle local optima situation and K-harmonic mean i.e. faster convergence. Several standard benchmark datasets are selected for assessing the simulation results of proposed clustering algorithm. It is observed that proposed clustering algorithm insensitive towards initialization and also have faster convergence rate. Bijari et al. [82] explored the capability of BB-BC algorithm for cluster analysis, called MBB-BC. The better coordination between global and local searches is achieved through memory based concept. The six datasets are adopted for computing the simulation results of MBB-BC algorithm. Furthermore, GA, PSO, GWO and original BB–BC algorithms are selected for comparing the results of MBB-BC algorithm. The simulation results showed that memory enriched BB-BC algorithm provides superior clustering results than other algorithms.

An improved CSO (ICSO) algorithm is reported for effective cluster analysis in [83]. In this work, a better tradeoff between both of searches i.e. local and global are established through several amendments. An improved local search method is also incorporated for handling local optima. The five benchmark datasets are adopted for evaluating the efficacy of ICSO algorithm. The well-algorithms are selected for comparing the results of ICSO algorithm. The Simulations result showed that ICSO achieves better accuracy rate than others. Kumar and Singh [84] presented an improved TLBO (I-TLBO) algorithm for partitional clustering. The population diversity and convergence speed issues are handled through chaotic maps. Further, an enhanced local search mechanism is developed for balancing exploration and exploitation features of algorithm. The performance of proposed TLBO clustering algorithm is tested over five benchmark datasets. Authors claimed that TLBO clustering algorithm conveys good quality results than other algorithms.

To optimize clustering problems, an enhanced version of black hole algorithm is reported in [85]. This works addresses the exploration issue of black hole algorithm and this issue is resolved through inclusion of Levy flight concept in black hole algorithm. The aim of levy flight concept is to restrict the movement of stars using step size which is determined through levy distribution. The above process can explore the entire search space effectively. Six benchmark datasets are adopted for assessing the efficacy of black hole algorithm. The simulation results are compared with K-means, PSO, ABC, BAT, GSA, BB-BC, CS, GWO and BH clustering algorithms. The black hole clustering algorithm achieves better quality clustering results. Kaur et al. [86] hybridized the chaos and flower pollination algorithm (C-FPA) to overwhelm the shortcoming of classical FPA clustering algorithm. The FPA algorithm is good to explore local and global search space for candidate solution, but sometimes traps in local optima. To avoid local optima situation, chaos maps are integrated into FPA algorithm. Sixteen standard clustering datasets are adopted for assessing the experimental results FPA algorithm. The classical FPA, CSA, BHA, BA, FFA, and

PSO over K-means algorithms are selected for comparing the simulation results of C-FPA. It is observed that C-FPA achieves better results in terms of cluster integrity than other algorithms.

Table 2.3: Depicts the improved partitional clustering methods

Authors	Approach/ Method	Adoption Criteria	Dataset	Performance parameter	Advantage
Yang et al. [66]	PSO-KHM algorithm	To design a robust clustering algorithm	ArtSet1, ArtSet2, Iris, Glass, Cancer, CMC and Wine.	KHM (X, C), runtime and F-measure	• Improved convergence rate • Escape from local optima • Solved cluster center initialization problem
Yin et al. [67]	Improved GSA and k-harmonic means	To design robust clustering approach	ArtSet1, ArtSet2, Iris, Glass, Cancer, CMC and Wine	KHM, F-measure and runtime	• Escape from local optima • Improved convergence rate
Yan et al. [68]	Hybrid artificial bee colony algorithm	To enhance the convergence speed	Iris, Wine, CMC, WBC, Glass and LD	Intra-cluster distance, standard deviation and rank	Enhanced convergence rate
Huang et al. [69]	Hybridized ACO and PSO clustering algorithm	To handle local optima problem	Breast cancer, Liver disorders,Ecoli, Pima, Yearst, Contraceptive method choice, Iris, Wine and German	Intra-cluster distance	Efficient to handle local optima

Hatamlou and Hatamlou [70]	Particle swarm optimization with heuristic search	To improve the convergence rate	Iris, wine, cancer, crude oil, CMC, glass and vowel	Error rate, sum of intra-cluster distances and rank	• Able to handle local optima • Improved the convergence rate.
Jiang and Wang [71]	Cooperative bare-bone PSO algorithm	To address the initial population section and convergence rate issues	Art1, Art2, Vowel, CMC, Glass, Wine, Ionosphere, Image Control, Iris and Cancer	Intra cluster distance, execution time and accuracy	• Improved the convergence rate. • Solved cluster center initialization problem
Kumar and Sahoo [72]	ICSO and KHM algorithm	To accelerate the convergence rate of cat swarm optimization algorithm	Synthetic1 Synthetic2, Iris, Glass, Cancer, CMC and Wine	KHM, F-measure and runtime	• Improved convergence rate • Escape from local optima
Wang et al. [73]	Flower pollination algorithm with bee pollinator	To solve Local optima and convergence rate issues	Artificial (set one, set two), Haberman's Survival, Iris, WBC, CMC, Statlog, Balance Scale, Seeds and Wine	Intra-cluster distance	• Able to handle local optima • Enhanced convergence rate
Pakrashi et al. [74]	Heuristic Kalman algorithm and K-means algorithm	To design a new approach for clustering.	Artset1, Artset2, Iris, Wine, Glass, CMC and Cancer	Intra-cluster distances, Adjusted rand index, Davies-Bouldin index DB and Time.	• Efficient to handle local optima • Improved convergence speed

Kumar and Sahoo [75]	Two-step artificial bee colony	To enhance the performance of artificial bee colony algorithm	ART 1, ART 2, Iris, Wine, CMC, Cancer and Glass	Intra-cluster distance, standard deviation and rank	• Efficiently solved initial cluster selection problem. • Better convergence speed
Han et al. [76]	BF and GSA	Diversify the solution search	Heart, Balance, Cancer, Credit, Dermatology, Diabetes, Glass, Horse, Iris, Thyroid, Cancer-Int and E. Coli	Intra-cluster distances and Error rate	Introduced new diversity mechanism
Abualigah et al. [77]	Krill herd with harmony search algorithm	To explore the the capabilities of Krill herd algorithm	CMC, Iris, Vowel, Cancer, Glass Seeds and Wine	Sum of intra-cluster distances and error rate	Efficient method for clustering
Zhou et al. [78]	SMSSO algorithm	To design a robust algorithm for clustering	Art1, Art2, Iris, TAE, Seeds, Heart, Haberman's survival, Balance scale, Cancer, CMC and Wine	Intra- cluster distance	• Enhanced local and global search ability • Able to handle local optima • Enhanced convergence speed
Boushaki et al. [79]	QCCS algorithm	Design a method to handle local optima, initial centroid selection and slow convergence rate issues	Iris, Seeds, Blood Wine, CMC and Cancer	Intra- cluster distance and error rate	• Solved cluster center initialization problem • Enhanced convergence speed

Lakshmi et al. [80]	Crow search algorithm and K-means	To address K-means clustering algorithm initialization issue	Iris, Wine, Glass, Cancer, CMC and Survival	Purity, NMI, RI and F-measure	Efficient method for clustering
Bouyer and Hatamlou [81]	K-Harmonic means combined with improved cuckoo search and particle swarm optimization	To address Local optima, initial centroid selection problems of clustering algorithm	ArtSet1, ArtSet2, Iris, Wine, Wisconsin Breast, Ripley's glass, Contraceptive method choice, Thyroid, Vowel and Ecoli	Sum of squared errors, Objective function of KHM, F-measure, runtime, DB index, Silhouette coefficient and error rate	Vivial method for clustering
Bijari et al. [82]	ME-BB-BC algorithm	To make a balanced tradeoff between exploration and exploitation processes	Iris, Wine, Cancer, CMC, Glass and Vowel	Intra-cluster distance	Enhanced convergence rate
Kumar and Singh [83]	Improved CSO algorithm	To accelerate the search mechanism	Iris, Wine, CMC, Cancer and Glass	Intra cluster distance, standard deviation and F-measure	Balanced exploration and exploitation processes
Kumar and Singh [84]	Chaotic TLBO algorithm	To handle the diversity issue	Iris, Cancer, CMC, Wine and Glass	Intra cluster distance, standard deviation and F-measure	• Enhanced convergence rate • Quality solution

Abdulwahab et al. [85]	Levy black hole clustering algorithm	Make balance among local and global search processes	Iris, Wine, CMC, Cancer, Glass and Vowel	Intra cluster distance and Error rate	• Enhanced convergence rate • Able to handle local optima
Kaur et al. [86]	Chaotic flower pollination algorithm	To handle local optima problem	Iris, Wine, Breast-cancer, Glass, Balance, Dermatology, Haberman, Ecoli, Heart, Tae, Spambase, Ilpd, Leaf, Libras, Qualitative bankruptcy and Synthetic	Integrity, execution time, NIC and stability	• Able to handle local optima situation • Enhanced convergence speed

CHAPTER 3

ACRO ALGORITHM FOR PARTITIONAL CLUSTERING

3.1 Introduction

Recently, large number of algorithms had developed to determine optimum solution for diverse optimization problems. In literature, ACRO algorithm is presented for handling large and complex global optimization problems [90,91]. This algorithm is inspired through different chemical processes like monomolecular, bi-molecular, etc. It is noticed that ACRO algorithm attains better quality results than same class of algorithms. This chapter investigates the ACRO algorithm for solving partitional clustering problems. It is seen that several performance issues are associated with ACRO algorithm such as convergence rate and local optima. This chapter addresses these issues of ACRO algorithm.

3.2 Artificial Chemical Reaction Optimization Algorithm

Alatas developed a new algorithm, called ACRO inspired from chemical reactions [90]. Like other algorithms, ACRO algorithm also contains a set of population and it is defined in terms of reactants. Further, a series of chemical reactions are computed to determine optimal reactants. The chemical reactions select the reactants on the basis of their concentrations and potentials. ACRO algorithm mainly consists of two types of reaction i.e. consecutive reaction and competing reaction. The consecutive reactions can join the reactants in serial manner, whereas, competing reactions occurs on different reactants on the basis of specific condition. Both of these reactions are utilized to compute optimal reactant. The same process is repeated, until termination condition is not met. The termination condition is described as maximum number of iterations. The algorithmic step of ACRO algorithm are given as

Step 1: Set the parameters of ACRO algorithm such as population, reactants, iteration number etc.

Step 2: Pick the reactants in random order and evaluate the objective function.

Step 3: If flag is on, go for consecutive reaction, otherwise go for competing reaction.

Step 4: Update reactants as per reactions mechanism.

Step 5: If termination condition is met. Stop the execution of algorithm, else repeat steps 3-4.

Step 6: Obtain the optimal solution in the form of reactants.

3.3 Proposed ACRO Algorithm

This section discusses the ACRO algorithm for solving partitional clustering problems. ACRO algorithm obtains optimal solution for diverse optimization problems due to strong exploitation capability [90,91]. But this algorithm, sometime does not converge on an optimal solution. It also noticed that in last iteration, the algorithm traps in local optimum instead of global. This work presents some amendments in ACRO algorithm for overcoming its performance issues. These amendments make the ACRO algorithm more promising for solving clustering problems. Hence, to address these aforementioned issues, two operators are developed and integrated into ACRO algorithm. The subsection 3.3.1 discusses the proposed amendments in details.

3.3.1 Proposed Modifications

This section discusses the significance of position-based operator and neighborhood operator in detail.

3.3.1.1 Position based Operator

This operator is inculcated into synthesis reaction of ACRO algorithm. This operator generates a new reactant for synthesis work and generated reactant is differ from previous one. This operator selects two reactants, suppose X_i and X_j in random order from population for generating the new reactant i.e. $X_{i,\,new}$. The new reactants consist of equal portion of participating reactants. The above process can be explained as below.

Let’s assume two reactants-

$$X_i = [2, 5, 6, 8,11]$$

$$X_j = [9, 4, 3, 5, 10]$$

$$X_{i,\,new} = [5, 8, 11, 4, 5, 10]$$

$X_{i,\,new}$ represents the new reactant and it is generated through X_i and X_j reactants. Both of reactants are equally participate to compute new reactant. The new reactant contains six elements, out of six, three elements are selected from reactant X_i, whereas, rest of are selected form reactant X_j.

3.3.1.2 Neighborhood Operator

The objective of this operator is to tackle the local optima situation of algorithm. It can be described as during the execution of algorithm; same reactants are generated through chemical reactions. So, the search area is not explored significantly for global optimum solution and algorithm will converge on local optimum. It is also interpreted as lack of diversity in population during the execution of algorithm. Hence, to deal with the local optima situation, diverse reactant will be generated. In this work, a neighborhood operator is adopted for the same. The working of neighborhood operator is described as. Suppose, a reactant X_i = [2, 5, 6, 8,11] is responsible for local optima problem. This operator selects two elements in the current reactant and interchange its position. Suppose, 2 and 8 are selected for the same. Now position of 2 and 8 are interchanged and the new reactant is [8, 5, 6, 2, 11].

3.3.2 Proposed ACRO Clustering Algorithm

The working of proposed ACRO clustering algorithm divides into three phases- Initialization Phase, Chemical reaction phase and Update, and Decision-making Phase.

3.3.2.1 Initialization phase: In this phase, various parameters of proposed ACRO viz. number of reactants (k), iterations and other algorithmic parameters are initialized. The dataset is loaded into memory. The number of centroids is defined and initial cluster centers are selected. Initially, two reactants $\mathrm{X_1} = \left\{\mathrm{a_{i,1}, a_{i,2}, \ldots, a_{i,d}}\right\}$ and $\mathrm{X_2} = \{\mathrm{a_{j,1}, a_{j,2}, \ldots, a_{j,d}}\}$ are selected from the population pool and further, rets of reactants are generated. Assume, k=2; then, two reactants $\mathrm{X_3}$and $\mathrm{X_4}$ are computed through initially selected reactants $\mathrm{X_1}$ and $\mathrm{X_2}$using equations 3.1-3.2.

$$\mathrm{X_3} = \left\{\mathrm{r * a_{i,1}, r * a_{i,2}, \ldots, r * a_{i,\frac{d}{2}};\ r * a_{\mathit{j},\frac{d}{2}+1}, r * a_{\mathit{j},\frac{d}{2}+2}, \ldots, r * a_{\mathit{j},d}}\right\} \tag{3.1}$$

$$\mathrm{X_4} = \left\{\mathrm{r * a_{j,1}, r * a_{j,2}, \ldots, r * a_{\mathit{j},\frac{d}{2}};\ r * a_{i,\frac{d}{2}+1}, r * a_{i,\frac{d}{2}+2}, \ldots, r * a_{i,d}}\right\} \tag{3.2}$$

Where d is length (dimension) of reactant, r is a random number, $\mathrm{a_i}$ and $\mathrm{a_j}$ denotes ith and jth reactants. If k > 2; equations 3.3-3.8 are used to generate reactants.

$$\mathrm{R_5} = \left\{\mathrm{r * a_{i,1}, r * a_{i,2}, \ldots, r * a_{i,\frac{2d}{3}};\ r * a_{j,2d/3+1}, r * a_{j,2d/3+2}, \ldots, r * a_{j,d}}\right\} \tag{3.3}$$

$$\mathrm{R_6} = \left\{\mathrm{r * a_{i,1}, r * a_{i,2}, .., r * a_{i,\frac{d}{3}}; r * a_{j,d/3+1}, r * a_{j,2d/3}, r * a_{i,2d/3+1}, .., r * a_{j,d}}\right\} \tag{3.4}$$

$$R_7 = \left\{r * a_{i,1}, r * a_{i,2}, \ldots, r * a_{i,\frac{d}{3}};\ r * a_{j,d/3+1},, \ldots, r * a_{j,d}\right\} \tag{3.5}$$

$$R_8 = \left\{r * a_{j,1}, r * a_{j,2}, \ldots, r * a_{j,\frac{d}{3}};\ r * a_{i,d/3+1}, \quad \ldots, r * a_{i,d}\right\} \tag{3.6}$$

$$R_9 = \left\{r * a_{j,1}, r * a_{j,2}, \ldots, r * a_{j,\frac{d}{3}};\ r * a_{i,\frac{d}{3}+1}, r * a_{i,\frac{2d}{3}}, r * a_{j,2d/3+1} \ldots, r * a_{i,d}\right\} \tag{3.7}$$

$$R_{10} = \left\{r * a_{j,1}, r * a_{j,2}, \ldots, r * a_{j,\frac{2d}{3}};\ r * a_{i,\frac{2d}{3}+1}, \quad r * a_{i,\frac{2d}{3}+2}, \ldots, r * a_{i,d}\right\} \tag{3.8}$$

3.3.2.2 Chemical reaction phase: The work of this phase is to allocate data objects to different clusters. An objective function is adopted for allocating data objects into clusters. Here, Euclidean distance is considered as objective function equation 1.1. The data objects are allocated to clusters using minimum Euclidean distance. Further, the chemical reactions are adopted for transformation of reactants and generating new compounds or reactants. In this work, bimolecular and monomolecular reactions are implemented to generate optimal solution.

1. Bimolecular reaction:

   Suppose, $X_1 = \{x_{i,1}, x_{i,2}, \ldots, x_{i,d}\}$ and $X_2 = \{x_{j,1}, x_{j,2}, \ldots, x_{j,d}\}$ are two reactants that can participate in a bimolecular reaction (synthesis reaction, displacement reaction, redox2 reaction)

   - Synthesis reaction

   Synthesis reaction combines two or more existing reactants to produce a new reactant. Let $X = (X_1, X_2, X_3, \ldots, X_i, X_j, X_k, \ldots, X_n)$ is a pool of reactant from which reactants are selected to participate in synthesis reaction. The new reactant $X_{i,new}$ is obtained using equation 3.9.

$$X_{i,new} = X_i + \lambda_i\left(X_j - X_i\right) \tag{3.9}$$

   Where X_i and X_j are randomly selected reactants and λ_i is a random value [0.25, 1.25].

- Displacement reaction.

  In displacement reaction, anion and cations of two different reactants interchanges positions and forms two different reactants. This reaction work in following manner. Let, $X_k = (X_1, X_2 \ldots X_i, X_j, \ldots, X_K)$ where, k=1, 2.…….K.

$$X_{i,new} = X_i\left(1 - \lambda_{td} X_j\right) \tag{3.10}$$

$$X_{j,new} = \lambda_{td}X_j + (1 - \lambda_{td}X_i) \quad (3.11)$$

Where$\lambda_{td} \in \{0,1\}$ and $\lambda_{td+1} = 2.3\ (\lambda_{td})^{2\sin(\pi\lambda_{td})}$; where, in λ_{td}, suffix 'td' is incremented using 1, when reaction is accomplished.

- Redox2 reaction

  This reaction act as an evolutionary reaction. If X_i is the reactant with better fitness value, then update the reactant using equation 3.12

$$X_{i,new} = \lambda_{tr}(X_i - X_j) + X_i \quad (3.12)$$

  where, $\lambda_{tr} \in \{0,1\}$ and value of λ_{tr+1} is computed using following equation 3.13.

$$\lambda_{tr+1} = \begin{cases} 0 & \lambda_{tr} = 0 \\ \dfrac{1}{\lambda_{tr}\mathrm{mod}(1)} & \lambda_{tr} \in (0,1) \end{cases} \quad (3.13)$$

$$1/\lambda_{tr}\mathrm{mod}(1) = \frac{1}{\lambda_{tr}} - \left\lfloor \frac{1}{\lambda_{tr}} \right\rfloor \quad (3.14)$$

2. Monomolecular reaction

- Decomposition reaction

This reaction is reverse of synthesis reaction. In this reaction, a single compound is divided into two or more compounds. Suppose $X = (X_1, X_2 \ldots X_i, X_j, \ldots, X_n)$ be the reactant and $C_k \in \{X_m, X_n\}$ be an atom. The new atom of the molecule $X_{i,new}$ is randomly selected reactant from population or reactant from $\{X_m, X_n\} \in C_k$, but $\{X_m, X_n\} \neq (X_i \text{ and } X_j)$.

- Redox1 reaction

Redox1 reaction generate the reactants using equation 3.15.

$$X_{i,new} = X_m + \lambda_t(X_n - X_m) \quad (3.15)$$

Where,$\lambda_t \in \{0,1\}$ such that initial $\lambda_0 \in \{0.0, 0.25, 0.5, 0.75, 1.0\}$ and $\lambda_{t+1} = 4\lambda_t(1 - \lambda_t)$. When, reaction is completed, "t" is updated by 1.

3.3.2.3 Update and decision-making phase: In this phase, an equilibrium test is performed and reactants with better fitness are considered. This step also deals with local optima situation, if local optima condition arise new solution are generated with neighborhood operator. After this the reactants for next iteration are updated. If the termination criteria is achieved algorithm stops its execution, else repeat phase 2-3. Figure 3.1 illustrates the flow chart of ACRO.

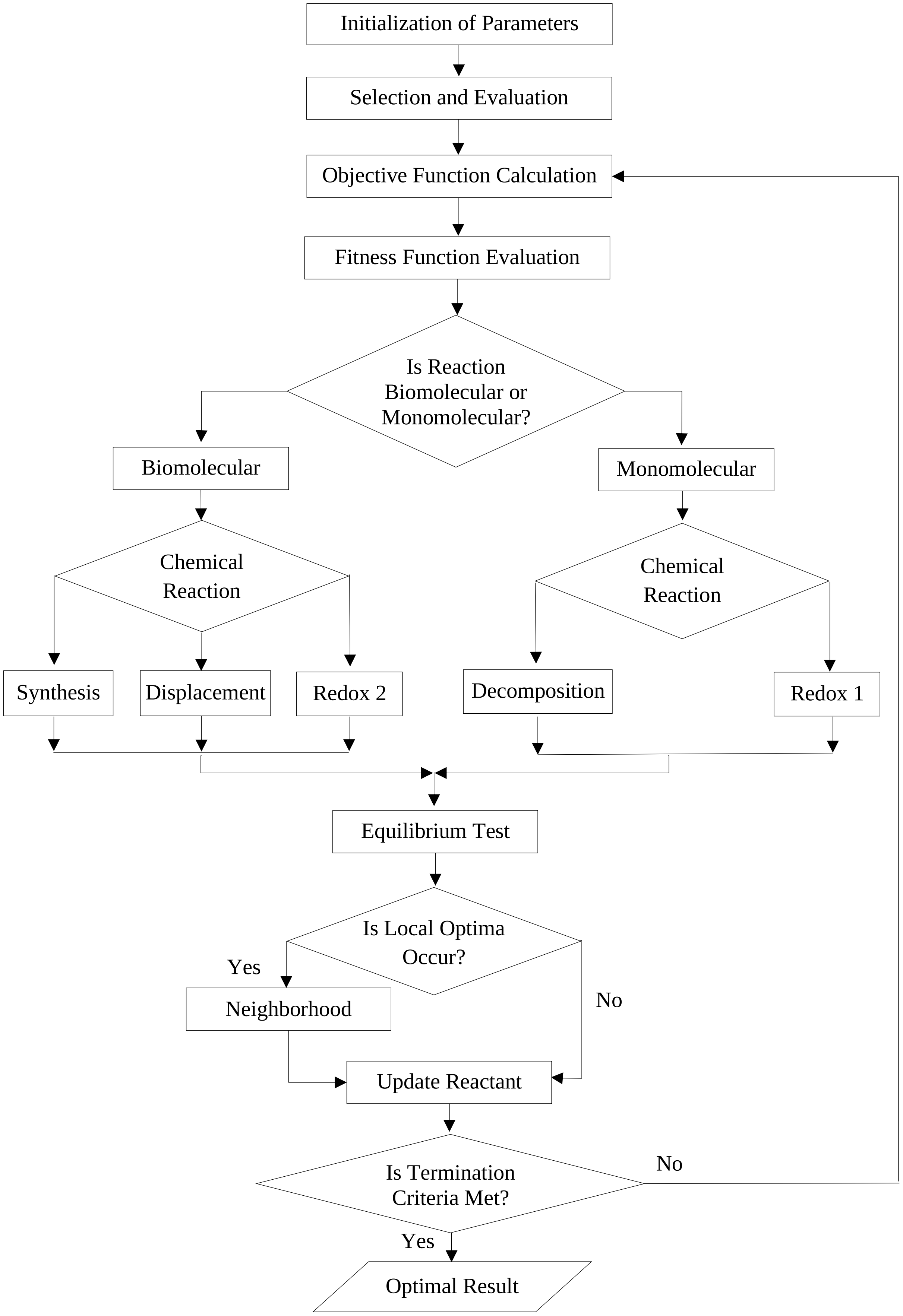

Figure 3.1: Flowchart of ACRO

The steps of proposed ACRO is summarized in Algorithm 3.1.

Algorithm 3.1: Pseudo code of ACRO algorithm for clustering problems

Input: Dataset and number of clusters (K).

Output: Optimized cluster centers.

```
1: Initialize parameters of ACRO clustering algorithm viz number of chemical reactants, iteration number etc.
2: Initialize the population (reactants) using equations 3.1-3.8.
3: Compute the objective function using equation 1.1.
4: Rearrange data objects into different clusters according to minimum value of objective function and compute the fitness function for each reactant i.e. cluster center using equation 3.16.
       $\mathrm{Fitness}\left(\mathrm{R_p}\right) = \sum_{j\in 1}^{K} \frac{\mathrm{SSE}(\mathrm{R_p})}{\sum_{j=1}^{K} \mathrm{SSE}(\mathrm{R_p})}$    (3.16)
5: Apply the chemical reactions phase of the algorithm.
   If (flag==0)          // Bimolecular reactions
   switch (index)
           case 1: Generate the new reactant using synthesis reaction equation 3.9.
           case 2: Generate the new reactants using displacement reaction equations 3.10-3.11.
           case 3: Generate the new reactants using redox2 reaction equation 3.12-3.14.
   If (flag==1)     //Monomolecular reactions:
   switch (index)
           case 1: Generate the new reactant using decomposition reaction.
           case 2: Generate the new reactant using redox1 reaction equation 3.15.
6: Perform equilibrium test to update the reactants.
7: If (local optima occur), Apply neighborhood operator.
8: Update the reactant.
9: Check termination criteria, If met stop execution, else repeat the steps 3-8.
```

SSE is the sum of squared intra-cluster Euclidean distances, $\mathrm{R_p}$ is reactant and K is number of clusters.

3.4 Simulation Results

The section discusses the experimental results of ACRO algorithm. MATLAB 2016 environment is used for implementing ACRO and the system configuration is Intel Core i5 processor, 8GB RAM and Windows 10 based operating system. The parameters of ACRO and other algorithms are mentioned in Table 3.1.

Table 3.1: Description of algorithms parameter

K-means		GA		CSO		PSO	
Population	K × d	Population	K × d	Population	K × d	Number of swarms	10 × K × d
		Crossover rate	0.8	SMP	10	c 1 = c 2	2
		Mutation rate	0.001	MR	0.5	ω min	0.5
				C	2	ω max	1
ACO		BA		ACRO			
Number of ants	50	Population	K × d	Population		K × d	
Threshold Probability	1	A_0	0.9	Dividing factor (k)		2	
Searching probability	0	R	0.1	λ_{tr} and λ_{td}		{0,1}	
Evaporation rate	0	α = γ	0.9	λi		{0.25, 1.25}	
Maximum number of iterations (T_{max}) = 200							

3.4.1 Results

This section discusses the experimental results of ACRO algorithm and other standard clustering algorithms. Eight real life datasets are adopted for evaluating the efficiency of ACRO algorithm. These datasets are Iris, Cancer, Letter-Recognition (LR), CMC, Wine, ISOLET, Statlog, and Glass, and taken from UCI repository (https://archive.ics.uci. edu/ml/datasets.php). Table 3.2 gives the detail information regarding these datasets.

Table 3.2: Information of datasets used in this work.

Datasets	K	D	N	Description
Iris	3	4	150	Fisher's iris data
Cancer	2	9	683	Cancer data
CMC	3	9	1,473	Contraceptive method choice
Wine	3	13	178	Wine data
Glass	6	9	214	Glass identification data
Statlog	7	9	58,000	Statlog (Shuttle)
LR	26	16	20,000	Letter-Recognition
ISOLET	26	617	7797	ISOLET

The simulation results are assessed using intra-cluster distance and f-measure parameters. Several standard clustering algorithms like K-means, GA, PSO, ACO, CSO and BA are selected for comparing the experimental results of ACRO algorithm [15,31-35,83-84,87-89]. Tables 3.3 presents the experimental results of ACRO and other clustering algorithms using eight real-life datasets. For every dataset, the algorithms run thirty times individually, and each run consists of 200 iterations. It is observed that ACRO algorithm having minimum intra-cluster distance for all datasets except glass dataset. It showed that data objects are tightly bound with clusters. Further, the clusters are also in compact. For glass dataset, K-means algorithm achieves minimum intra-cluster distance. For rest of datasets, it is noticed that K-means exhibits worst results for Wine, CMC, Statlog and LR datasets. Whereas, GA exhibits worst results for Iris, Cancer, Glass and ISOLET datasets. Further, f-measure is also computed for investigating the efficiency of ACRO in clustering filed. It is also noted that most of algorithms correctly classified the data objects to corresponding clusters. The ACRO algorithm obtains better f-measure rate for all algorithms except glass and LR datasets. For glass dataset, BA achieves higher f-measure rate, whereas, GA achieves higher f-measure rate for LR dataset. Moreover, it is observed that GA having lower f-measure rate for most of datasets except Statlog, LR and ISOLET.

Table 3.3: Simulation results of proposed ACRO, K-means, GA, PSO, ACO, CSO and BA algorithms using intra cluster distance and f-measure

Sr No	Parameters	Algorithms						
		K-means	GA	PSO	ACO	CSO	BA	ACRO
Iris	Best case	97.52	113.98	97.05	97.21	96.98	96.84	95.56
	Avg. case	113.56	125.19	98.73	98.36	97.64	97.53	96.73
	Worst case	125.23	139.77	99.89	99.59	98.78	98.09	97.48
	F-measure	0.781	0.774	0.78	0.778	0.781	0.782	0.785
Cancer	Best case	2989.46	2999.32	2978.68	2983.49	2985.16	2972.36	2912.66
	Avg. case	3248.25	3249.46	3116.64	3178.09	3124.15	3098.93	3063.34
	Worst case	3566.94	3427.43	3358.43	3292.41	3443.56	3282.75	3179.25
	F-measure	0.832	0.819	0.826	0.829	0.831	0.833	0.835
CMC	Best case	5834.21	5705.63	5792.48	5756.42	5712.78	5689.16	5681.56
	Avg. case	5912.46	5756.59	5846.63	5831.25	5804.52	5778.14	5746.32
	Worst case	5983.06	5812.64	5936.14	5929.36	5921.28	5914.25	5894.63
	F-measure	0.337	0.324	0.333	0.332	0.334	0.336	0.339
Wine	Best case	16775.32	16490.41	16424.26	16456.81	16429.54	16372.02	16248.23
	Avg. case	18059.91	16530.53	16491.52	16526.12	16486.21	16556.89	16334.85
	Worst case	18783.23	16590.53	16589.13	16621.44	16595.45	16557.76	16396.56
	F-measure	0.520	0.515	0.517	0.521	0.522	0.523	0.526
Glass	Best case	222.43	272.37	264.56	273.22	256.53	256.47	261.47
	Avg. case	246.51	282.32	278.71	281.46	264.44	269.61	266.23
	Worst case	258.38	291.77	283.52	286.08	282.27	278.24	274.14
	F-measure	0.426	0.333	0.412	0.402	0.416	0.431	0.428

Statlog	Best case	812090400	793000800	522200060	542208805	513208000	448208805	440010006
	Avg. case	812558906	793000994	522200928	542216190	513208164	450769448	440813720
	Worst case	813000125	793001140	522209000	542229001	513219100	452300087	441003000
	F-measure	0.262	0.314	0.322	0.329	0.312	0.316	0.399
LR	Best case	620900	610000	608000	608000	610000	612000	604550.04
	Avg. case	624765.58	611731.68	608470.77	608495.87	611102.88	613775.68	604612.52
	Worst case	626775.18	613600	609054.11	608786.61	612027.05	615000	604691.47
	F-measure	0.461	0.488	0.412	0.427	0.416	0.439	0.441
ISOLET	Best case	446201.02	460280.78	450493.89	454350.19	447176.93	441222.8	440906.04
	Avg. case	446502.65	460851.88	451718.88	455837.78	447733.55	442361.25	441268.61
	Worst case	446905	462196.28	453961.88	458270.68	448585.87	443202.55	441942.04
	F-measure	0.361	0.332	0.392	0.301	0.311	0.369	0.408

Figures 3.2-3.9 shows the convergence behavior of BA, CSO, ACO, PSO, GA, K-means and proposed ACRO clustering algorithm. In these figures, horizontal axis denotes number of iterations, whereas, vertical axis denotes intra-cluster distance. It is seen that proposed ACRO clustering algorithm converges on minimum values except for LR and glass datasets. However, ACRO provides better convergence rate for maximum datasets. Hence, it is concluded that ACRO algorithm outperforms than other clustering algorithms.

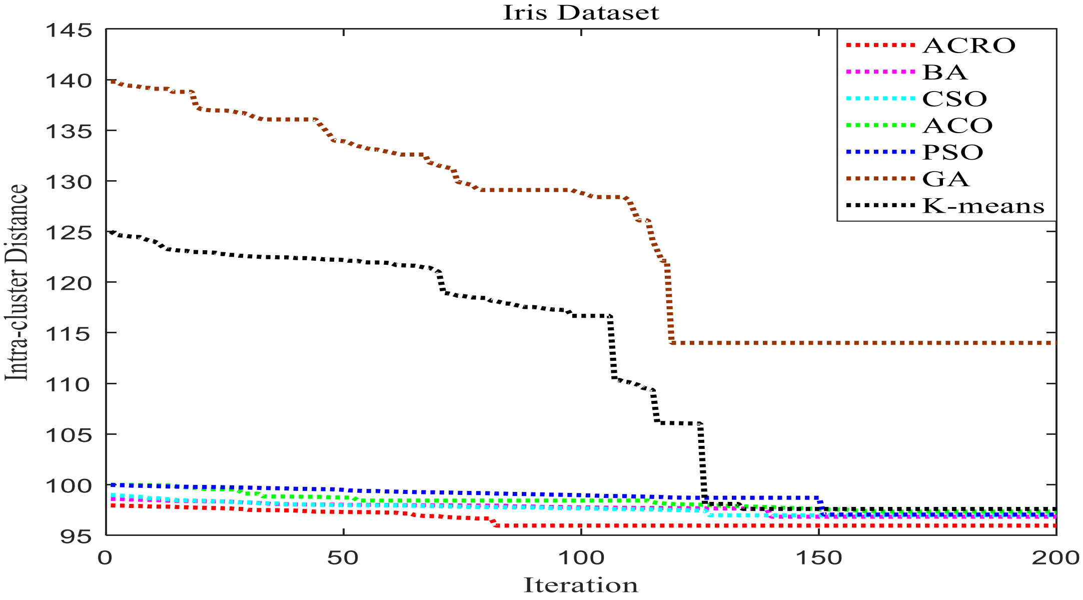

Figure 3.2: Convergence behavior using iris dataset

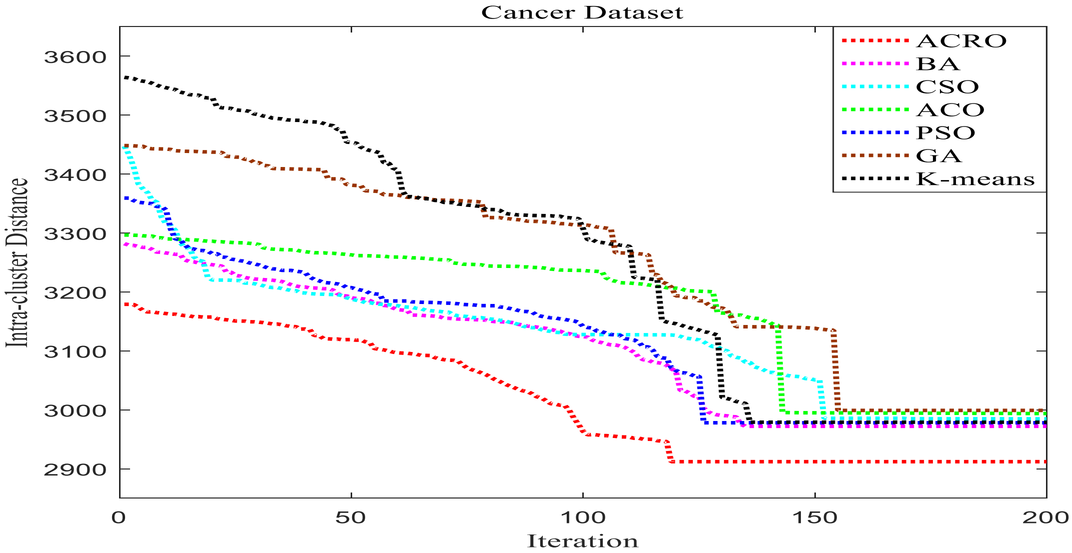

Figure 3.3: Convergence behavior using cancer dataset

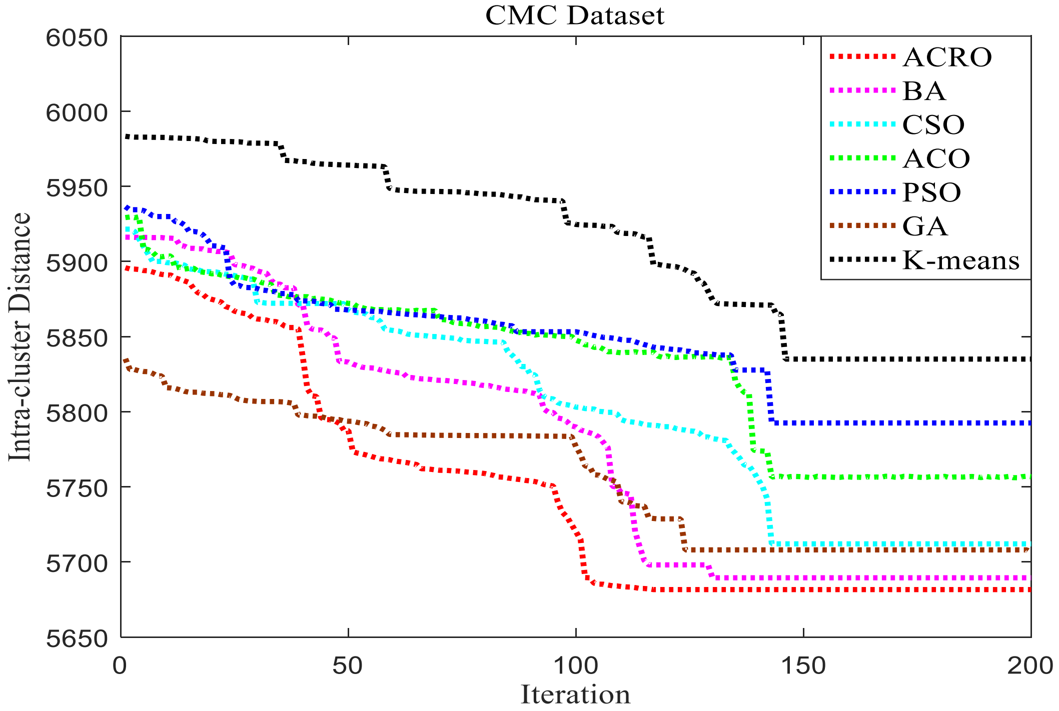

Figure 3.4: Convergence behavior using CMC dataset

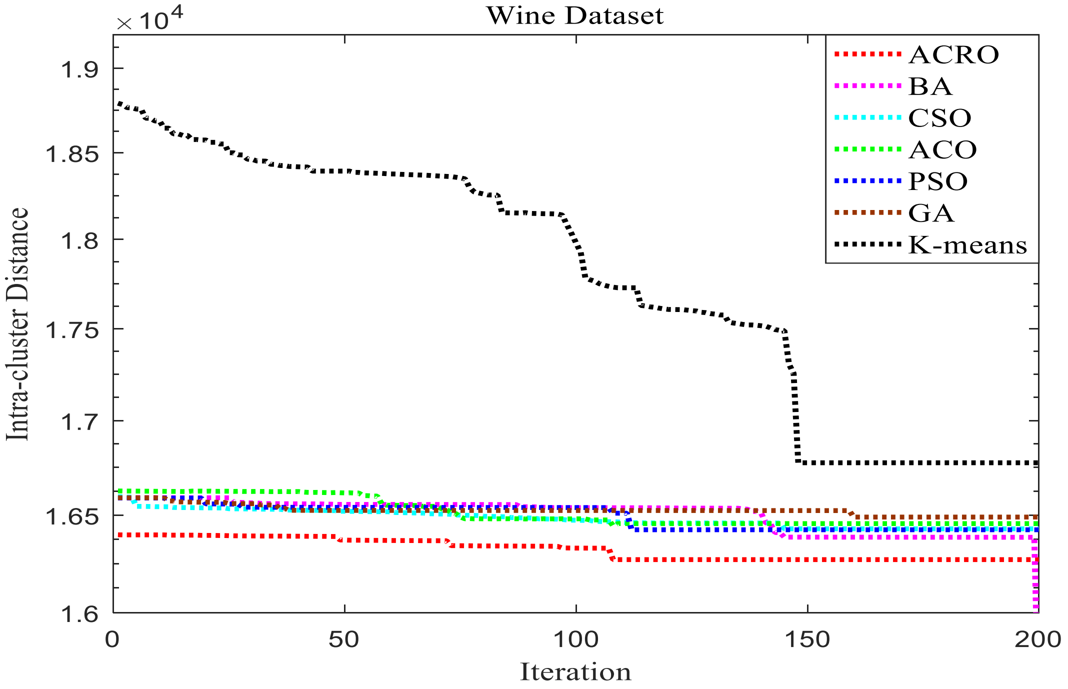

Figure 3.5: Convergence behavior using wine dataset

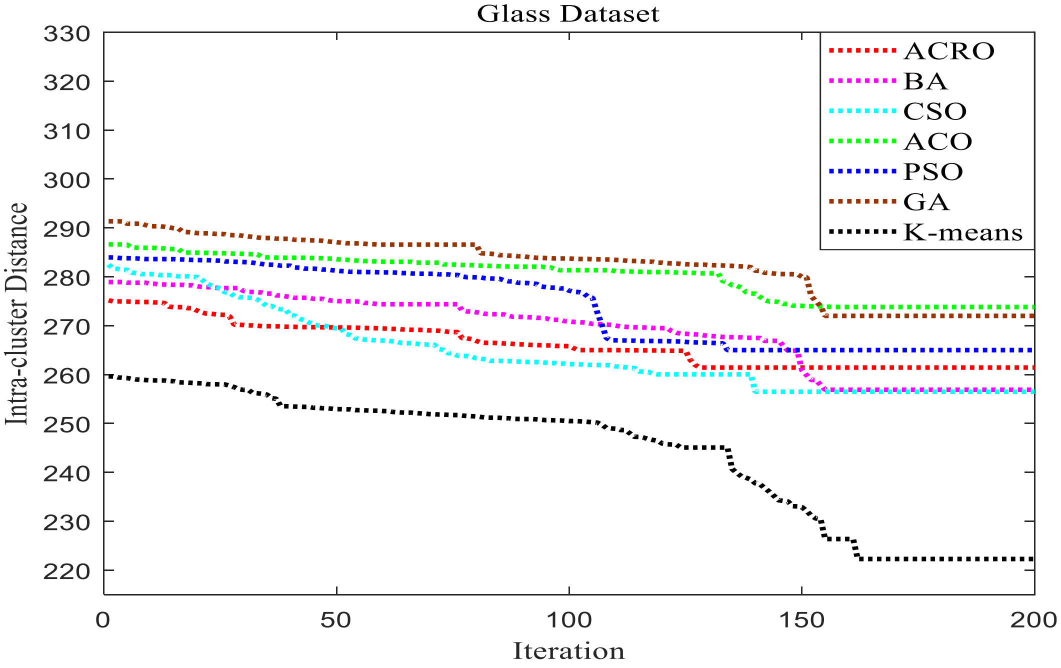

Figure 3.6: Convergence behavior using glass dataset

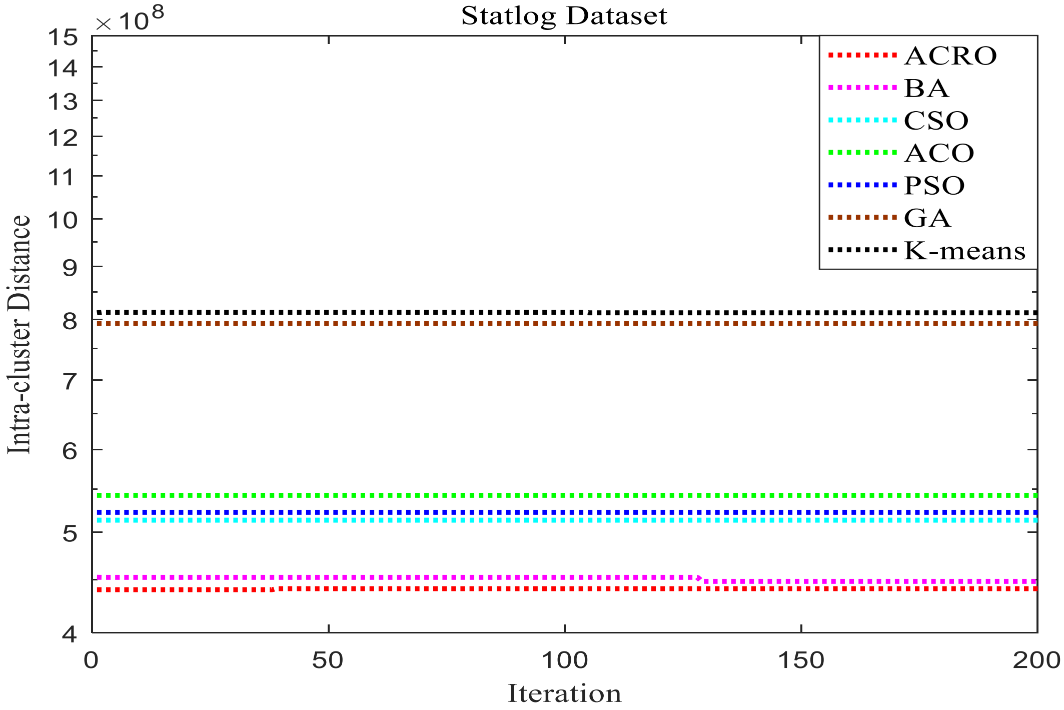

Figure 3.7: Convergence behavior using statlog dataset

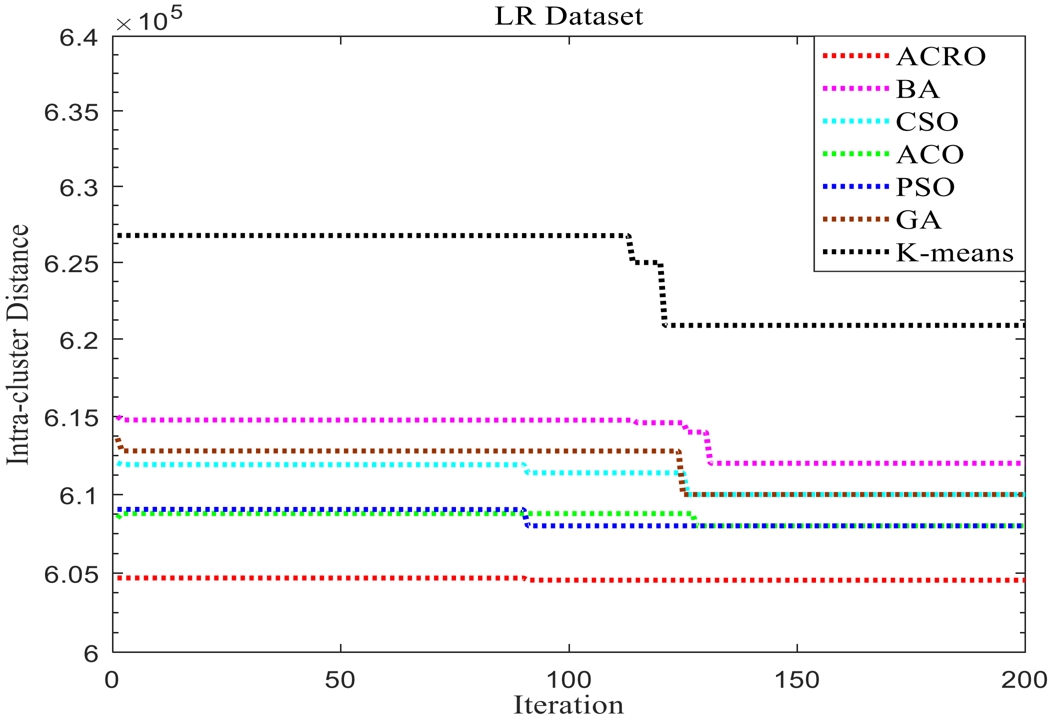

Figure 3.8: Convergence behavior using LR dataset

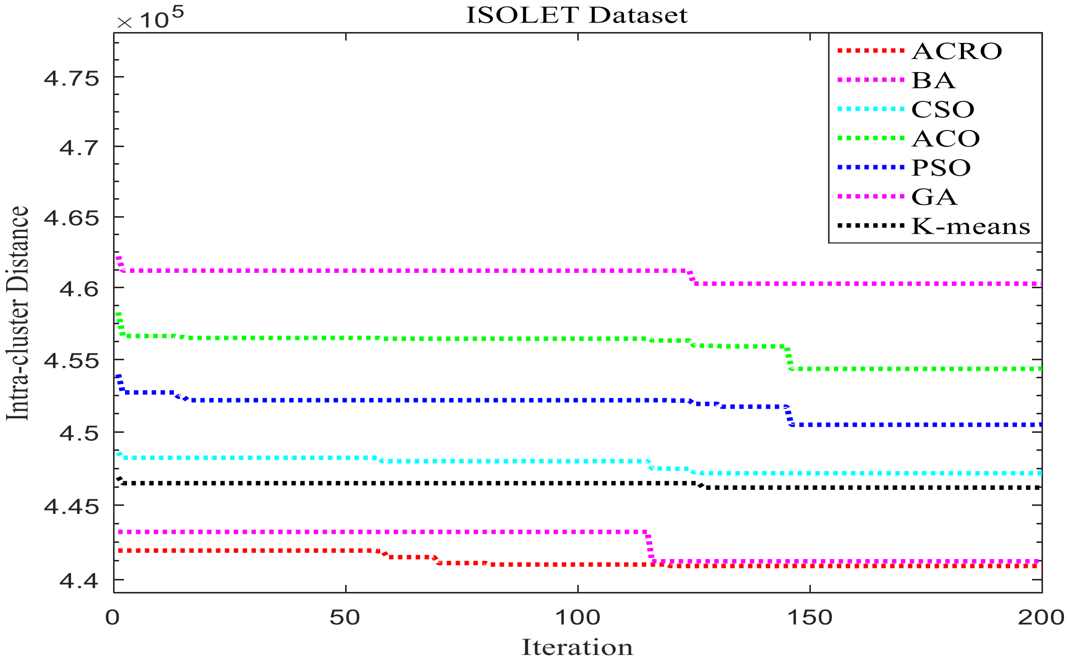

Figure 3.9: Convergence behavior using ISOLET dataset

Figures 3.10-3.17 shows the clustering of data objects using ACRO algorithm. Figure 3.10 illustrates the clustering of data objects presented in iris dataset. It is noted that petal length, sepal width and sepal length attributes are selected to demonstrate the clustering of data objects. The corresponding clusters are setosa, versicolour and virginica. The proposed algorithm is capable to allocate the data objects into different clusters. Figure 3.11 shows the clustering of data objects presented in cancer dataset using cell size, cell shape and bare nuclei attributes. This dataset consists of two clusters i.e. malignant and benign. ACRO algorithm allocates data objects to different clusters in effective manner. Figures 3.12 demonstrate the clustering of data objects presented in CMC dataset using ACRO based clustering algorithm. CMC dataset is divided into three clusters as ‘Cluster No use1’, ‘Cluster Long Term2’ and ‘Cluster Short Term3’. Further, it is concluded that data objects are non-linearly inseparable. Figure 3.13 shows the clustering of data objects presented in wine dataset. The clustering is performed using alcohol, malic acid and ash attributes and wine dataset contains three clusters such as wine type 1, wine type 2 and wine type 3.

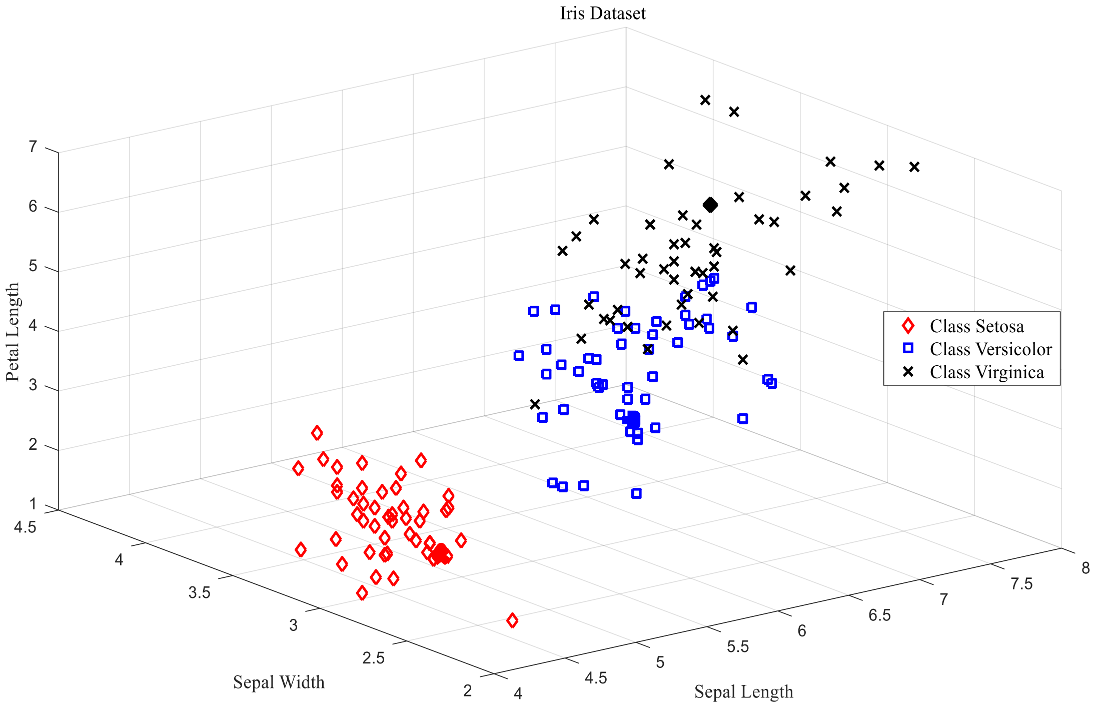

Figure 3.10: Clustering of iris data objects using proposed ACRO clustering algorithm

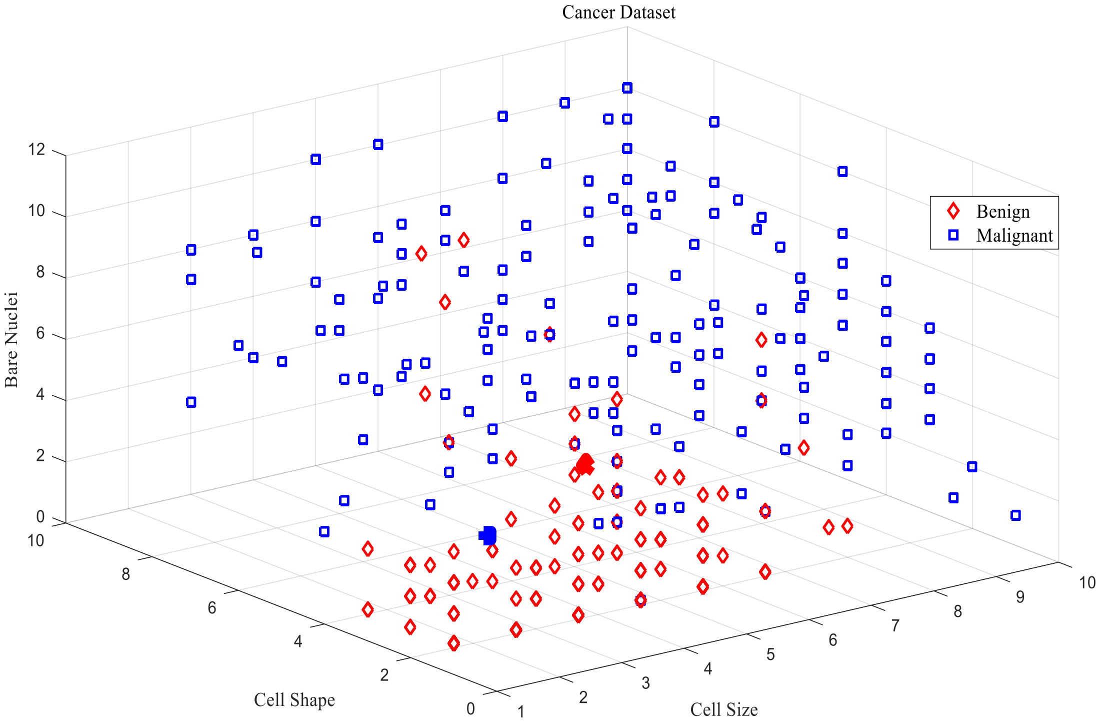

Figure 3.11: Clustering of cancer data objects using proposed ACRO clustering algorithm

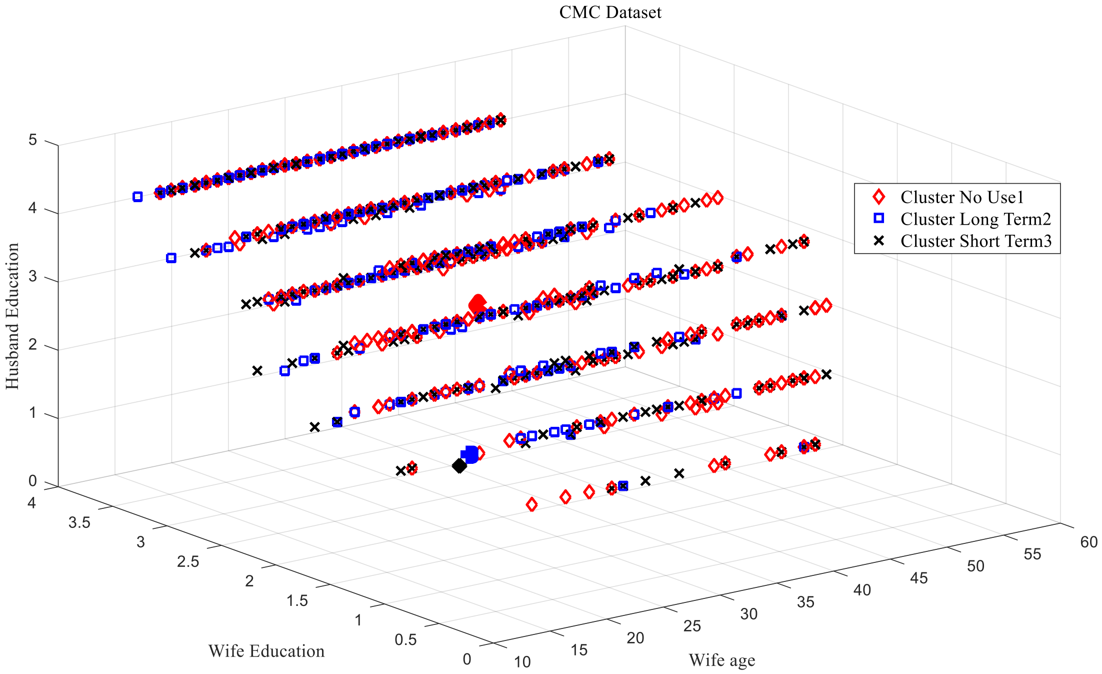

Figure 3.12: Clustering of CMC data objects using proposed ACRO clustering algorithm

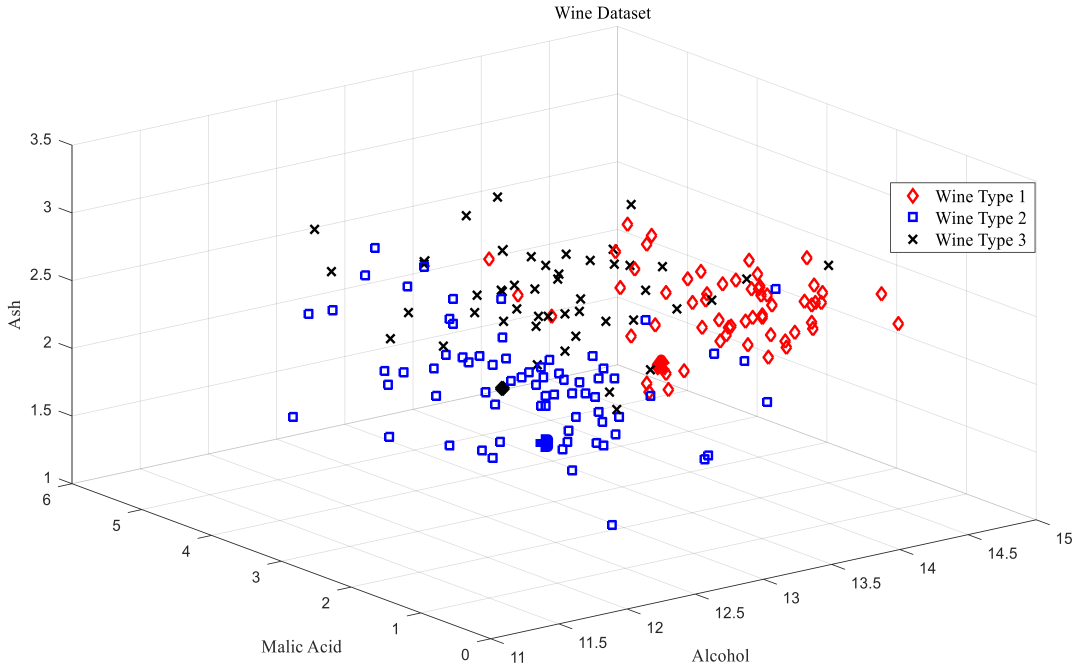

Figure 3.13: Clustering of wine data objects using proposed ACRO clustering algorithm

Figure 3.14 shows the clustering of data objects belong to glass dataset using proposed ACRO algorithm. The glass dataset contains six clusters it is observed that ACRO provides average case results for glass dataset. Figure 3.15 shows the clustering of the statlog(shuttle) dataset objects using ACRO algorithm. Data objects of statlog dataset are divided into seven clusters (Rad Flow, Fpv Close, Fpv Open, High, Bypass, Bpv Close and Bpv Open). It is observed that 80% of data objects belongs to cluster 'Rad Flow' and are linearly inseparable from other clusters. Figure 3.16 displays the clustering of letter recognition (LR) dataset objects using ACRO algorithm. Data objects of LR dataset are grouped into 26 clusters (A-Z), that are linearly inseparable. It is noticed that the rise in the number of clusters affects the performance of the proposed algorithm. It is concluded that the proposed ACRO algorithm provides average case results for LR dataset. Figure 3.17 depicts the clustering of the ISOLET dataset using ACRO algorithm. This figure displays the clustering results using real-valued attributes. Data objects of ISOLET datasets are divided into twenty-six clusters, that are linearly inseparable. Furthermore, it is noticed that the performance of the proposed algorithm is affected due to large number of clusters.

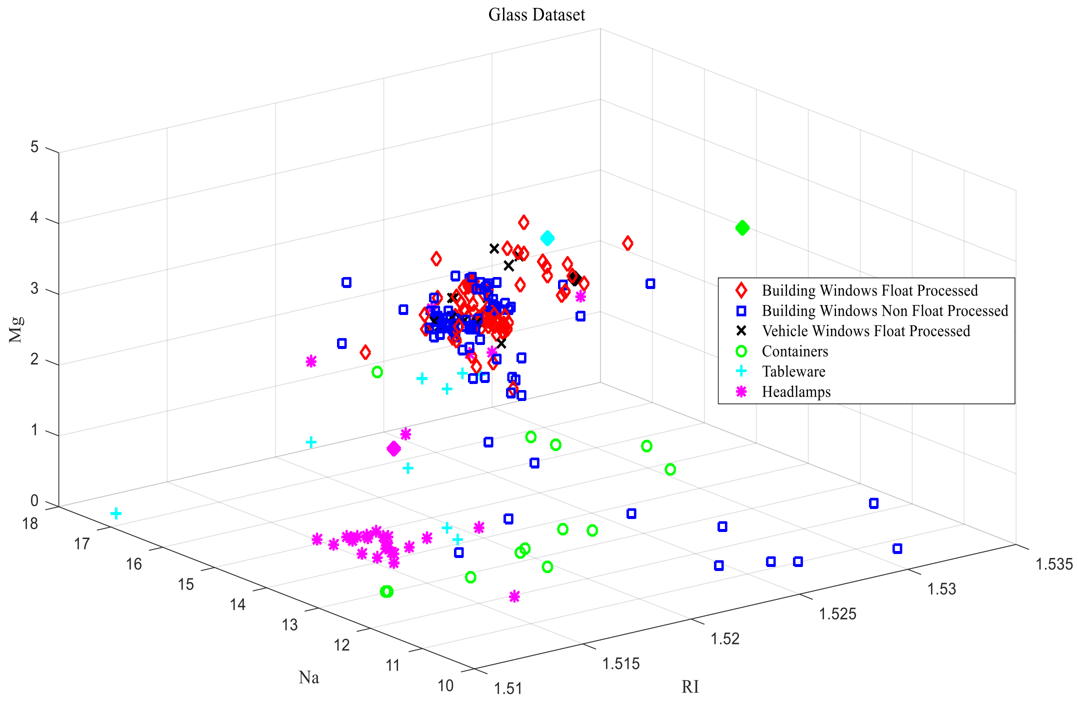

Figure 3.14: Clustering of glass data objects using proposed ACRO clustering algorithm

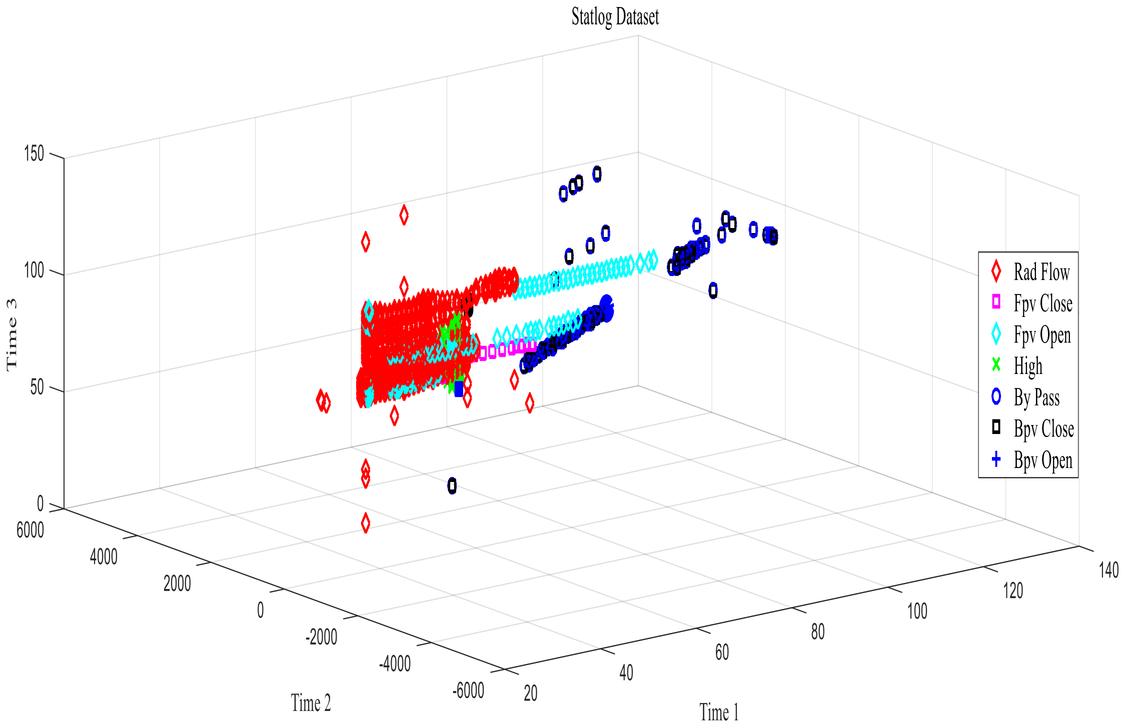

Figure 3.15: Clustering of statlog data objects using proposed ACRO clustering algorithm

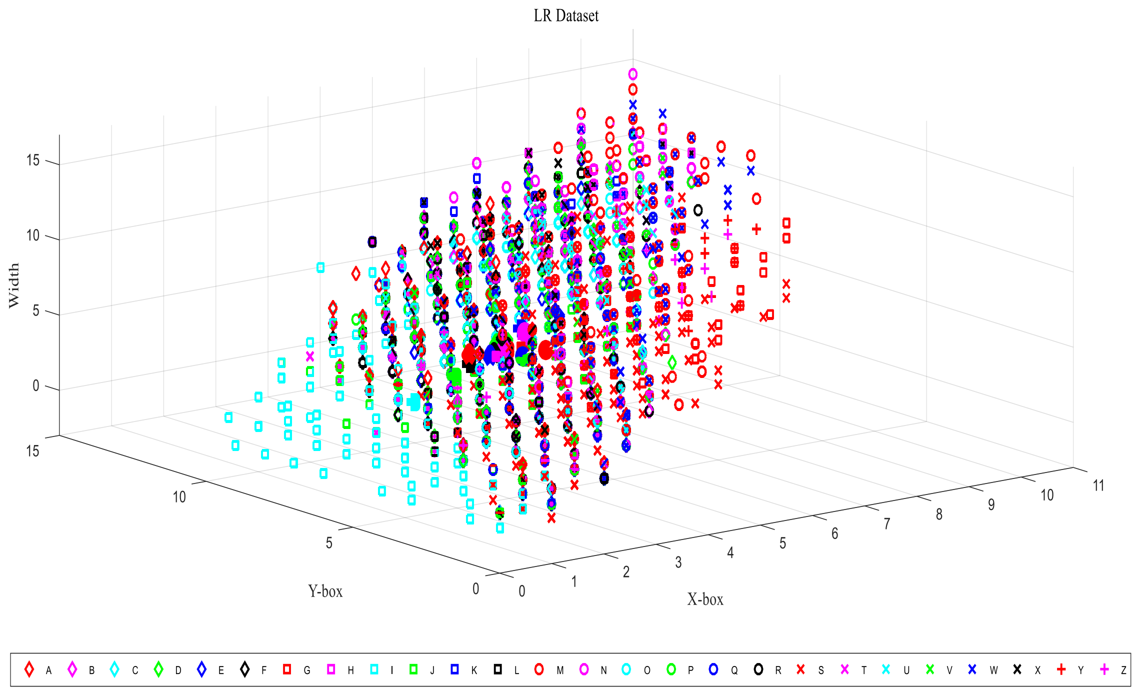

Figure 3.16: Clustering of LR data objects using proposed ACRO clustering algorithm

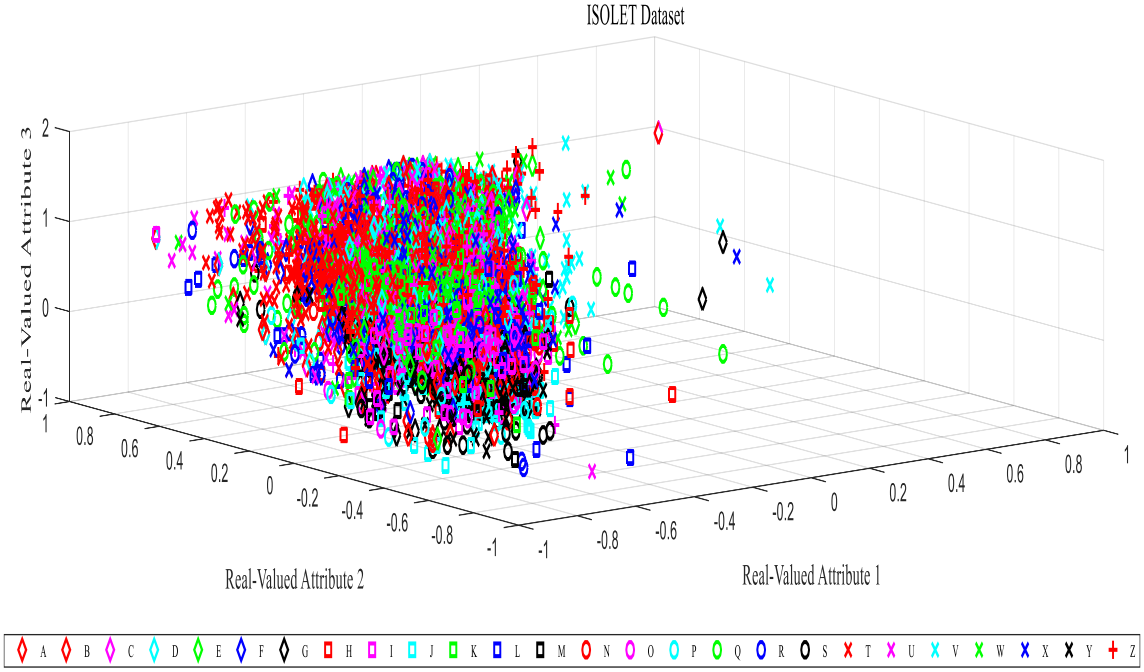

Figure 3.17: Clustering of ISOLET data objects using proposed ACRO clustering algorithm

3.4.2 Statistical Test

This subsection discusses the results of statistical test on ACRO algorithm. Statistical test also proves the importance an algorithm like simulation results. A statistical test confirms the existence of new algorithm and its significance in the research community. A Friedman statistical test is selected for proving the existence of newly proposed ACRO algorithm in the clustering field. The aim of this test is to prove that ACRO algorithm is statistical differ than other algorithms. The statistical test is applied on both of parameters.

Table 3.4 illustrates the results of statistical test using intra-cluster distance parameter. For, Friedman test, two hypotheses are designed; hypothesis (H_0) stated that algorithm having similar performance while, hypothesis (H_1) stated that algorithm having dissimilar performance. Its revealed that proposed ACRO clustering algorithm claims first rank (1.25) among all other algorithms. The critical value and p values are 12.5915 and 0.000038. Hypothesis (H_0) is strongly rejected and proposed algorithm have dissimilar performance in comparison to existing algorithms at the significance level 0.05. Hence, the ACRO algorithm is significantly different than other clustering algorithms.

Table 3.4: Statistics of Friedman test using avg. intra-cluster distance parameter

Datasets	Clustering Algorithms						
	K-means	GA	PSO	ACO	CSO	BA	ACRO
Iris	6	7	5	4	3	2	1
Cancer	6	7	3	5	4	2	1
CMC	7	2	6	5	4	3	1
Wine	7	5	3	4	2	6	1
Glass	1	7	5	6	2	4	3
Statlog	7	6	4	5	3	2	1
LR	7	5	2	3	4	6	1
ISOLET	3	7	5	6	4	2	1
Sum	44	46	33	38	26	27	10
Rank	5.5	5.75	4.13	4.75	3.25	3.38	1.25

Number of observations: 56	Number of problems: 08	Number of algorithms: 7
Sum of squares of rank sums: 8090	Correction factor: 896	Friedman test statistic: 24.69
Degree of freedom: 6	p-value: 0.000389	Critical value: 12.5915

Table 3.5 demonstrates the results of Friedman test using f-measure. ACRO clustering algorithm obtains first rank with most datasets except glass and letter recognition datasets. Moreover, GA algorithm exhibits poor performance using F-measure parameter. The critical value is 12.591587 and it is concluded that a significant difference is occurred. Hence, it is stated that ACRO is significantly different from other algorithms.

Table 3.5: Statistics of Friedman test using F-measure parameter

Datasets	Clustering Algorithms						
	K-means	GA	PSO	ACO	CSO	BA	ACRO
Iris	3.5	7	5	6	3.5	2	1
Cancer	3	7	6	5	4	2	1
CMC	2	7	5	6	4	3	1
Wine	5	7	6	4	3	2	1
Glass	3	7	5	6	4	1	2
Statlog	7	5	3	2	6	4	1
LR	2	1	7	5	6	4	3
ISOLET	4	5	2	7	6	3	1
Sum	29.5	46	39	41	36.5	21	11
Rank	3.69	5.75	4.88	5.13	4.56	2.63	1.38

Number of observations: 56	Number of problems: 08	Number of algorithms: 7
Sum of squares of rank sums: 8082.5	Correction factor: 896	Friedman test statistic: 24.55
Degree of freedom: 6	p-value: 0.000423	Critical value: 12.5915

3.5 Summary

This chapter explores the efficacy of new metaheuristic algorithm inspired from the chemical reaction processes, called ACRO for solving clustering problems. Few amendments are inculcated in ACRO algorithm to make more promising and robust. These amendments are position based operator and neighborhood operator. Some benchmark clustering datasets are adopted for evaluating the experimental results of ACRO clustering algorithm. The K-means, GA, PSO, ACO, CSO and BA clustering algorithms are selected for comparing the results of ACRO algorithm. The ACRO clustering algorithm achieves minimum intra-cluster distance and higher F-measure rate as compared to aforementioned algorithms. It is observed that proposed amendments significantly overcome the performance issues associated with ACRO algorithm such local optima and convergence rate. Although, it is also noticed that as number of clusters increases, the performance gets affected and average result are obtained.

CHAPTER 4

IMPROVED BB-BC ALGORITHM FOR CLUSTERING

4.1 Introduction

BB-BC algorithm is recent algorithm inspired from “universe evolvement theory” [92] and applied to solve global optimization problems. This algorithm consists of BB and BC phases. BB phase responsible to generate random points and acted as exploration phase and BC phase responsible to optimize the random points via the center of mass or minimum cost approach and acted as exploitation phase. From extensive literature survey, it is found that convergence rate and diversity are essential aspects of optimization process. The convergence rate can be stated as number of iterations required for obtaining the optimal solution, i.e. when the solution no longer changes. Whereas, diversity is related to the searching mechanism of the algorithm i.e. generation of diverse population during the execution of algorithm. Both of aspects contribute a significant role for obtaining good quality solution. Furthermore, Hatamlou et al. explored the capabilities of the BB-BC algorithm in clustering domain [20]. But, BB-BC algorithm suffers with lack of diversity and slow convergence rate [82]. Hence, this chapter addresses the convergence rate and diversity issues of BB-BC algorithm in clustering domain. For addressing the aforementioned issues, an improved BB-BC (IBB-BC) algorithm is developed. The convergence rate and diversification issues of traditional BB-BC algorithm addressed via chaotic maps and cellular automata-based concepts [79,93].

4.2 BB-BC Algorithm

Initially, BB-BC algorithm is considered for solving numerical optimization problems [92]. In this algorithm, BB phase responsible for generating new random points near center of mass. Whereas, BC phase responsible for optimizing the random points using center of mass. The center of mass (COM) is described in equation 4.1. Every execution of BB-BC algorithm consists of BB phase followed by BC phases and finally optimal solution is generated after specified number of iterations.

$$x^c = \frac{\sum_{i=1}^{N} \frac{x_i}{f^i}}{\sum_{i=1}^{N} \frac{1}{f^i}} \tag{4.1}$$

x^c represents COM, N is total number of points, f_i denotes fitness i^{th} points and $x_i$is i^{th} solution. The new point is computed using equation 4.2, after evaluation of COM.

$$x_{new}^{i} = x^c + \frac{lr}{k}\ i = 1,2, \dots N \tag{4.2}$$

In equation 4.2, “r” denotes random function, and l is defined as limit operator. The computational steps of algorithm are illustrated as follows.

Step 1: This step corresponds to set algorithmic parameters and selection of initial point from dataset in random order.

Step 2: This step evaluates the value of objective function. The objective function is problem dependent.

Step 3: BC Phase: New points are computed near COM using Equation 4.2.

Step 4: BC Phase: COM is computed using equation 4.1 and best solution is selected as COM.

Step 5: Termination Condition: If specified number of iterations is met, then stop the execution of algorithm, until, repeat steps 2-4.

Step 6: Obtain the optimal solution in terms of optimized points.

4.3 Improved BB-BC Algorithm

This section explains the improvements inculcated in BB-BC and working of IBB-BC algorithm.

4.3.1 Proposed Improvements

To address the convergence rate and diversification issues of BB-BC algorithm, chaotic maps and cellular automata-based concepts are introduced. The detailed description of these improvements are discussed in section subsections 4.3.1.1. and 4.3.1.2.

4.3.1.1 Chaotic maps

Chaotic maps are widely adopted to address the convergence rate issue of meta-heuristic algorithms [79]. In BB-BC algorithm, random numbers are generated through rand() function. The range of rand () function is ranging between 0 and 1. But, this function generates the random number in an improper sequence e.g. in first iteration, $rand()$ function can return 0.23, and in next iteration, it can return 0.9. This improper sequence of random numbers can affect the convergence rate of the algorithm. In contrast, chaotic maps follow a predefined sequence to generate random numbers and prove its competency over $rand()$ function. In this work, the chaotic logistic map is adopted to generate random numbers and it is described using equation 4.3.

$$c_{n+1} = ac_n(1 - c_n) \tag{4.3}$$

The c_{n+1} is a chaotic value and "a" is constant number, $0 < c_n < 1$.

$$x_j^{i,new} = x_j^c + c_n \frac{l}{k} \tag{4.4}$$

$$l = (\max(x_j) - \min(x_j))/2 \tag{4.5}$$

Where, $x_j^{i,new}$ is new centroid position, x_j^c is center of mass and $\max(x_j)$, $\min(x_j)$are maximum and minimum values, and k is number of clusters.

4.3.1.2 Cellular Automata

In BB-BC algorithm, new candidate solution is generated near to COM. So, it is said that exploration process of BB-BC explores area surrounding near to COM. Moreover, this algorithm uses a greedy approach to determine the best candidate solution. The greedy approach gives locally optimum solution, and further, global optimal solution is achieved through locally optimum solution. However, sometimes locally optimal solution cannot converge on globally optimal solution. Hence, a cellular automata-based approach is considered to improve diversification and obtain global optimum solution instead of greedy approach. The cellular automata provide an optimal solution through the interaction of multiple local best solutions. Every local solution interprets as cells, called state and next state computes through neighboring states i.e. neighboring local solutions. It is noticed that COM is the principle component of BB-BC algorithm. Each candidate solution is generated near to COM and further, COM is utilized to optimize the candidate solution. So, dimension candidate solution and number of instances are taken into consideration for implementing the cellular automata-based approach. Moreover, a heat function is also developed for each cell of automata. It maps the number of instances with class labels. It is also assumed that each cell having the information of counter and temperature. Temperature corresponds to heat function, whereas counter describes number of instances and entire search space is explored through the heat transfer rate. The steps of heat transfer function are heighted below.

- Heat function is computed using equation 4.6 for each cell.

$$c_{\mathrm{Heat},i} = \text{counter value} \quad i = 1,2 \ldots K \tag{4.6}$$

- Temperature is computed using equation 4.7.

$$c_{\mathrm{temp},i} = \ln\left(c_{\mathrm{Heat},i} + 1\right) \tag{4.7}$$

- Heat transfer rate is computed through equation 4.8.

$$c_{h\_t\_r} = c_{\mathrm{temp},1} + c_{\mathrm{temp},2} + \cdots + c_{\mathrm{temp},k} \tag{4.8}$$

Where, $c_{\mathrm{temp},i}$ and $c_{\mathrm{Heat},i}$ denotes temperature and heat values of i^{th} cell. The value of heat transfer function is used to select the appropriate value of center of mass.

4.3.2 Improved BB-BC Clustering Algorithm

This subsection demonstrates the working of IBB-BC algorithm. The working of IBB-BC algorithm divides into three phases. Figure 4.1. demonstrates the working of IBB-BC algorithm.

4.3.2.1 Initialization Phase: In this phase, user defined and other algorithmic parameters are initialized. The dataset is loaded in memory. The number of centroids is defined and initial cluster centers are selected using random selection method. A cost function is computed for measuring the similarity between data objects. Further, the data objects assign to different clusters based on minimum cost function value.

4.3.2.2 BB phase: is responsible for the generation of random points and also acted as exploration phase. This phase explores the search area for determining the good candidate solution. The equation 4.4 is used to determine the new candidate solutions. The efficacy of new candidate solutions is evaluated using cost function and data objects assigns to different clusters based on minimum cost function values.

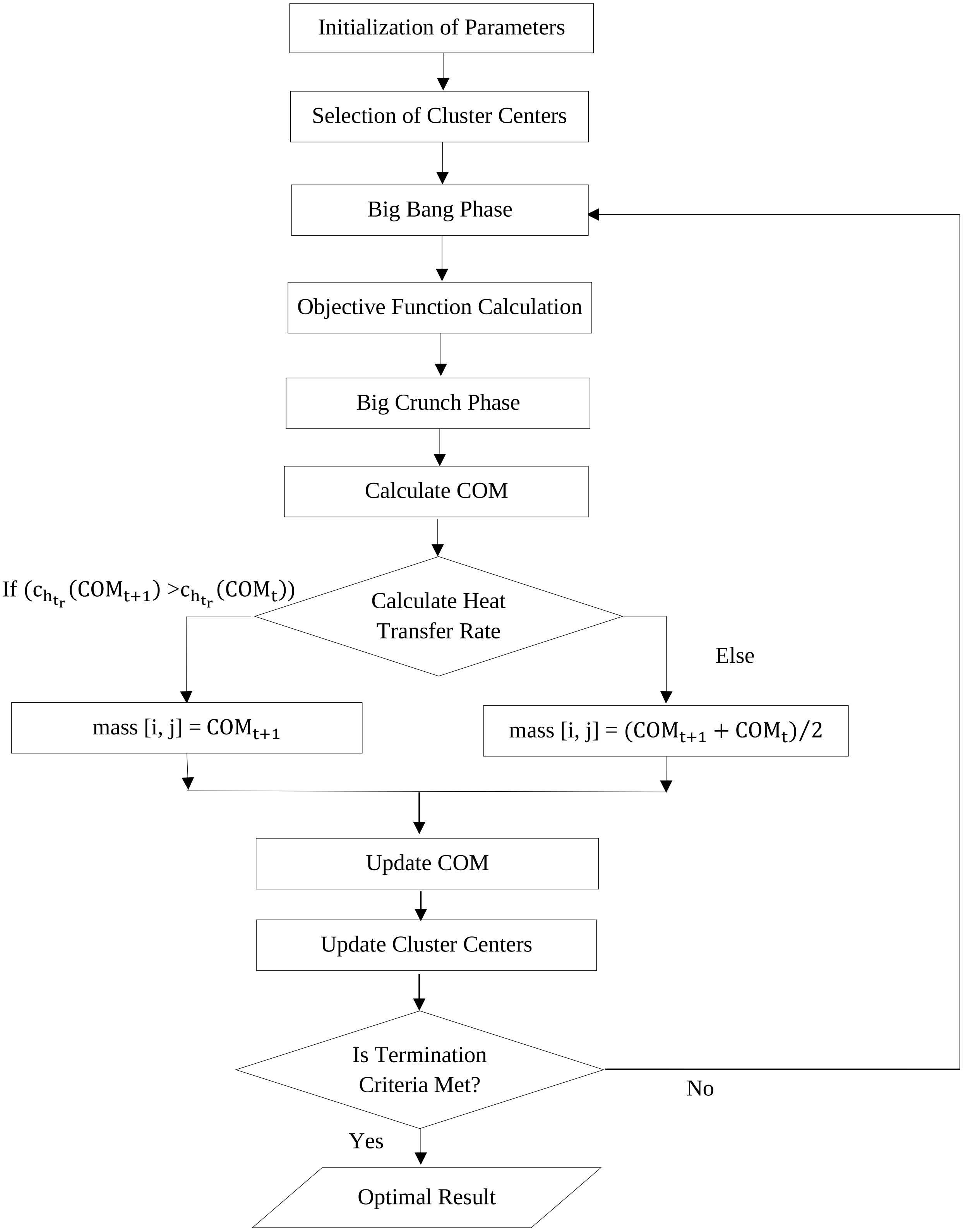
Initialization of Parameters
Selection of Cluster Centers
Big Bang Phase
Objective Function Calculation
Big Crunch Phase
Calculate COM
If ($c_{h_{t_r}}(\mathrm{COM}_{t+1})$ >$c_{h_{t_r}}(\mathrm{COM}_t)$)
Calculate Heat Transfer Rate
Else
mass [i, j] = COM_{t+1}
mass [i, j] = $(\mathrm{COM}_{t+1} + \mathrm{COM}_t)/2$
Update COM
Update Cluster Centers
Is Termination Criteria Met?
No
Yes
Optimal Result

Figure 4.1: Flowchart of IBB-BC Algorithm

4.3.2.3 BC phase:

BC phase responsible for the optimization of candidate solution and determines the global best solution. Further, COM is computed for each candidate solution using equation 4.1. This phase also consists of a cellular automata-based approach. The aim of this approach is to initiate the heat transfer function and it is measured for every candidate solution. If, value of heat function is larger than previous one, COM values moves to next iteration. Otherwise, an average value of COM is computed which is give as $\text{previous}\,(\text{COM}) + \text{current}\,(\text{COM})$. Further, the fitness of each candidate solution is determined and optimal candidate is selected for next iteration. The aforementioned process is continued, until optimum solution is not obtained. If, termination criterion is met, stop the execution of algorithm and obtain the final solution. Otherwise, phases 2-3 are repeated. The pseudo code of the Improved BB-BC clustering algorithm is summarized as

Algorithm 4.1: Pseudo code of IBB-BC clustering algorithm

Input: Dataset and number of clusters (K).

Output: Optimized cluster centers.

```
Set parameters of IBB-BC algorithm like number clusters, iterations, dimension etc.
Initialize the population of IBB-BC algorithm in terms of clusters and select initial clusters centers in random order.
Compute the cost function using equation 1.1.
Rearrange data objects into different clusters according to minimum value of cost function.

Apply Big Bang Phase:
    Generate candidate solution (update centroid) using equations 4.4.
Apply Big Crunch Phase:
    Compute the fitness function and select the best fit as centre of mass (c.o.m) using equation 4.1

Compute heat transfer rate ($c_{h\_t\_r}$) using equations 4.6-4.8.
If ( $c_{h\_t\_r}(\mathrm{COM}_{t+1}) > c_{h\_t\_r}(\mathrm{COM}_t)$ )
    mass [i, j] = $\mathrm{COM}_{t+1}$
Else
    mass [i, j] = $(\mathrm{COM}_{t+1} + \mathrm{COM}_t)/2$
    End if
8:  Update COM
9:  Check the termination condition to stop the execution of algorithm, otherwise repeat the steps
    3-8.
```

4.4 Simulation Results

This section discusses the experimental results of IBB-BC algorithm. The details of simulation environment are taken same as mentioned in section 3.4. The parameters setting for BB-BC (Population = K × d, r = (0,1)) and IBB-BC algorithms (Population = K × d, a = 4). Whereas, parameters of other clustering algorithms are mentioned in section 3.4 (Table 3.1).

4.4.1 Results and Discussion

This subsection presents the comparative analysis of the performance of IBB-BC and standard clustering algorithms. The eight datasets are adopted for evaluating the experimental results of IBB-BC algorithm. The details of these datasets are given in subsection 3.4.1 (Table 3.2). The intra cluster distance and f-measure are considered for assessing the performance of IBB-BC. Several clustering algorithms like K-means, GA, PSO, ACO, CSO, BA [15,31-35,83-84,87-89], and BB-BC [20] are selected for comparing the results of IBB-BC clustering algorithms. For every dataset, the algorithms run thirty times individually, and each run consists of 200 iterations. The simulation results of IBB-BC and other algorithms are discussed in Table 4.1. It is revealed that IBB-BC algorithm achieves minimum intra-cluster distance for most of datasets. It is noticed that K-means obtains higher intra cluster distance for CMC, Wine, LR and Statlog datasets. Wheras, GA algorithms achieves higher intra cluster distance for Iris, Cancer, Glass and ISOLET datasets. Hence, it is concluded that both GA and K-means exhibit worst performance as compared to other algorithms. F-measure parameter is also adopted for evaluating the efficiency of IBB-BC algorithm. It is stated that IBB-BC algorithm achieve higher f-measure rate for all datasets except CMC dataset. For CMC dataset, K-means algorithm obtains higher f-measure rate. It is also observed that BA and IBB-BC exhibits similar performance for cancer dataset.

Table 4.1: Presents simulation results of proposed IBB-BC and other algorithms using intra cluster distance and f-measure

Sr No	Parameters	Algorithms							
		K-means	GA	PSO	ACO	CSO	BA	BB-BC	IBB-BC
Iris	Best case	97.52	113.98	97.05	97.21	96.98	96.84	102.01	94.32
	Avg. case	113.56	125.19	98.73	98.36	97.64	97.53	105.8	95.4
	Worst case	125.23	139.77	99.89	99.59	98.78	98.09	110.02	98.69
	F-measure	0.781	0.774	0.78	0.778	0.781	0.782	0.78	0.784
Cancer	Best case	2989.46	2999.32	2978.68	2983.49	2985.16	2972.36	2935.09	2824.1
	Avg. case	3248.25	3249.46	3116.64	3178.09	3124.15	3098.93	2960.03	2870.06
	Worst case	3566.94	3427.43	3358.43	3292.41	3443.56	3282.75	3162.71	3029.02
	F-measure	0.832	0.819	0.826	0.829	0.831	0.833	0.821	0.833
CMC	Best case	5834.21	5705.63	5792.48	5756.42	5712.78	5689.16	5702.41	5654.42
	Avg. case	5912.46	5756.59	5846.63	5831.25	5804.52	5778.14	5782	5726.94
	Worst case	5983.06	5812.64	5936.14	5929.36	5921.28	5914.25	5889.26	5875.52
	F-measure	0.337	0.324	0.333	0.332	0.334	0.336	0.286	0.232
Wine	Best case	16775.32	16490.41	16424.26	16456.81	16429.54	16372.02	16135.36	16077.48
	Avg. case	18059.91	16530.53	16491.52	16526.12	16486.21	16556.89	16714.4	16138.73
	Worst case	18783.23	16590.53	16589.13	16621.44	16595.45	16557.76	17246.48	16244.31
	F-measure	0.520	0.515	0.517	0.521	0.522	0.523	0.566	0.578
Glass	Best case	222.43	272.37	264.56	273.22	256.53	256.47	260.46	240.09
	Avg. case	246.51	282.32	278.71	281.46	264.44	269.61	264	242.52
	Worst case	258.38	291.77	283.52	286.08	282.27	278.24	269.28	249.7
	F-measure	0.426	0.333	0.412	0.402	0.416	0.431	0.462	0.471

Statlog	Best case	812090400	793000800	522200060	542208805	513208000	448208805	484638208	440009756
	Avg. case	812558906	793000994	522200928	542216190	513208164	450769448	485003646	440813477
	Worst case	813000125	793001140	522209000	542229001	513219100	452300087	485307375	441002759
	F-measure	0.262	0.314	0.322	0.329	0.312	0.316	0.32	0.399
LR	Best case	620900	610000	608000	608000	610000	612000	609600	604455.54
	Avg. case	624765.58	611731.68	608470.77	608495.87	611102.88	613775.68	610715.37	604516.7
	Worst case	626775.18	613600	609054.11	608786.61	612027.05	615000	611636.23	604593.5
	F-measure	0.461	0.488	0.412	0.427	0.416	0.439	0.418	0.496
ISOLET	Best case	446201.02	460280.78	450493.89	454350.19	447176.93	441222.8	450326.3	440906.04
	Avg. case	446502.65	460851.88	451718.88	455837.78	447733.55	442361.25	451084.54	441268.61
	Worst case	446905	462196.28	453961.88	458270.68	448585.87	443202.55	452369.77	441942.04
	F-measure	0.361	0.332	0.392	0.301	0.311	0.369	0.398	0.408

Figures 4.2-4.9 shows the convergence behavior of all clustering algorithm using all datasets. In these figures, horizontal axis denotes number of iterations, whereas, vertical axis denotes intra-cluster distance. It is seen that IBB-BC algorithm convergence on minimum values using all datasets. For few dataset, GA converges on larger values and K-means also converges on maximum values for few datasets. It is also noticed that IBB-BC algorithm takes less iteration for optimizing the solutions. It is revealed that IBB-BC algorithm succeeds better clustering results than other algorithm.

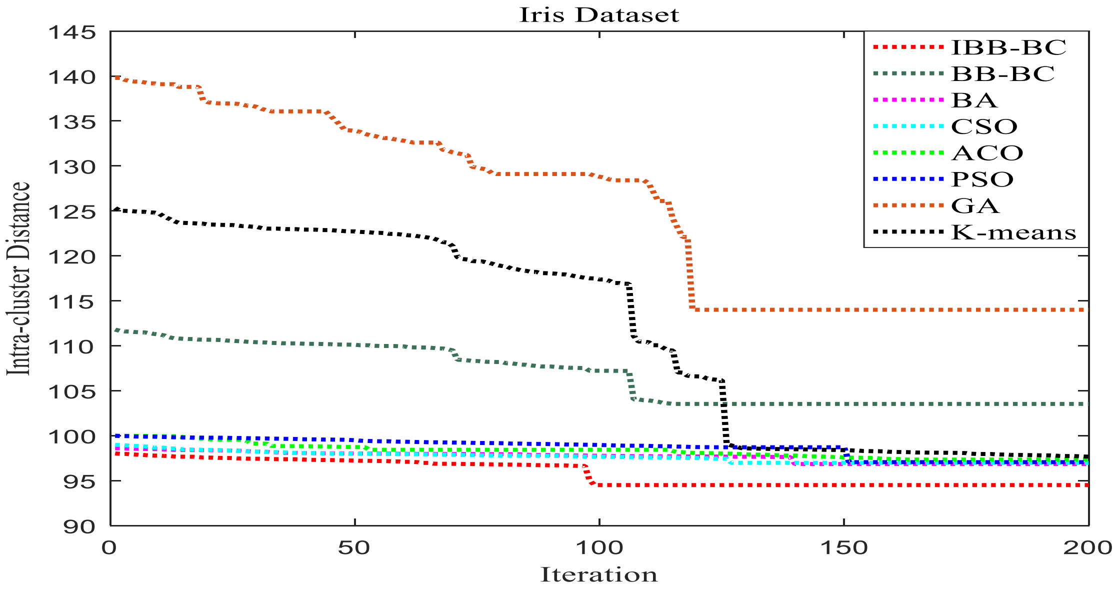

Figure 4.2: Convergence behavior using iris dataset

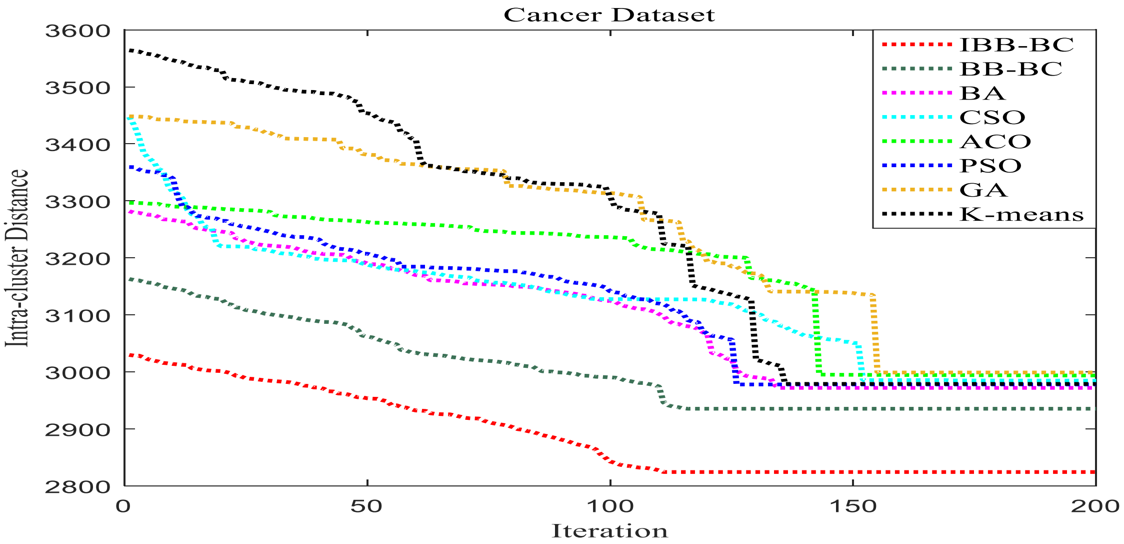

Figure 4.3: Convergence behavior using cancer dataset

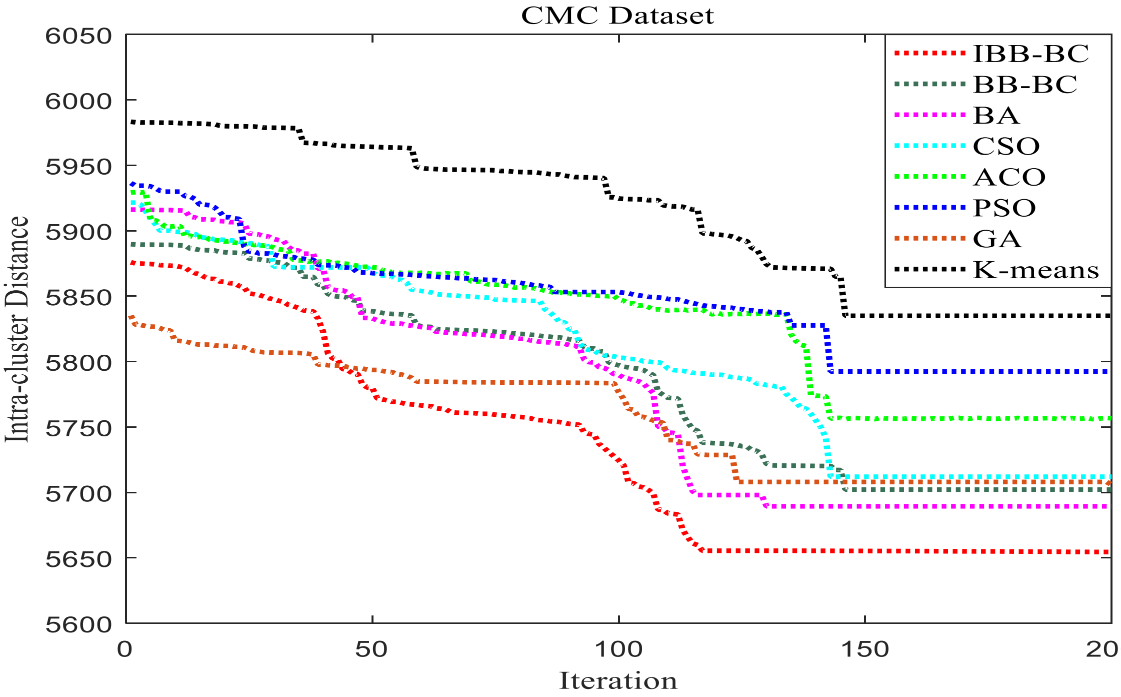

Figure 4.4: Convergence behavior using CMC dataset

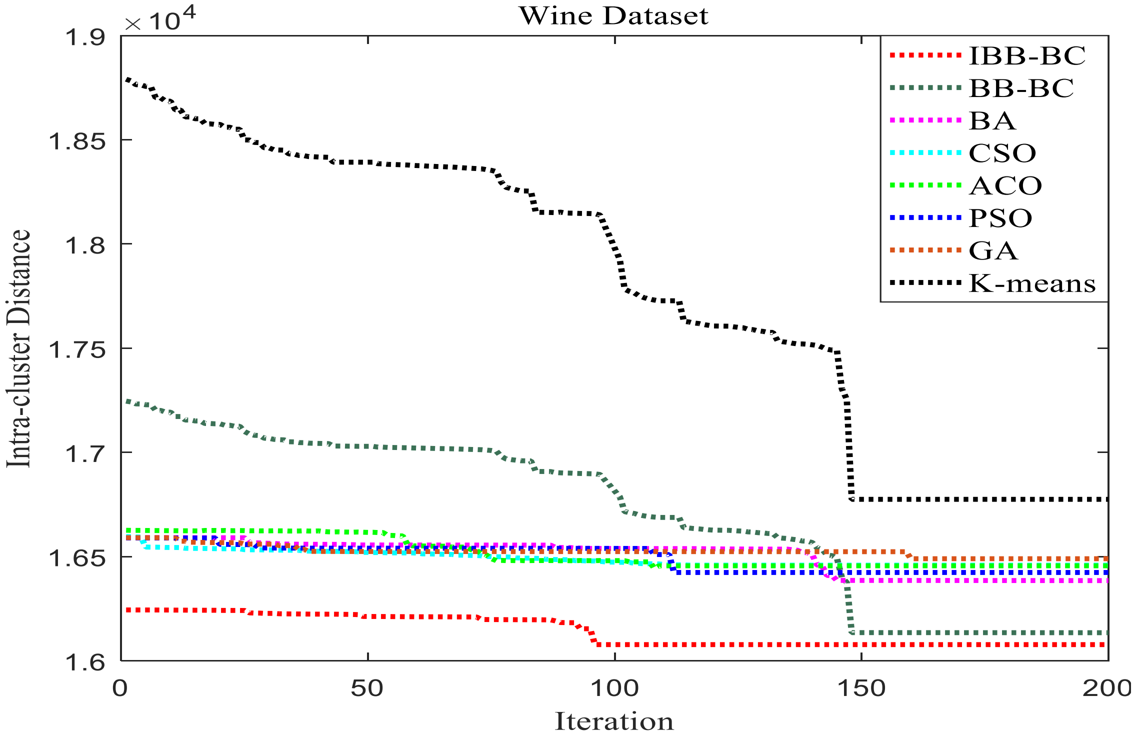

Figure 4.5: Convergence behavior using wine dataset

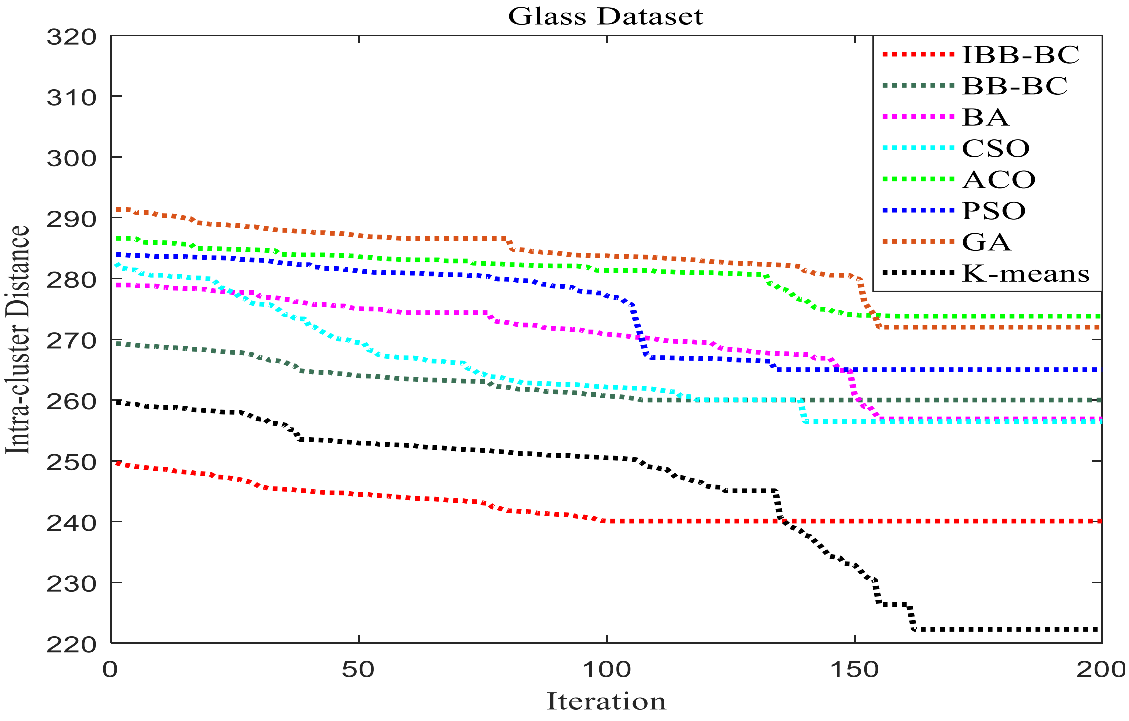

Figure 4.6: Convergence behavior using glass dataset

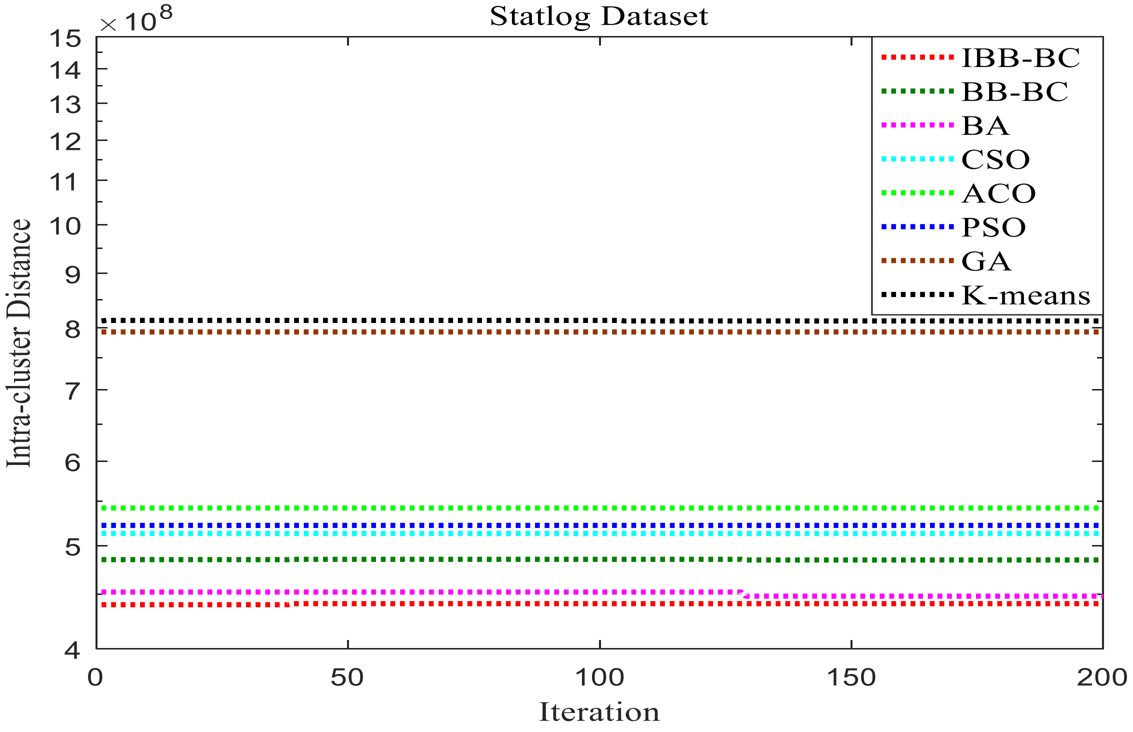

Figure 4.7: Convergence behavior using statlog dataset

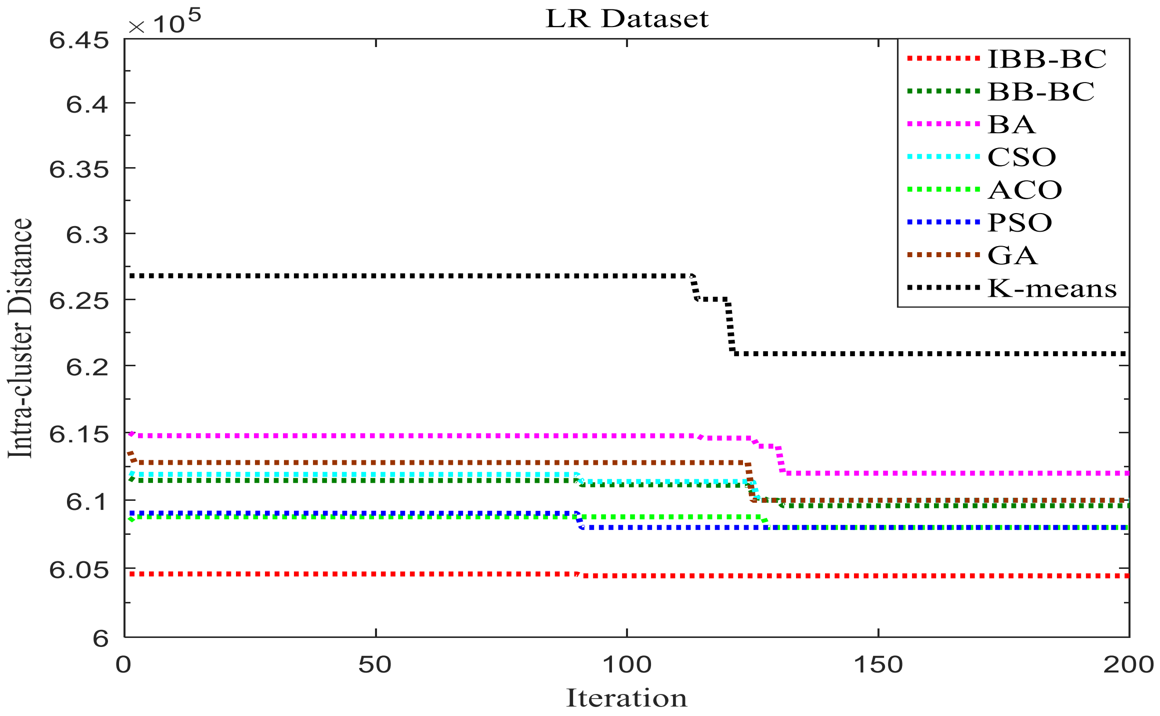

Figure 4.8: Convergence behavior using LR dataset

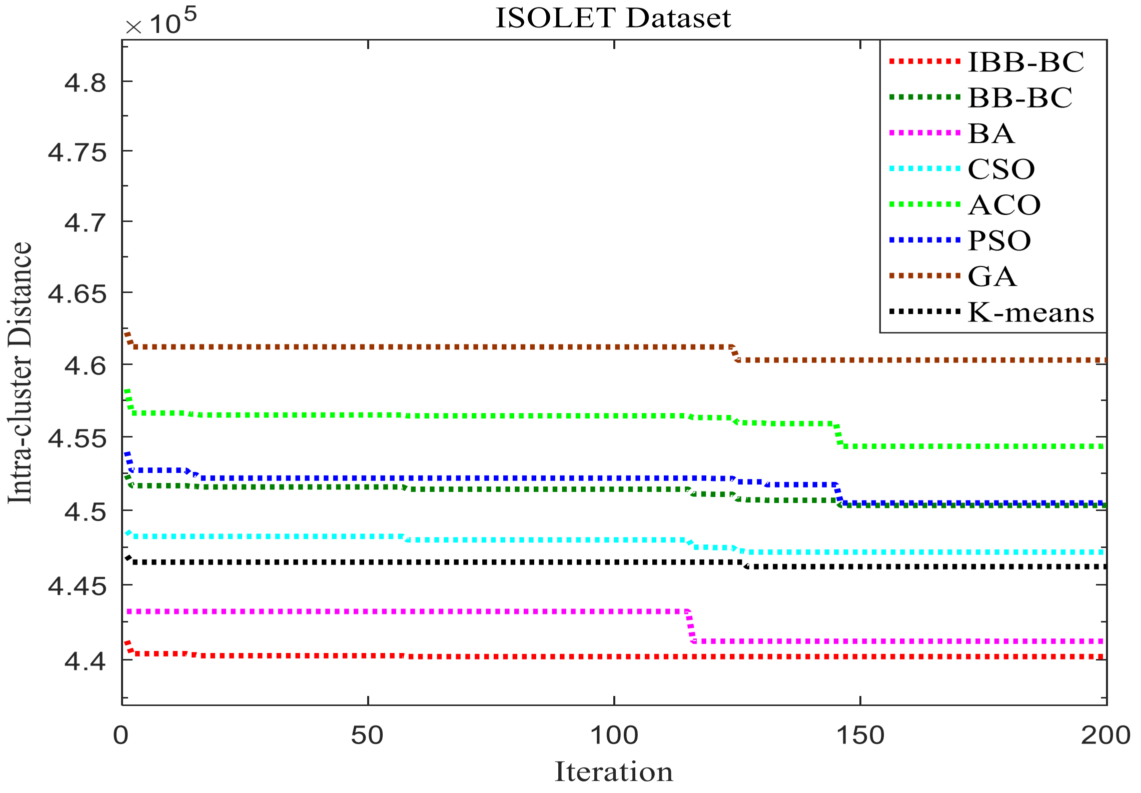

Figure 4.9: Convergence behavior using ISOLET dataset

Figures 4.10-4.17 shows the data objects belong to different clusters using clustering proposed IBB-BC algorithm. Figure 4.10 illustrates the clustering of data objects presented in iris dataset. The clusters in iris datasets are setosa, versicolour and virginica. The petal length, sepal width and sepal length attributes are selected to demonstrate the clustering of data objects. It is revealed that objects of setosa cluster are easily separable than versicolour and virginica clusters. But, rest of two are not linearly separable. It is stated that IBB-BC algorithm is capable to allocate the data objects into different clusters. Figure 4.11 shows the clustering of data objects of using cell size, cell shape and bare nuclei attributes of cancer dataset. This dataset consists of two clusters i.e. malignant and benign. The proposed algorithm successfully allocates the data objects to both of clusters. Figure 4.12 presents the clustering of data objects belong to CMC dataset using IBB-BC clustering algorithm. In CMC dataset, data objects are assigned to three clusters as 'Cluster No use1', 'Cluster Long Term2' and 'Cluster Short Term3'. It is revealed that data objects of CMC data are non-linear in nature, but IBB-BC algorithm significantly allocate the data objects to corresponding clusters as compared to existing algorithms. Figure 4.13 shows the clustering of data objects presented in wine dataset using alcohol, malic acid and ash attributes. This dataset contains three clusters such as wine type 1, wine type 2 and wine type 3. It is noticed that all clusters of wine datasets are non-linear in nature. Due to non-linearity, the clusters are not well separated. But, it is seen that proposed IBB-BC algorithm achieves better results for clustering the data objects. Figure 4.14 illustrates the clustering of data objects belong to glass dataset. This dataset consists of six cluster and further, the nature of data is non-linear. Figure 4.15 depicts the clustering of statlog data objects using IBB-BC algorithm. Data objects of statlog dataset are divided into seven groups, that are linearly inseparable. Furthermore, is seen that maximum data objects belong to 'Rad Flow' cluster. Figure 4.16 shows the clustering of LR dataset using IBB-BC algorithm. The LR dataset is divided into 26 clusters that affect the performance of the proposed algorithm. The IBB-BC provides average case results for LR dataset. Also, the clusters of LR dataset are linearly inseparable. Figures 4.17 display the clustering of the ISOLET dataset using IBB-BC algorithm. This figure depicts the clustering results using three real-valued attributes. The data objects of ISOLET datasets are divided into twenty-six clusters, that are linearly inseparable from others. The IBB-BC algorithm is capable to allocate the data objects into different clusters.

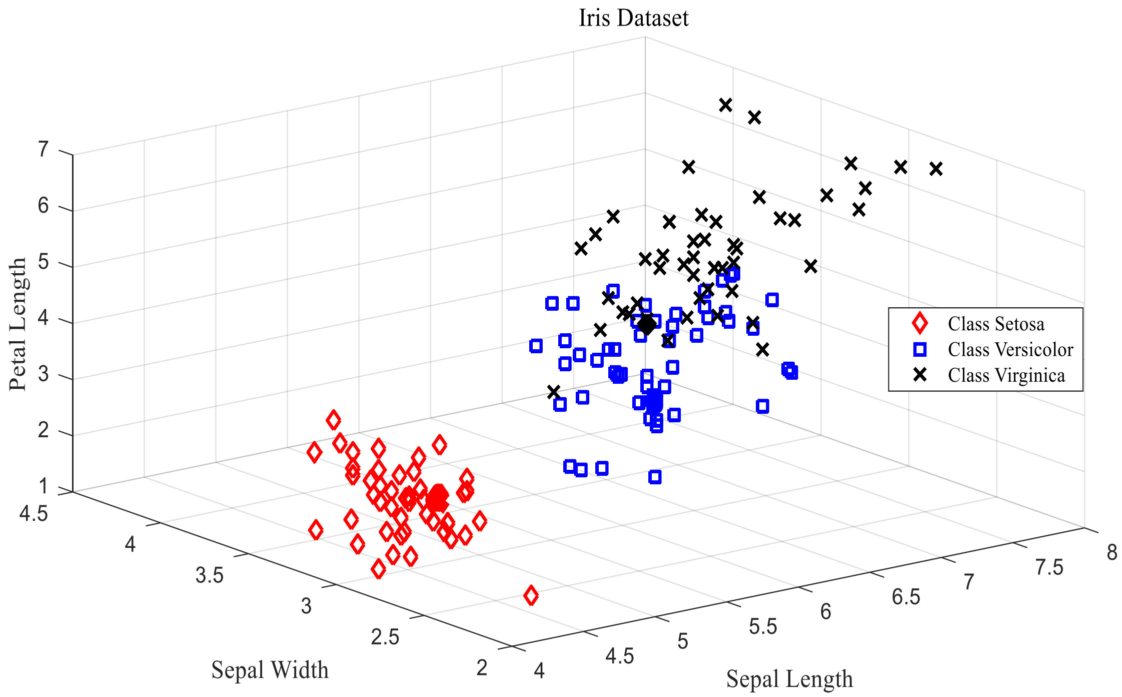

Figure 4.10: Clustering of iris data objects using IBB-BC algorithm

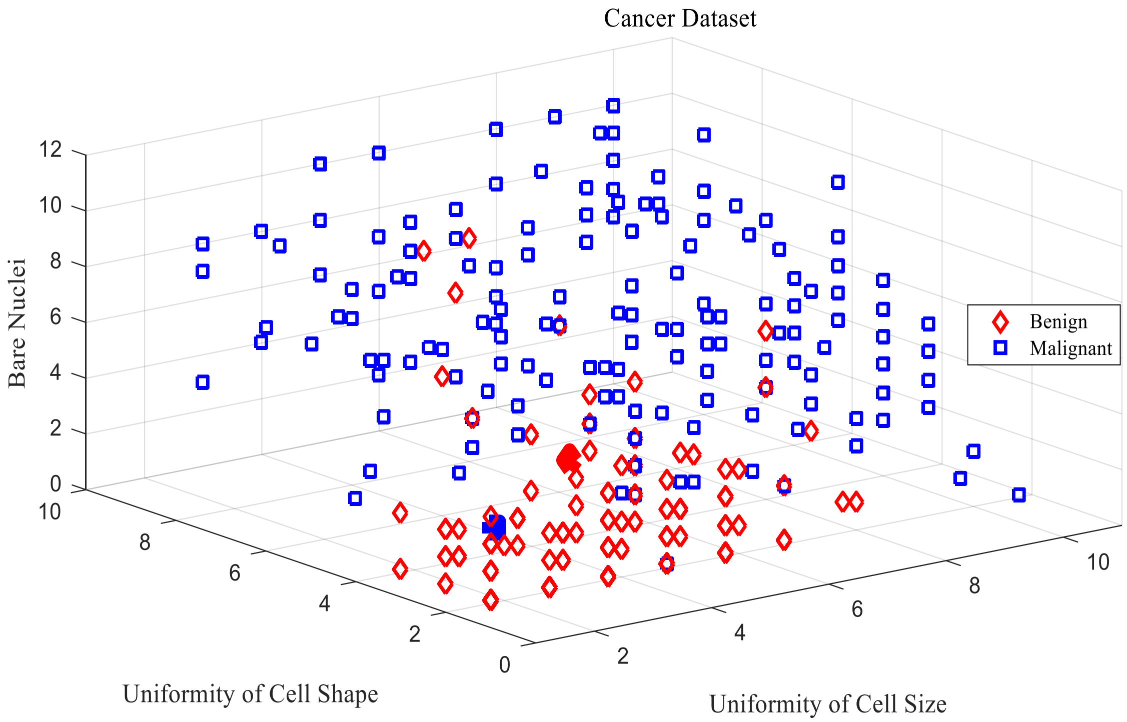

Figure 4.11: Clustering of cancer data objects using IBB-BC algorithm

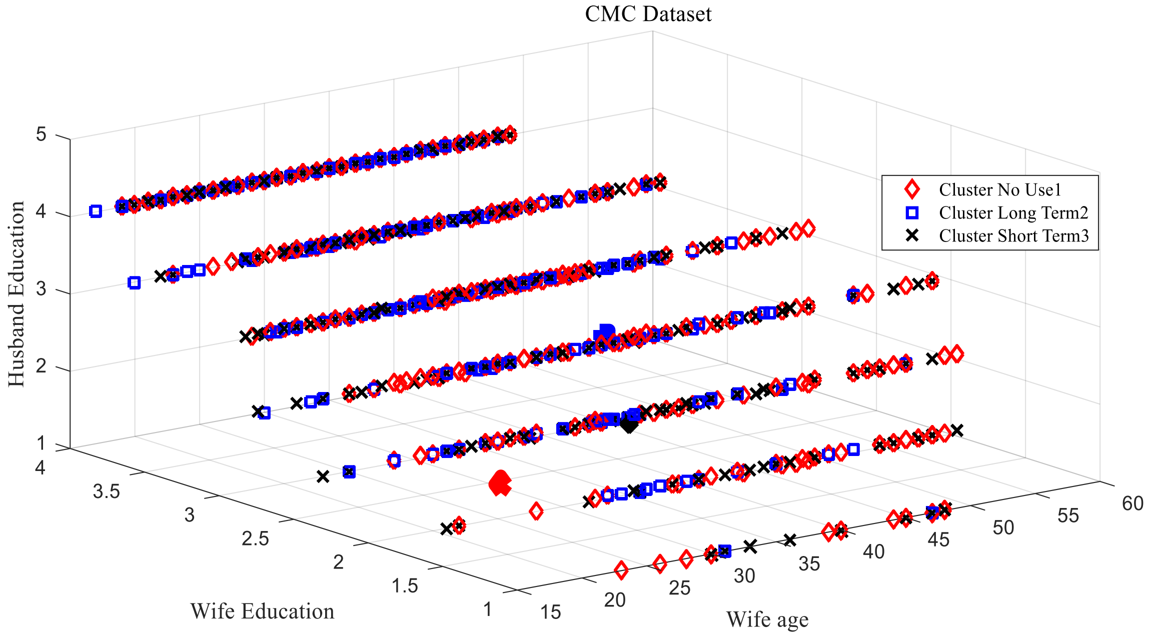

Figure 4.12: Clustering of CMC data objects using IBB-BC algorithm

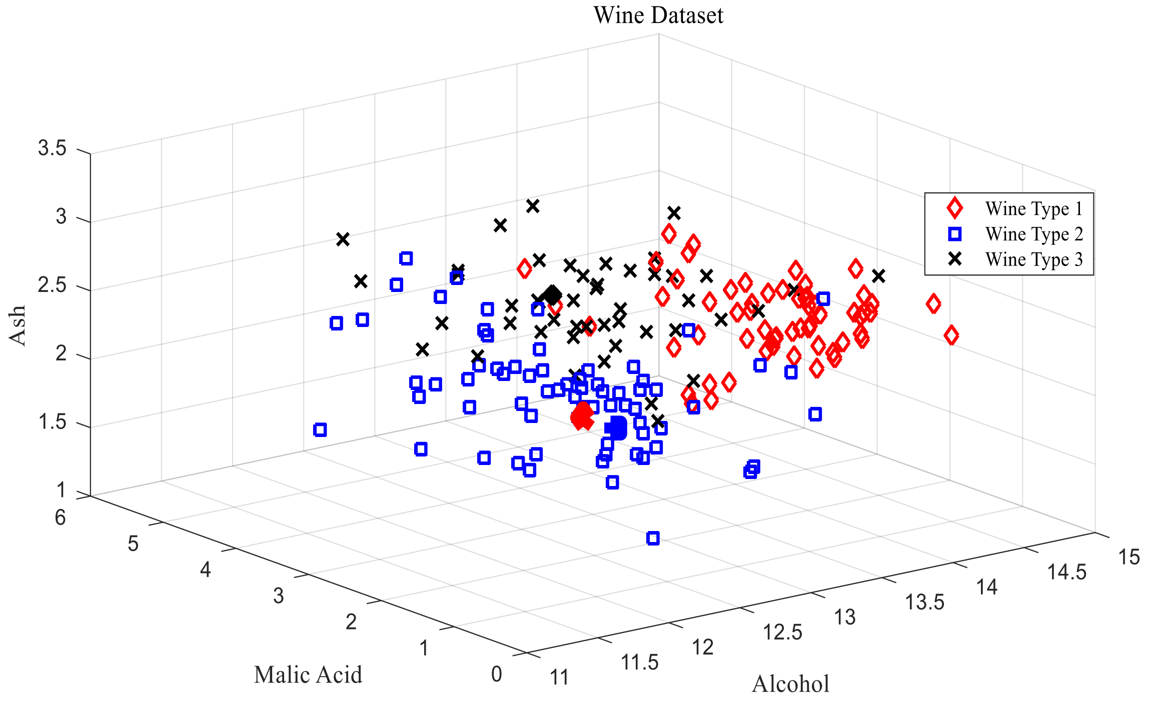

Figure 4.13: Clustering of wine data objects using IBB-BC algorithm

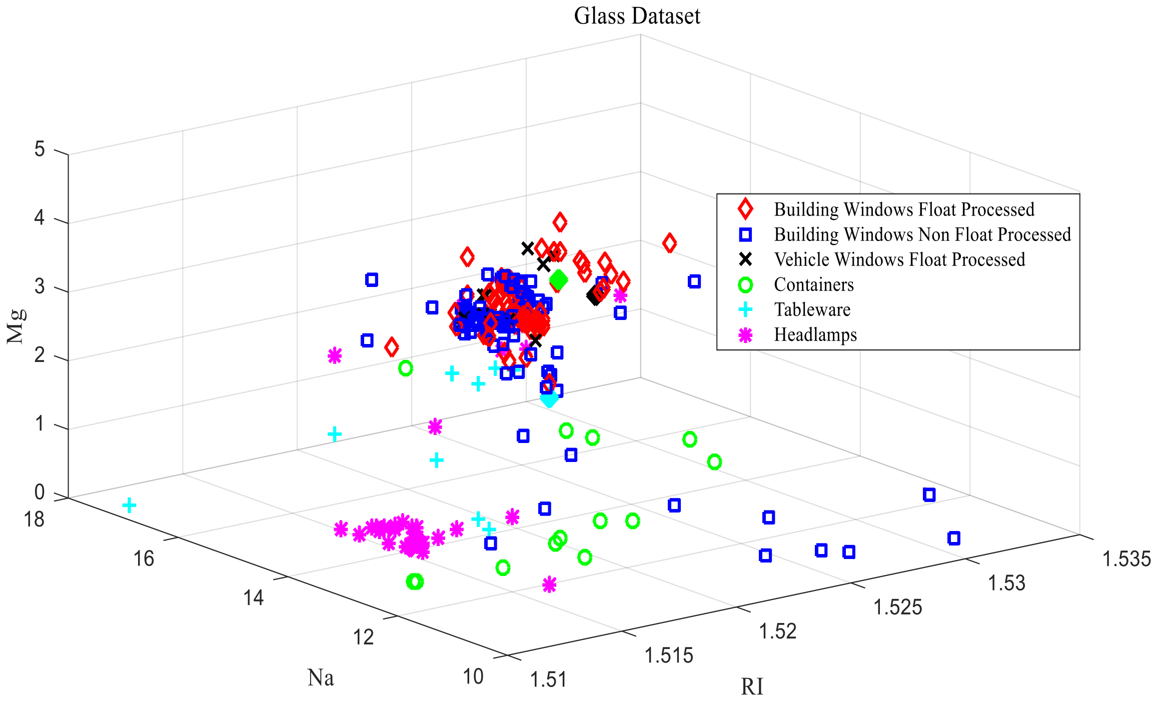

Figure 4.14: Clustering of glass data objects using proposed IBB-BC algorithm

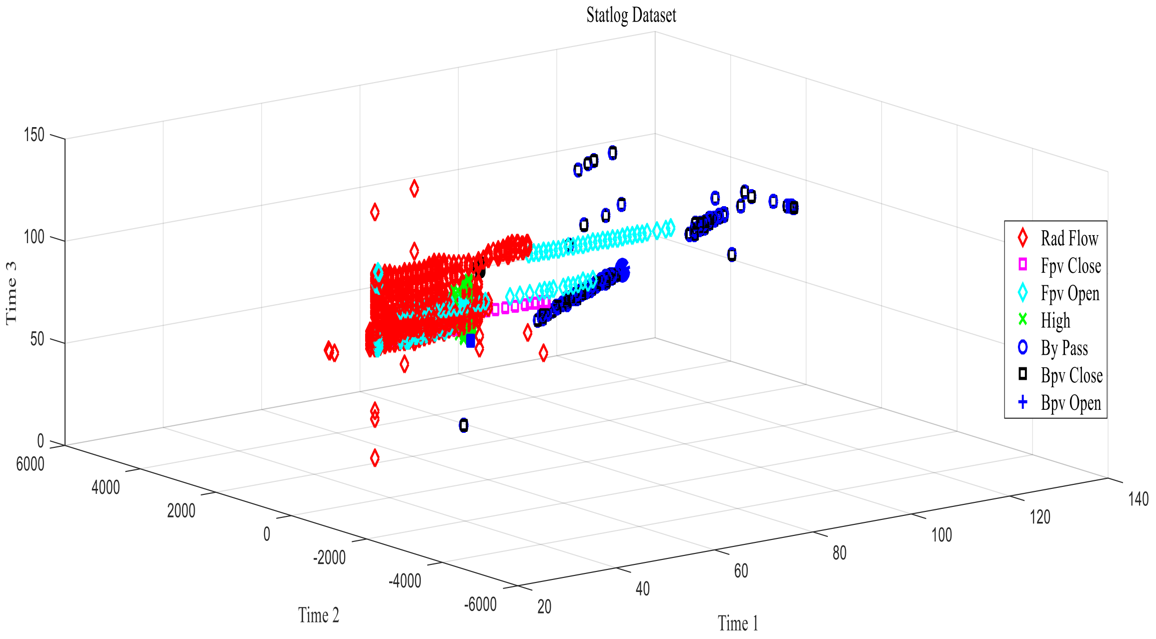

Figure 4.15: Clustering of statlog data objects using IBB-BC clustering algorithm

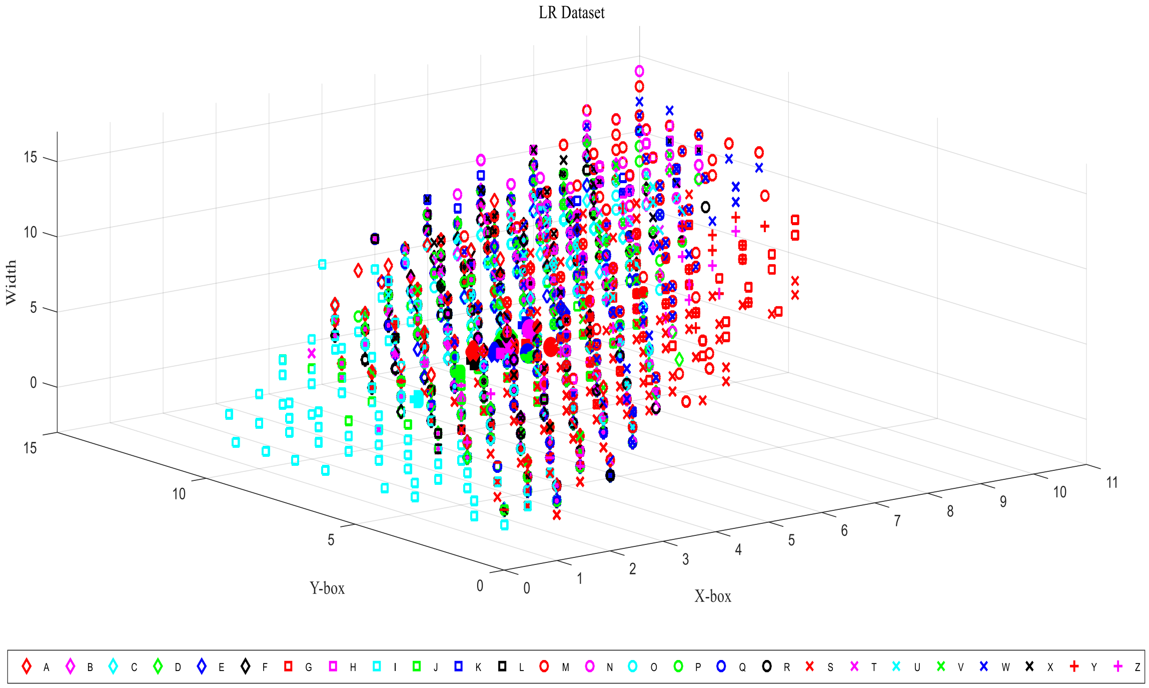

Figure 4.16: Clustering of LR data objects using IBB-BC clustering algorithm

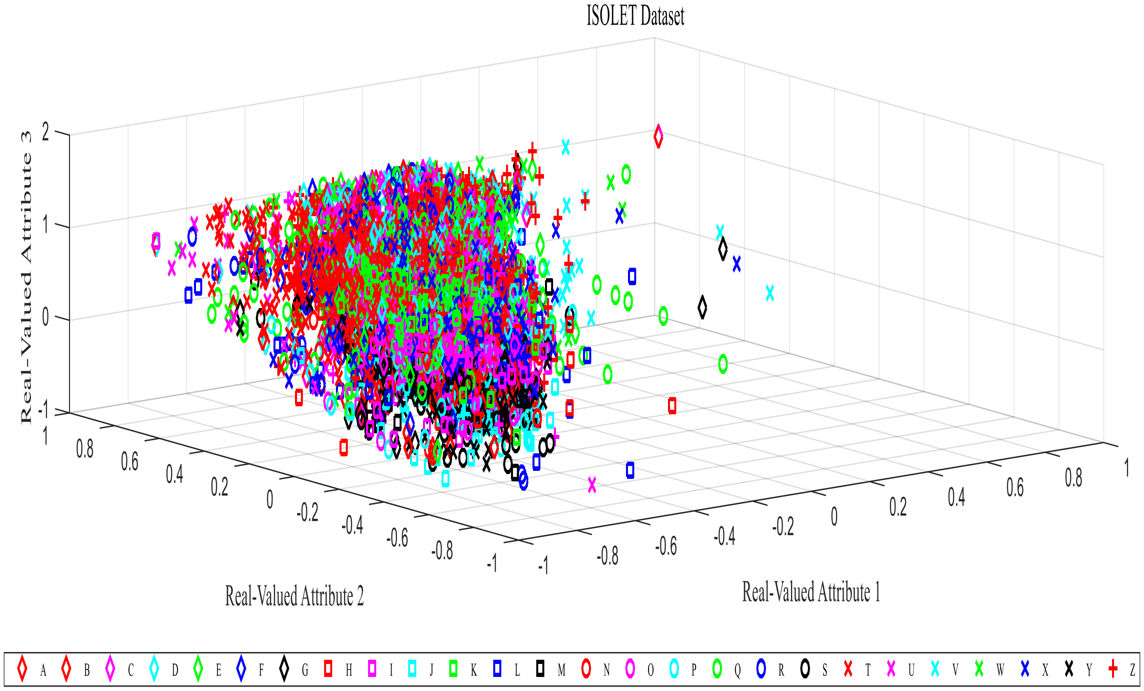

Figure 4.17: Clustering of ISOLET data objects using IBB-BC clustering algorithm

4.4.2 Statistical Test

The statistical results of Friedman test are discussed in this subsection. The reason behind inclusion of statistical test is explained in subsection 3.4.2. The results of the Friedman test are illustrated in Table 4.2 using intra-cluster distance parameter. For Friedman test, two hypotheses are designed; hypothesis (H_0) stands for performance are similar, while, hypothesis (H_1) stands for performance are dissimilar at certain significance level. The IBB-BC algorithm attains the first rank i.e. 1, among all other algorithms. The critical and p-values are 14.067 and 0.000169. The hypothesis is strongly rejected and the IBB-BC algorithm having dissimilar performance than other algorithms. It is stated that IBB-BC algorithm is significantly different in terms of simulation results.

Table 4.2: Statistics of Friedman using avg. intra-cluster distance parameter

Datasets	Clustering Algorithms							
	K-means	GA	PSO	ACO	CSO	BA	BB-BC	IBB-BC
Iris	7	8	8	4	3	2	6	1
Cancer	7	8	4	6	5	3	2	1
CMC	8	2	7	6	5	3	4	1
Wine	8	5	3	4	2	6	7	1
Glass	2	8	6	7	4	5	3	1
Statlog	8	7	5	6	4	2	3	1
LR	8	6	2	3	5	7	4	1
ISOLET	3	8	6	7	4	2	5	1
Sum	51	52	38	43	32	30	34	8
Rank	6.38	6.5	4.75	5.38	4	3.75	4.25	1
Number of observations: 64			Number of problems: 08			Number of algorithms: 8		
Sum of squares of rank sums: 11742.5			Correction factor: 1296			Friedman test statistic: 28.625		
Degree of freedom: 7			p-value: 0.000169			Critical value: 14.067		

Table 4.3 reported the statistics results of Friedman test using f-measure. The IBB-BC algorithm obtains first rank in most cases except cancer and CMC datasets. Moreover, GA algorithm exhibits poor performance using F-measure parameter. The critical value is 14.067, whereas, p-value is 0.006249. Hypothesis (H_0) is rejected and it showed that the significant difference occurs between the IBB-BC and other algorithms.

Table 4.3: Statistics of Friedman test using F-measure

Datasets	Clustering Algorithms							
	K-means	GA	PSO	ACO	CSO	BA	BB-BC	IBB-BC
Iris	3.5	8	5.5	7	3.5	2	5.5	1
Cancer	3	8	6	5	4	1.5	7	1.5
CMC	1	6	4	5	3	2	7	8
Wine	6	8	7	5	4	3	2	1
Glass	4	8	6	7	5	3	2	1
Statlog	8	6	3	2	7	5	4	1
LR	3	2	8	5	7	4	6	1
ISOLET	5	6	3	8	7	4	2	1
Sum	33.5	52	42.5	44	40.5	24.5	35.5	15.5
Rank	4.19	6.5	5.31	5.5	5.06	3.06	4.44	1.94

Number of observations: 64	Number of problems: 08	Number of algorithms: 8
Sum of squares of rank sums: 113.9.5	Correction factor: 1296	Friedman test statistic: 19.70
Degree of freedom: 7	p-value: 0.006249	Critical value: 14.067

4.5 Summary

This chapter presents the IBB-BC algorithm for cluster analysis. is proposed for solving partitional clustering problems. In proposed IBB-BC algorithm, two improvements are inculcated to address the issues related to traditional BB-BC algorithm. These issues are convergence rate and diversification. To address the same, chaotic maps and cellular automata-based concepts are integrated into BB-BC algorithm. Experimental results revealed that IBB-BC algorithm gives better quality clustering results than existing clustering algorithm., Statistical test also confirms the existence of IBB-BC in clustering field. Hence, it is said that IBB-BC performs well than other clustering algorithms in terms of experimental results as well statistical results.

CHAPTER 5
IMPROVED CSO FOR CLUSTERING

5.1 Introduction

In this chapter, CSO algorithm is considered for cluster analysis. CSO is a new algorithm inspired from cat behavior [35]. The main reason for the selection of CSO algorithm is its powerful exploration ability i.e. local search. In turn, CSO algorithm explores entire search space effectively to determine optimum solution. It is also reported that this algorithm suffers with local optima and slow convergence rate [72-78]. It is also noted that CSO algorithm having weak exploitation ability. This can also affect the performance of CSO algorithm specially to determine global optimum solution [83]. This chapter focuses on the shortcomings of CSO algorithm and explores the efficacy of CSO for cluster analysis.

5.2 CSO Algorithm

A new algorithm, called CSO is designed on the behavior of cats for addressing hard optimization problems [94]. The behavior of cats is studied in two modes-seeking and resting. The seeking mode illustrates the skills of cats to capture their target, whereas, resting mode depicts the behavior of cats in sleeping position. The computational steps of algorithm are listed as.

Step 1: Set algorithmic parameters of CSO algorithm such as population, SMP, CDC, MR, velocity, c, SRD etc. and select the initial population in random order.

Step 2: Evaluate the objective function. This function is formulated using as per problem definition.

Step 3: If flag is 1, start seeking mode and change the position of cats using SMP and SRD.

Step 4: Otherwise, start tracing mode and determine the updated position of cat using equations 5.1 and 5.2.

$$V_{i,new} = w * V_i + c * r * (X_{best} - X_i) \tag{5.1}$$

$$X_{i,new} = X_i + V_i \tag{5.2}$$

Step 5: Obtain the position of global best cat using fitness function.

Step 6: Check termination condition, if condition exist, obtain the optimal solution in terms of position of cats.

Step 7: Otherwise, repeat steps 2-5.

5.3 Improved CSO Algorithm for Clustering Problems

This section presents an improved version of CSO algorithm for clustering problems. Some improvements are integrated in CSO to overcome its deficiencies and robust for cluster analysis. These improvements are discussed in subsection 5.3.1.

5.3.1 Proposed Improvements

5.3.1.1 Modified Search Equations

The convergence rate issue of CSO is addressed through a modified search equation. It is reported that CSO having weak global search mechanism. So, to improve the exploitation ability, the position update equation of CSO is modified. The levy flight component and personal global best component is included in position update equation of CSO algorithm. Hence, the best solution is guided with help of position update equation. The new position update equation is described as below.

$$\mathrm{X}_{i,new} = \mathrm{X}_i + (\mathrm{P_g} * \mathrm{rand()}) \oplus \mathrm{Levy} + \mathrm{V_j^d} \tag{5.3}$$

Levy (U) $\sim |\mathrm{U}|^{-1-\alpha}$ where α denotes a value in between 0< α ≤ 2, p describes the step size $\mathrm{U} = \frac{S}{|\mathrm{T}|^{\frac{1}{\alpha}}}$; S and T are determined through equations 5.4-5.5.

$$\mathrm{S} \sim \mathrm{N}(0, \sigma_\mathrm{u}^2), \mathrm{T} \sim \mathrm{N}(0, \sigma_\mathrm{v}^2) \tag{5.4}$$

$$\sigma_\mathrm{u} = \left\{\frac{\Gamma(1+\alpha)\sin(\pi\alpha/2)}{\alpha\Gamma[(1+\alpha)/2]2^{(\alpha-1)/2}}\right\}^{1/\alpha}, \sigma_\mathrm{v} = 1 \tag{5.5}$$

Where, $\Gamma(1+\alpha) = \int_0^\infty \mathrm{t}^\alpha \mathrm{e}^{-\mathrm{t}}\,\mathrm{dt}$

To make a balanced co-ordination among local search and global search, a new velocity equation is developed along with the position update equation. Hence, equation 5.4 is used to compute the velocity of cats.

$$\mathrm{V}_{i,new} = \mathrm{V}_i + (\mathrm{P_{best,i}} - \mathrm{X}_i)\mathrm{rand()} \tag{5.6}$$

V_i, represents the new velocity values and V_i denotes the old velocity values, X_i denotes the current position and $\mathrm{P_{best,i}}$ denotes the personal best position.

5.3.1.2 Neighborhood based Search Strategy

This subsection presents a neighborhood strategy to deal the local optima situation. The aim of neighborhood strategy is to obtain better candidate solution such that algorithm will converge on global optimum solution. In this work, three neighborhood search strategies are developed to determine feasible solution. Out of three strategies, one corresponds to local search and rest of two are performed global search. The local neighborhood search is described through equation 5.7.

$$X_j^1 = r_1 \times X_j + r_2 \times \mathrm{best}_{\mathrm{cat}_j} + r_3 \times (X_{j1} - X_{j2}) \tag{5.7}$$

Xj1 and Xj2 denotes two cats selected into neighborhood radius of Xj ($j1 \neq j2 \neq j$) in random order, $\mathrm{best}_{\mathrm{cat}_i}$is the best position of cat, r_1, r_2 and r_3 are random value between (0,1) and sum is 1 i.e. $r_1 + r_2 + r_3 = 1$.

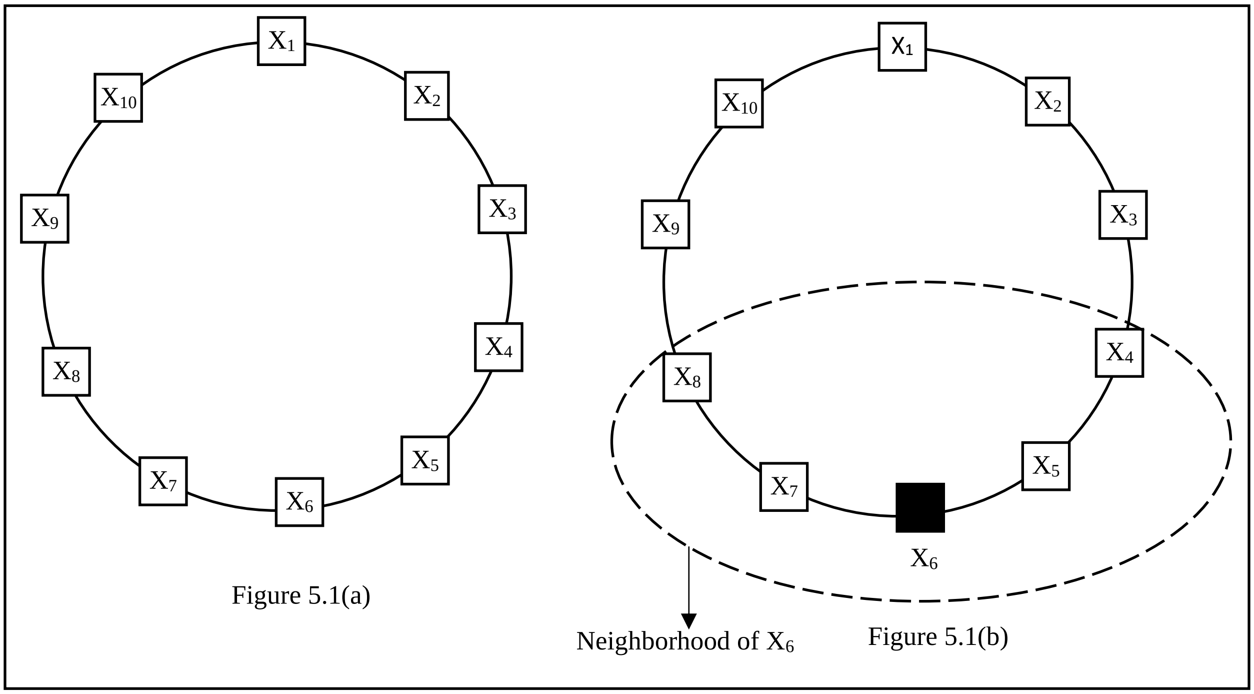

Figure 5.1(a & b): a. Represents the all neighborhood cats, b. illustrates neighborhood structure

Further, a global neighborhood search strategy is developed for improving the search ability of global optimum solution. It is similar to local neighborhood search and equation 5.8 is used to guide the global search.

$$X_j^2 = r_4 \times X_j + r_5 \times \mathrm{best}_{\mathrm{cat}_j} + r_6 \times (X_{j3} - X_{j4}) \tag{5.8}$$

X_{j3} and X_{j4} are randomly nominated cats (two cats) from the search space $(j3 \neq j4 \neq j)$, $\text{best}_{\text{cat}_j}$is the global best position of cat, r_4, r_5 and r_6 are random values and sum of these equal to 1 i.e. $r_4 + r_5 + r_6 = 1$.

It is observed that Cauchy distribution helps the algorithm to prevent the local optima situation [95]. Hence, Cauchy mutation operator is also considered in this work to guide the search in the direction of global optimum and also overcome local optima. It is described using equation 5.9 and corresponds to third neighborhood strategy.

$$X_j^3 = X_j + \text{Cauchy}( ) \tag{5.9}$$

Cauchy () denotes a random from Cauchy distribution and X_j^3 is a new candidate solution. Figure 5.2(a-c) illustrates the searching of feasible solutions in a given search space.

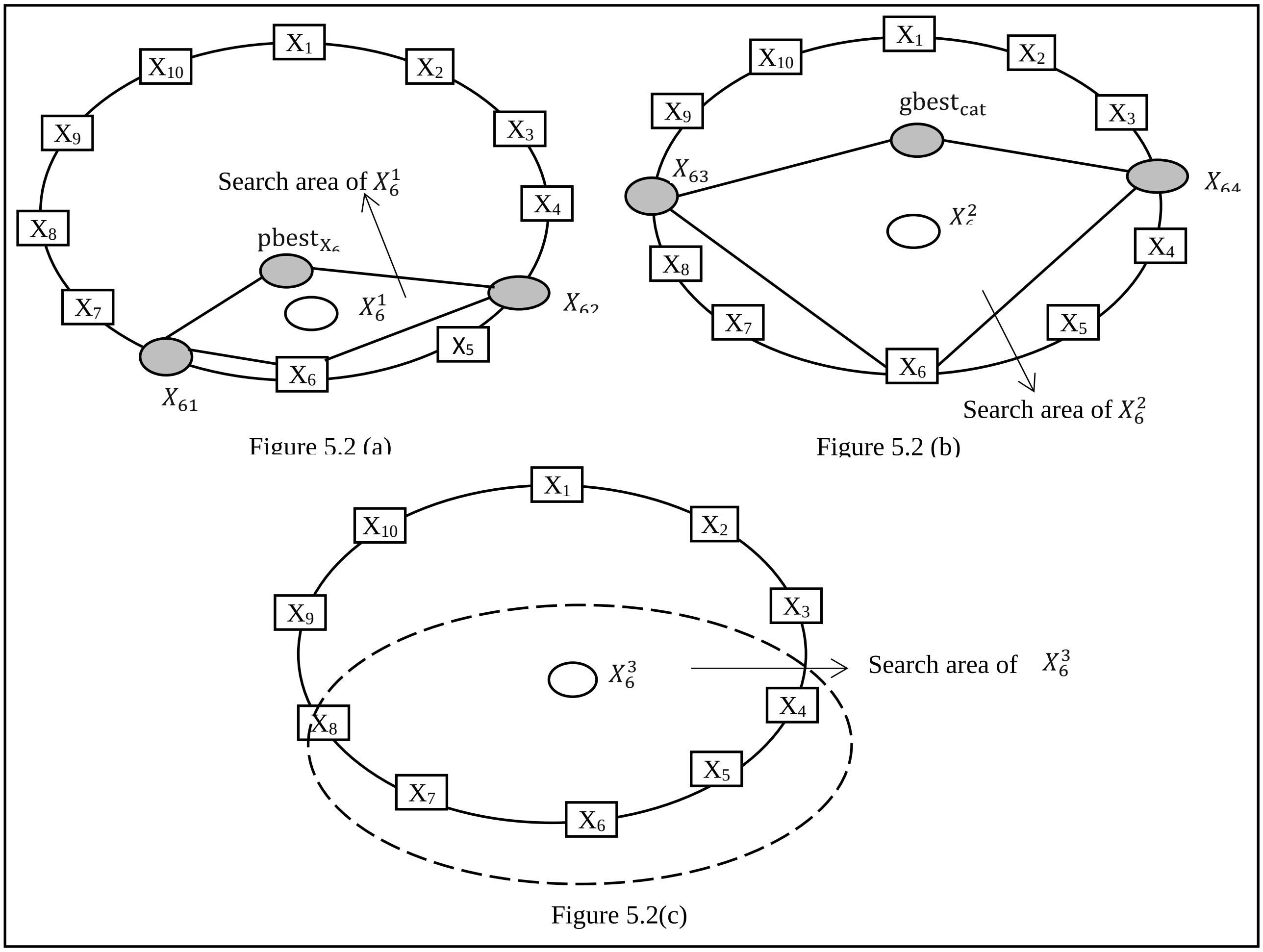

Figure 5.2 (a)

Figure 5.2 (b)

Figure 5.2(c)

Figure 5.2 (a-c): Shows the of local, global and good candidate solutions searching mechanism.

In the end of neighborhood identification, the solutions generated from local neighborhood search X_j^1, global neighborhood search X_j^2 and Cauchy operator X_j^3 are compared and best among these is selected as new candidate solution $X_{j.}$

Algorithm 5.1: Neighborhood Search Strategy

```
Begin
Step 1: Initialize the initial solution $X_i$, K- Neighborhood structures, function f to evaluate solutions.
Step 2: Generate the three trial candidate solutions $X_i^1$, $X_i^2$ and $X_i^3$ Using Eqs. 5.7-5.9.
Step 3: Compute the fitness of new generated candidate solutions $X_i^1$, $X_i^2$ and $X_i^3$.
Step 4: FE=FE+3 /* no. of Fitness Evaluation (FE).
Step 5: Pick the best solution among $X_{i,}$ $X_i^1$, $X_i^2$ and $X_i^3$according to fitness values as new $X_{i.}$
End
```

5.3.2 Improved CSO Clustering Algorithm

The ICSO clustering algorithm consists of three phases- Initialization, Evaluation and Assignment, and Update and Decision-making.

5.3.2.1 Initialization phase: In this phase, algorithmic parameters i.e. the number of cluster centers, iterations, neighborhood structure, β, α, C etc. are initialized. The dataset is loaded into memory, and the position and velocity of cats are initialized.

5.3.2.2 Evaluation and assignment phase: In this phase the seeking and tracking modes of the algorithm. The main work of this phase is to allocate data objects into different clusters. This allocation is done through objective function i.e. Euclidean distance. The fitness value of cats is also computed and best position are placed into memory pool.

5.3.2.3 Update and decision-making phase: In this phase, positions of cats are updated using search mechanism of ICSO algorithm. If, local optima situation occurs, then neighborhood search strategy (Algorithm 1) is implemented. The candidate solution is updated according to neighborhood strategies. Check the termination condition. If, the termination condition met, then stop the execution and obtains the optimal solution. The optimal solution is described in terms of optimal cluster centers. Otherwise, repeat the above process. Figure 5.3 demonstrates the flow chart of ICSO algorithm.

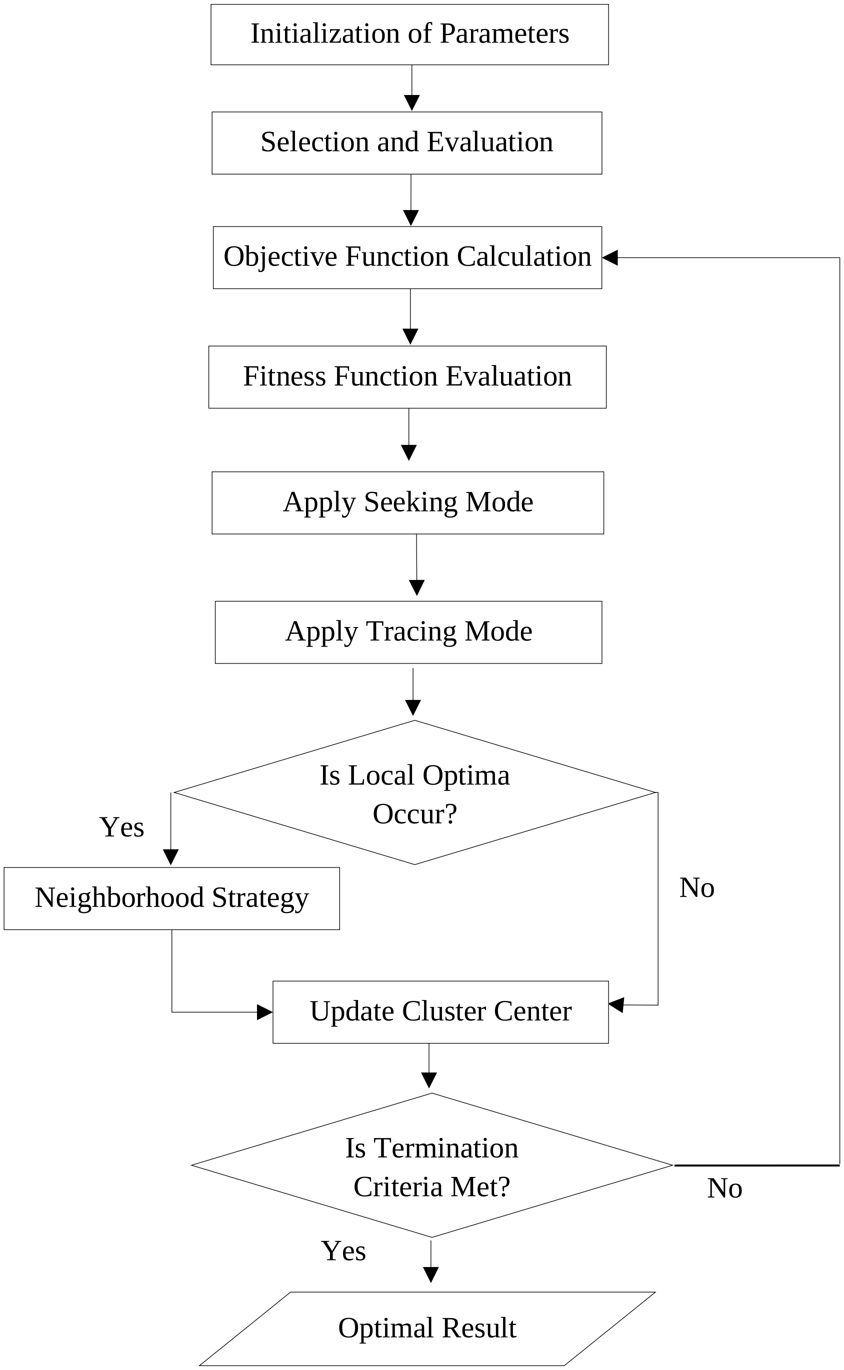

Figure 5.3: Flowchart of ICSO Algorithm

The computational steps of ICSO algorithm are listed as

Algorithm 5.2: Pseudo code of ICSO algorithm for clustering

```
Uniformly distribute cats in search space and set different algorithmic parameters viz population of
        cats, $\beta, \alpha, C$, SMP and SRD.
Determine the position and velocity of each cat.
Calculate the fitness of cats and determine best cat according to the fitness value.
While (i < maximum number of iterations /* i = iteration number), do
In accordance with Flag value, randomly distribute cats in search space.
If (Flag==1)   /* seeking mode starts
    • Construct the $j^{th}$ copy of each cat.
    • Using SRD, compute shifting bit value for each cat.
    • Compute new position of cats through addition/subtraction of SRD with previous position of
      cats.
    • Calculate the fitness function of new cats.
    • Compare the fitness and determine best position of cat
Else   /* Tracing mode starts
    • Compute velocity vector of each cat using Equation 5.6.
    • Compute the new position of each cat using Equation 5.3.
    • Calculate the fitness function.
    • Compare the fitness of each cat and determine the best cat
Updates the position of cats and determine the global best cat.
If $(rand(0,1) \leq Fit_i)$ , then
Apply the neighborhood search strategy.
Updates the position of cats and ($P_g$) global best position.
$i = i + 1$
Is termination condition met, Obtain the final solution; otherwise repeat above process
```

5.4 Experimental Results and Discussion

This section discusses the experimental results of ICSO algorithm. The simulation environment are taken same as mentioned in section 3.4. The parameters setting of ICSO algorithm are Population = K × d, SMP = 10, MR = 0.5, $\beta = 0.1 \sim 0.7$, $\alpha = 0.1 \sim 0.5$ and C = 2. Whereas, parameters of other clustering algorithms are mentioned in section 3.4 (Table 3.1).

5.4.1 Results and Discussion

This subsection presents the simulation results of ICSO and well-known clustering algorithms. The eight real life datasets are adopted for evaluating the experimental results of ICSO algorithm. The details of these datasets are given in subsection 3.4.1 (Table 3.2). The intra cluster distance and f-measure parameters are considered for assessing the performance of ICSO algorithm. For every dataset, the algorithms run thirty times individually, and each run consists of 200 iterations. The experimental results of ICSO and other algorithms are discussed in Table 5.1 [15,31-35,83-84,87-89]. The ICSO algorithm having minimum intra-cluster distance for most of datasets, although in case of CMC and glass datasets BA and ICSO has approximately equal values (average case). Furthermore, F-measure parameter is also adopted for evaluating the efficiency of ICSO algorithm. The results of f-measure parameter are presented in Table 5.1. Its revealed that ICSO algorithm achieves higher f-measure rate with most of datasets.

Table 5.1: Presents simulation results of ICSO and other algorithms using intra cluster distance and f-measure

Sr No	Parameters	Algorithms						
		K-means	GA	PSO	ACO	CSO	BA	ICSO
Iris	Best case	97.52	113.98	97.05	97.21	96.98	96.84	96.08
	Avg. case	113.56	125.19	98.73	98.36	97.64	97.53	97.18
	Worst case	125.23	139.77	99.89	99.59	98.78	98.09	97.83
	F-measure	0.781	0.774	0.78	0.778	0.781	0.782	0.784
Cancer	Best case	2989.46	2999.32	2978.68	2983.49	2985.16	2972.36	2972.36
	Avg. case	3248.25	3249.46	3116.64	3178.09	3124.15	3098.93	3045.93
	Worst case	3566.94	3427.43	3358.43	3292.41	3443.56	3282.75	3282.75
	F-measure	0.832	0.819	0.826	0.829	0.831	0.833	0.833
CMC	Best case	5834.21	5705.63	5792.48	5756.42	5712.78	5698.16	5689.16
	Avg. case	5912.46	5756.59	5846.63	5831.25	5804.52	5778.14	5778.04
	Worst case	5983.06	5812.64	5936.14	5929.36	5921.28	5921.01	5914.25
	F-measure	0.337	0.324	0.333	0.332	0.334	0.336	0.336
Wine	Best case	16775.32	16490.41	16424.26	16456.81	16429.54	16372.02	16357.89
	Avg. case	18059.91	16530.53	16491.52	16526.12	16486.21	16556.89	16372.02
	Worst case	18783.23	16590.53	16589.13	16621.44	16595.45	16557.76	16556.76
	F-measure	0.520	0.515	0.517	0.521	0.522	0.523	0.523
Glass	Best case	222.43	272.37	264.56	273.22	256.53	256.47	261.47
	Avg. case	246.51	282.32	278.71	281.46	264.44	269.61	269.61
	Worst case	258.38	291.77	283.52	286.08	282.27	278.24	274.24
	F-measure	0.426	0.333	0.412	0.402	0.416	0.431	0.424

Statlog	Best case	812090400	793000800	522200060	542208805	513208000	448208805	440009854
	Avg. case	812558906	793000994	522200928	542216190	513208164	450769448	440813574
	Worst case	813000125	793001140	522209000	542229001	513219100	452300087	441002855
	F-measure	0.262	0.314	0.322	0.328	0.312	0.316	0.329
LR	Best case	620900	610000	608000	608000	610000	612000	607752.66
	Avg. case	624765.58	611731.68	608470.77	608495.87	611102.88	613775.68	607757.11
	Worst case	626775.18	613600	609054.11	608786.61	612027.05	615000	607837.97
	F-measure	0.461	0.488	0.412	0.427	0.416	0.439	0.461
ISOLET	Best case	446201.02	460280.78	450493.89	454350.19	447176.93	441222.8	440105.55
	Avg. case	446502.65	460851.88	451718.88	455837.78	447733.55	442361.25	440136.35
	Worst case	446905	462196.28	453961.88	458270.68	448585.87	443202.55	441145.47
	F-measure	0.361	0.332	0.392	0.301	0.311	0.369	0.398

Figures 5.4-5.11 shows the convergence behavior of BA, CSO, ACO, PSO, GA, K-means and ICSO clustering algorithm. In these figures, horizontal axis denotes number of iterations, whereas, vertical axis denotes intra-cluster distance. The ICSO clustering algorithm converges on minimum values except glass dataset. Whereas, GA and K-means converge on larger values for most of datasets. It is noticed that ICSO algorithm converges faster than other algorithms. i.e. provide optimize results in lesser iterations.

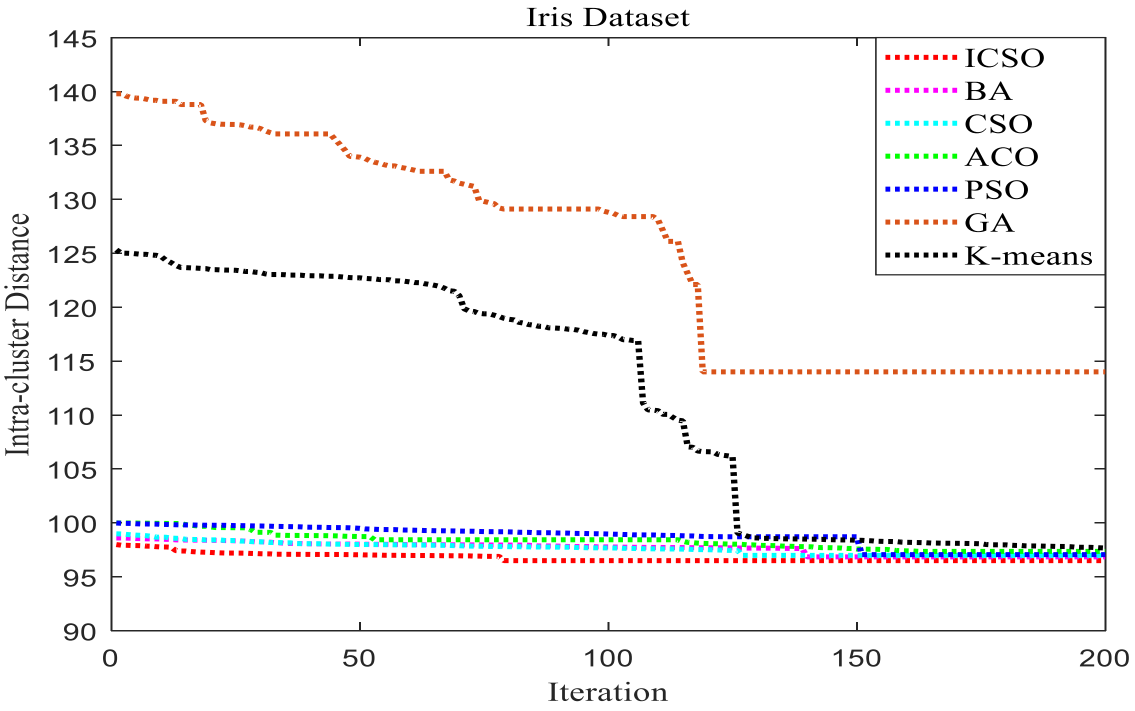

Figure 5.4: Convergence behavior using iris dataset

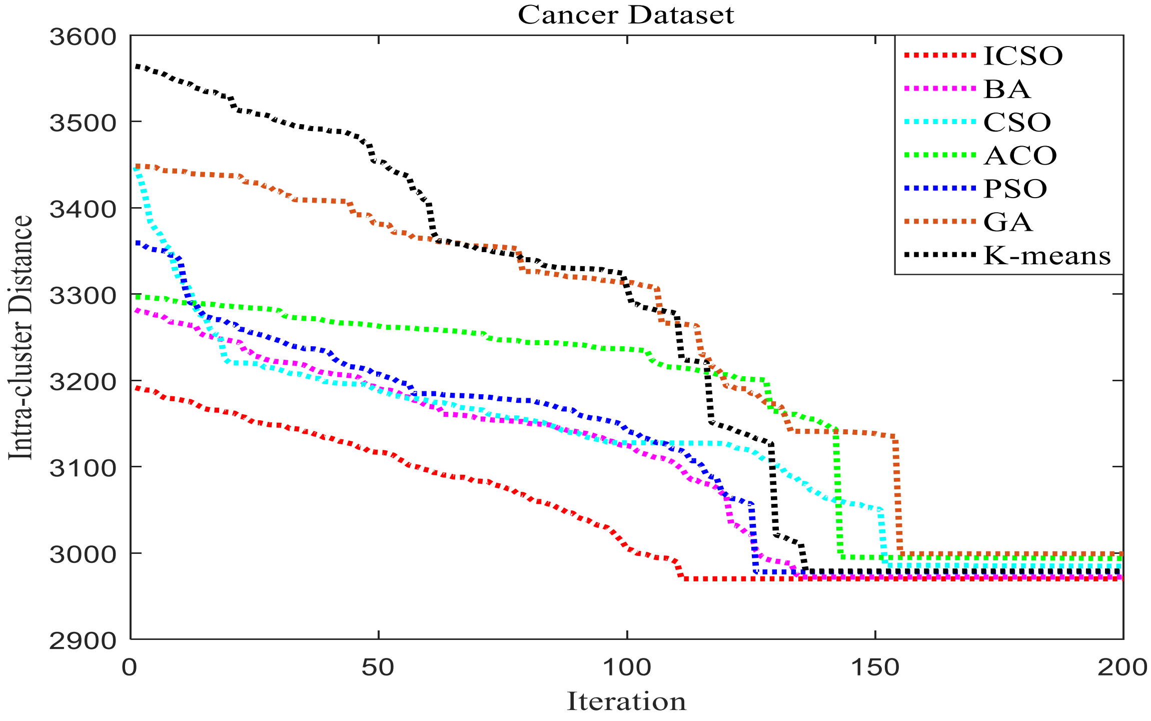

Figure 5.5: Convergence behavior using cancer dataset

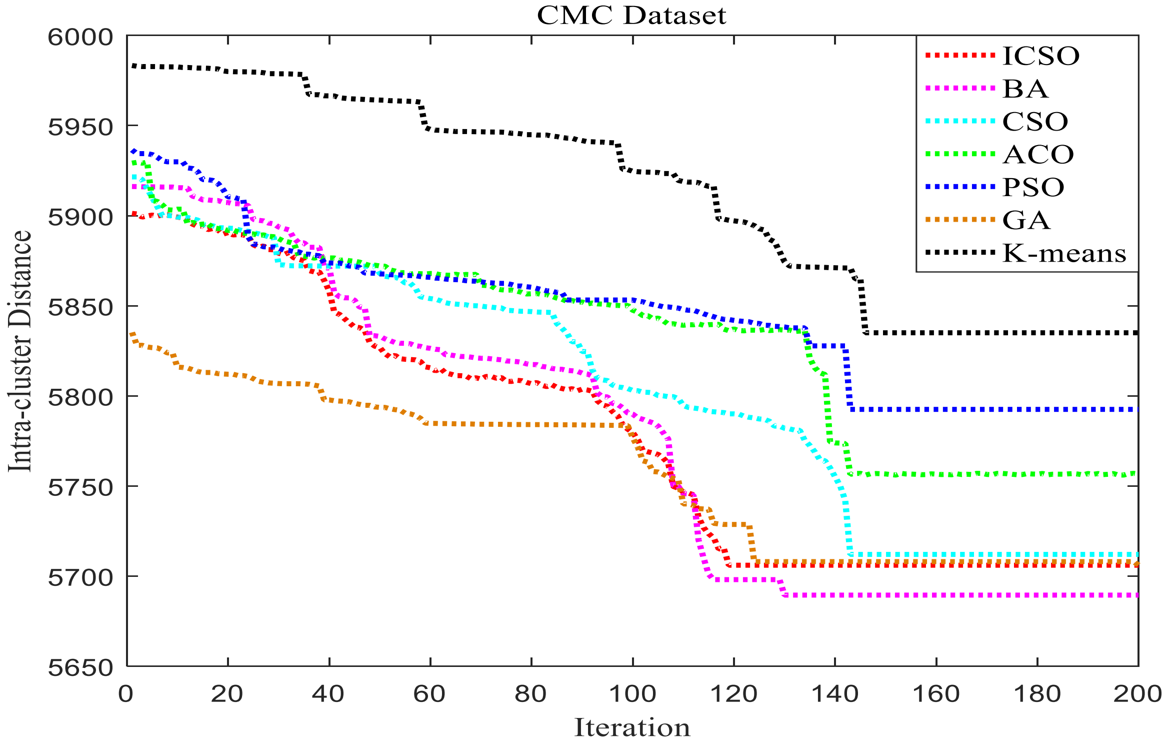

Figure 5.6: Convergence behavior using CMC dataset

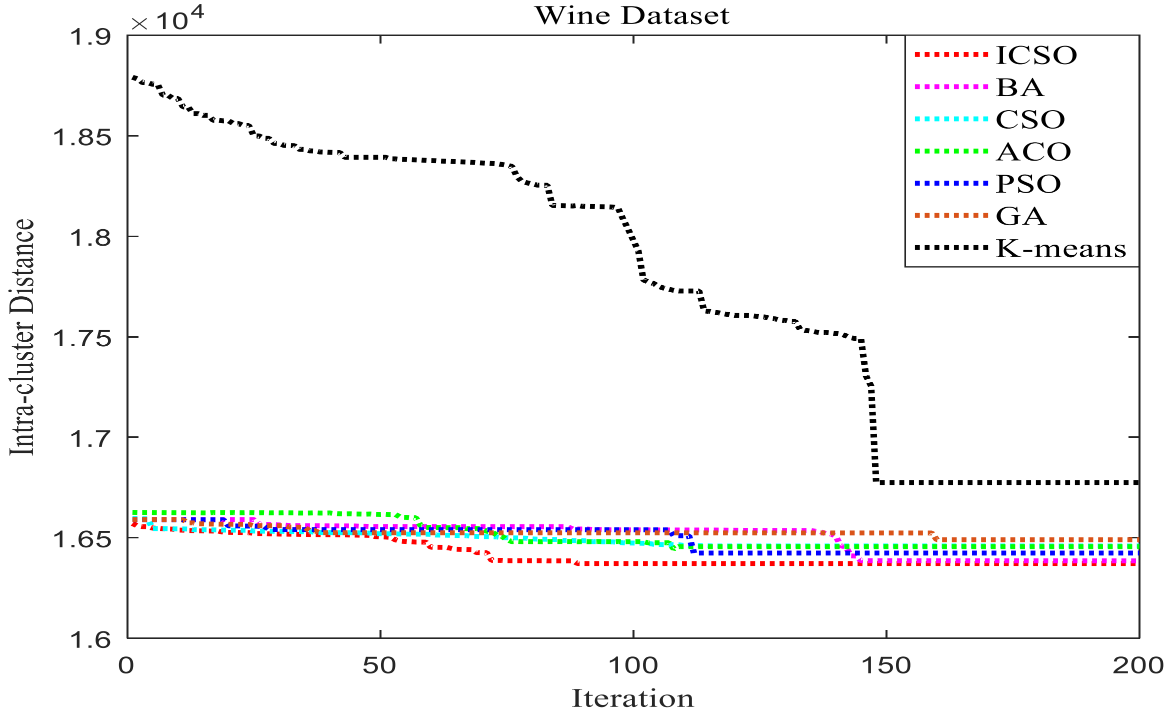

Figure 5.7: Convergence behavior using wine dataset

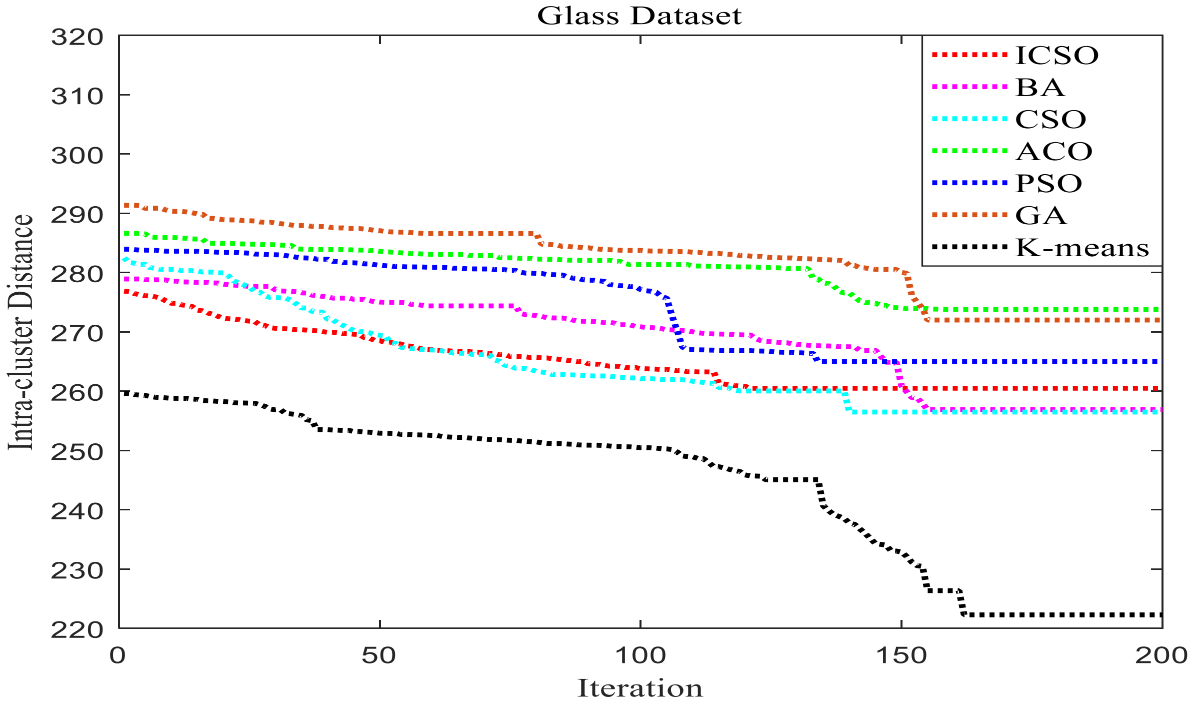

Figure 5.8: Convergence behavior using glass dataset

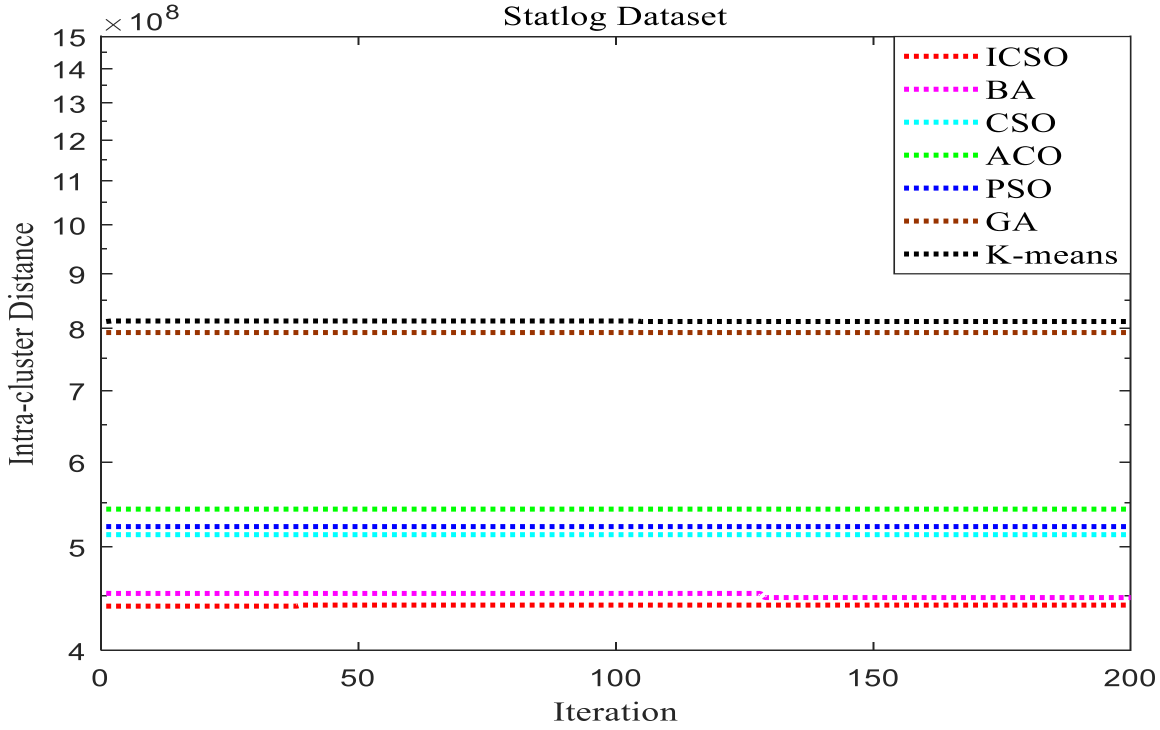

Figure 5.9: Convergence behavior using statlog dataset

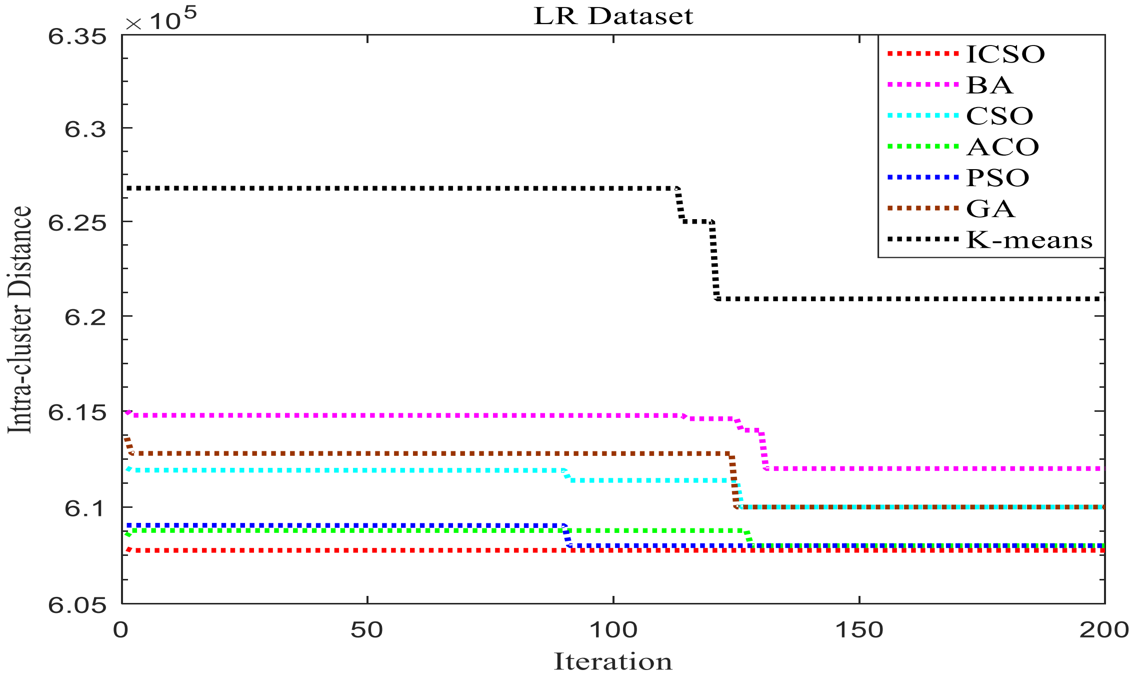

Figure 5.10: Convergence behavior using LR dataset

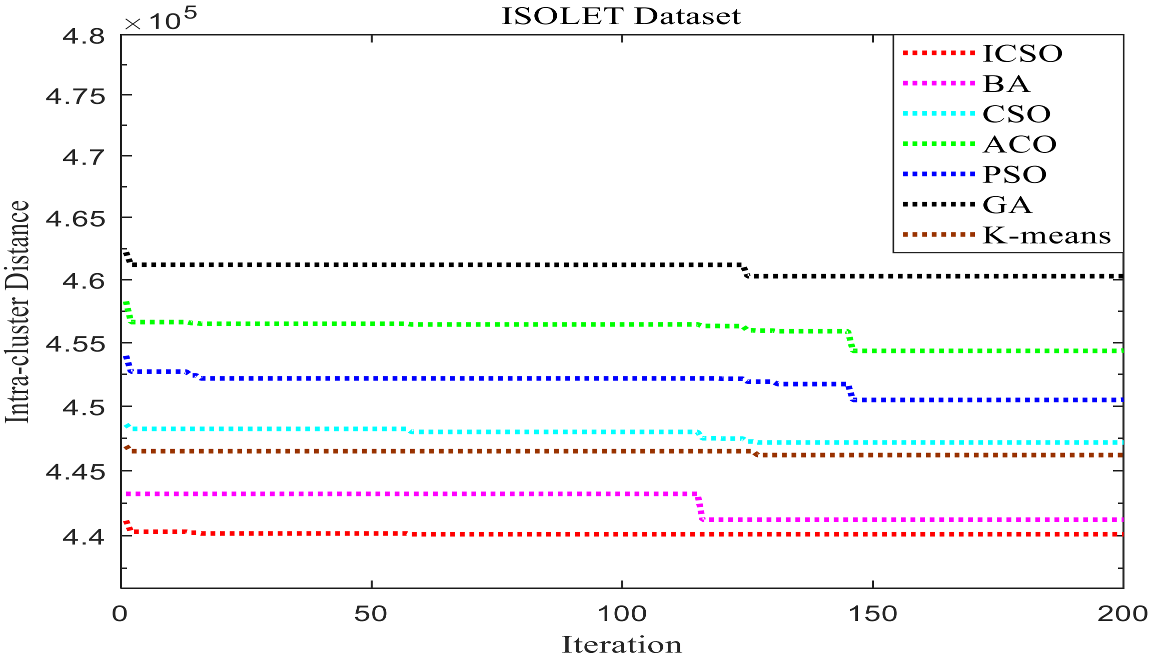

Figure 5.11: Convergence behavior using ISOLET dataset

Figures 5.12-5.19 shows the clustering of data objects into different clusters using ICSO clustering algorithm. Figure 5.12 illustrates the clustering of data objects presented in iris dataset. This dataset contains three cluster such as setosa, versicolour and virginica. The petal length, sepal width and sepal length attributes are selected to demonstrate the clustering of data objects. It is revealed that setosa cluster consists of linearly separable data objects. While, versicolour and virginica clusters contains non-linearly separable data objects. It is stated that ICSO algorithm is capable to allocate the data objects into clusters in efficient manner. Figure 5.13 illustrates the clustering of data objects using cell size, cell shape and bare nuclei attributes of cancer dataset. This dataset consists of two clusters i.e. malignant and benign. The ICSO algorithm allocates the data objects into different clusters successfully. Figures 5.14 presents the clustering of data objects belong to CMC dataset using ICSO clustering algorithm. In CMC dataset, data objects are assigned to three clusters as ‘Cluster No use1’, ‘Cluster Long Term2’ and ‘Cluster Short Term3’. It is revealed that data objects of CMC data are non-linear in nature, but ICSO algorithm significantly allocate the data objects to corresponding clusters as compared to existing algorithms. Figure 5.15 depicts the clustering of data objects presented in wine dataset using alcohol, malic acid and ash attributes. This dataset contains three clusters such as wine type 1, wine type 2 and wine type 3. It is noticed that all clusters of wine datasets are non-linear in nature. Due to non-linearity, the clusters are not well separated. The ICSO algorithm effectively assigns the data objects to clusters. Figure 5.16 illustrates the clustering of data objects belong to glass dataset. This dataset consists of six cluster and further, the nature of data is non-linear. The ICSO algorithm is capable to allocate the data objects into different clusters. Figure 5.17 shows the clustering of the statlog(shuttle) dataset using ICSO algorithm. Data objects of statlog dataset are divided into seven clusters (Rad Flow, Fpv Close, Fpv Open, High, Bypass, Bpv Close and Bpv Open). It is observed that maximum of data objects belongs to cluster 'Rad Flow' and are linearly inseparable from other clusters. Figure 5.18 shows the clustering of LR dataset using ICSO algorithm. The LR dataset is divided into 26 clusters (A-Z), that are linearly inseparable. Figures 5.19 displays the clustering of the ISOLET dataset using ICSO algorithm. The data objects of ISOLET datasets are divided into twenty-six clusters, that are inseparable from others. It is noticed that the performance of the proposed algorithm is affected due to large number of clusters.

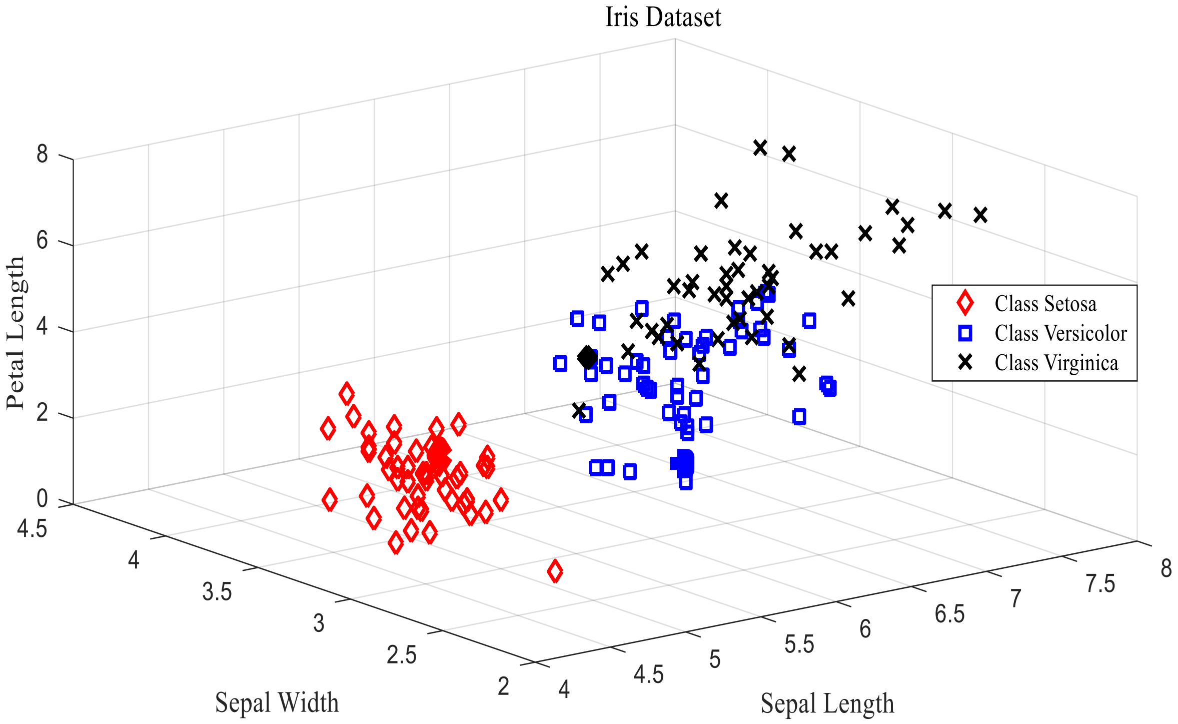

Figure 5.12: Clustering of iris data objects using proposed ICSO algorithm

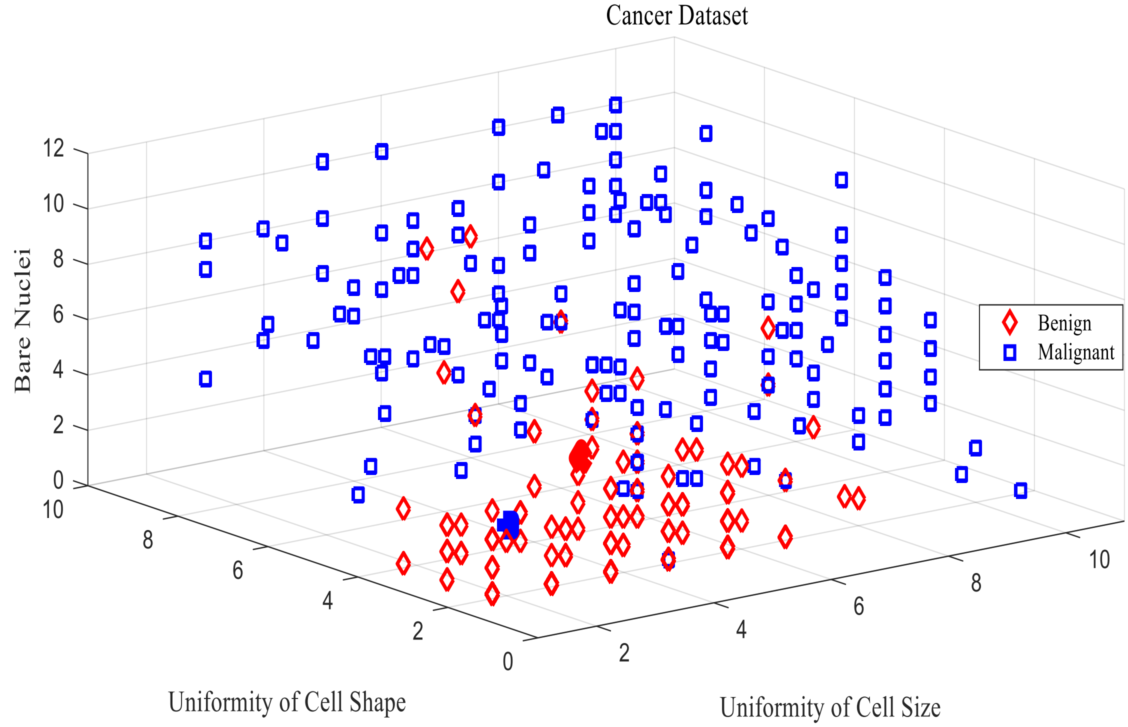

Figure 5.13: Clustering of cancer data objects using proposed ICSO algorithm

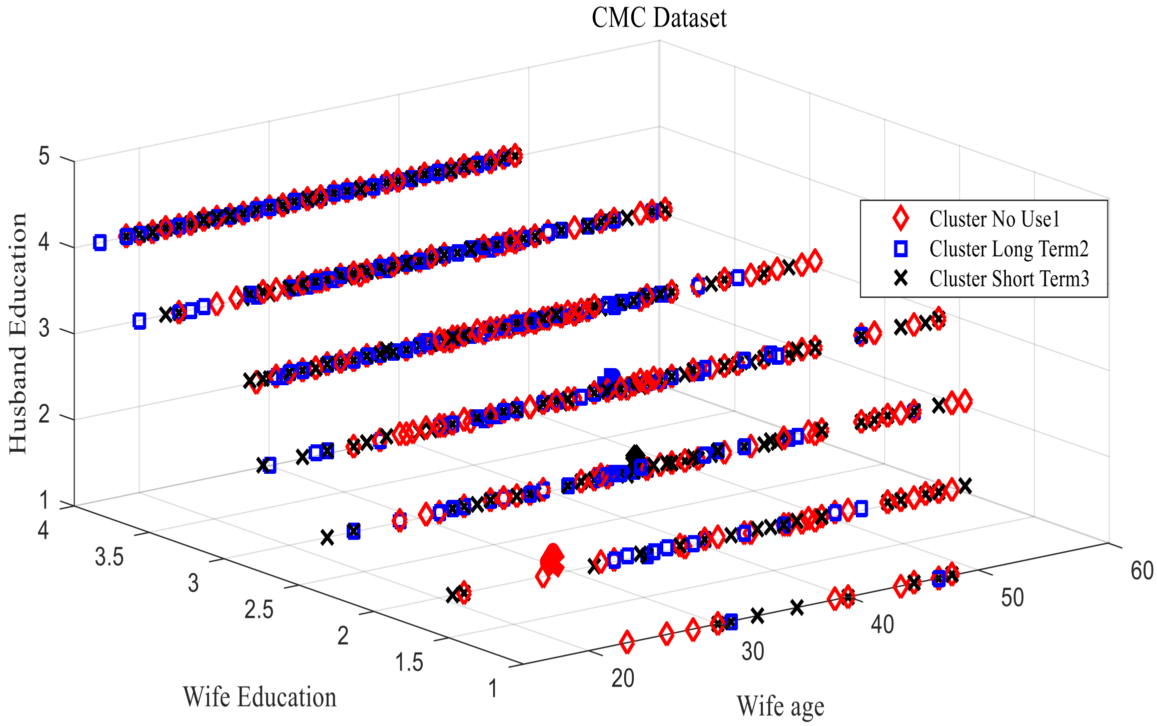

Figure 5.14: Clustering of CMC data objects using proposed ICSO algorithm

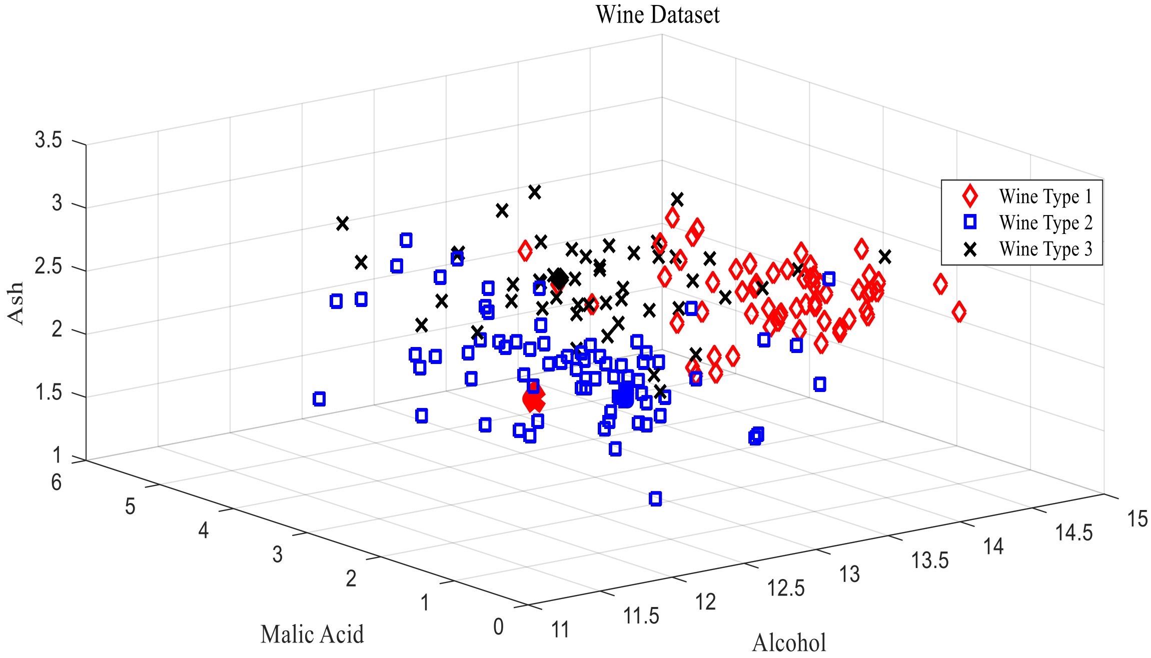

Figure 5.15: Clustering of wine data objects using proposed ICSO algorithm

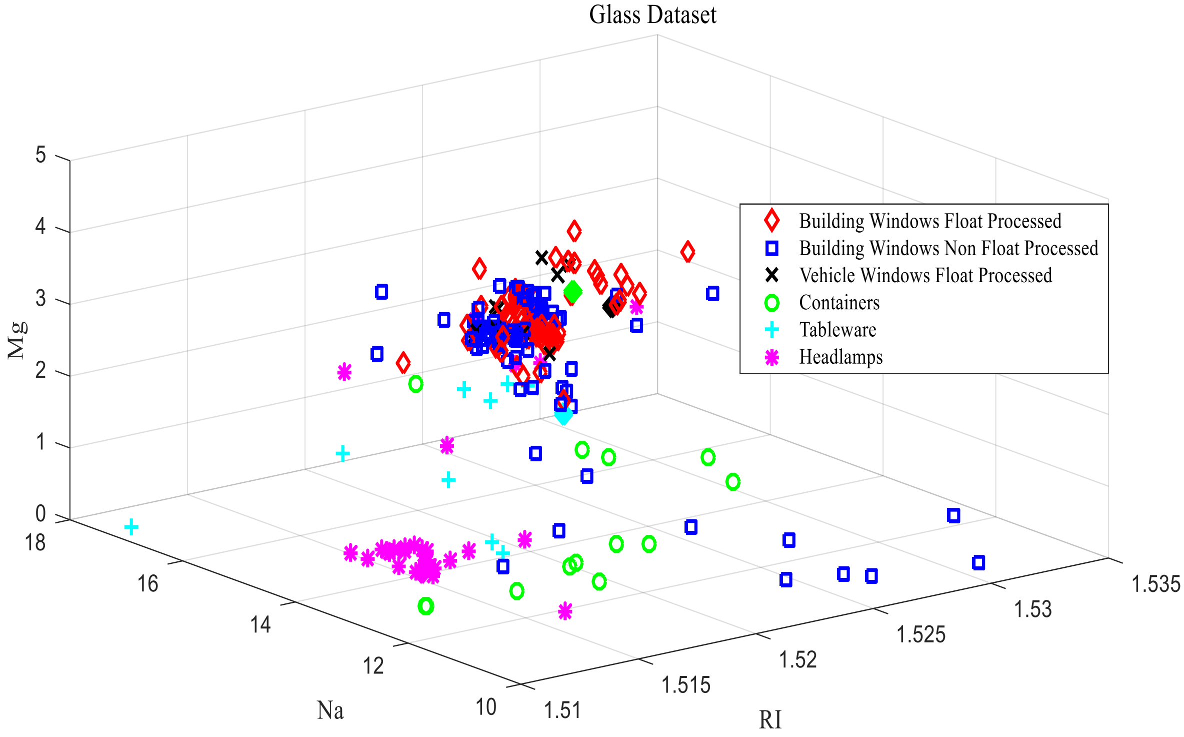

Figure 5.16: Clustering of glass data objects using proposed ICSO algorithm

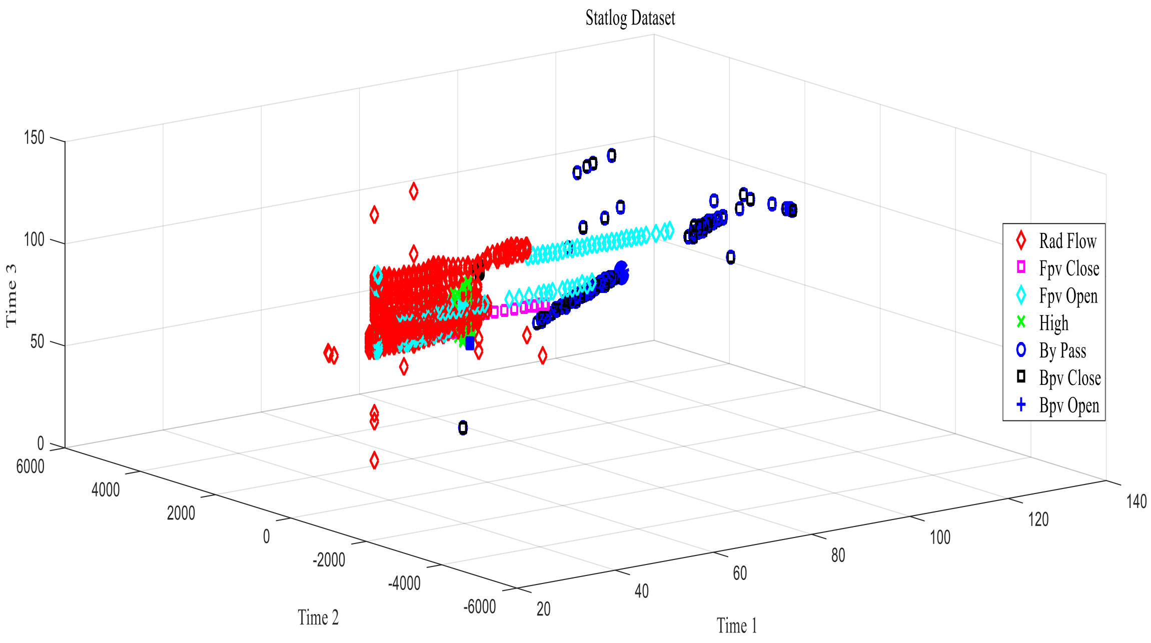

Figure 5.17: Clustering of statlog data objects using ICSO clustering algorithm

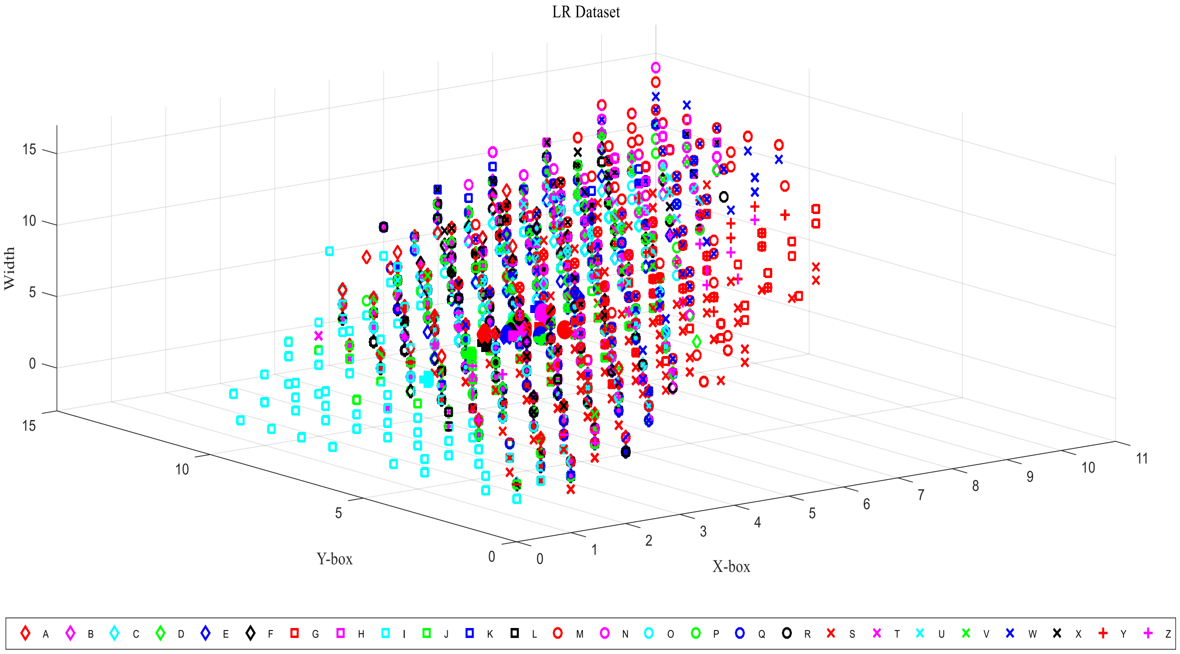

Figure 5.18: Clustering of LR data objects using ICSO clustering algorithm

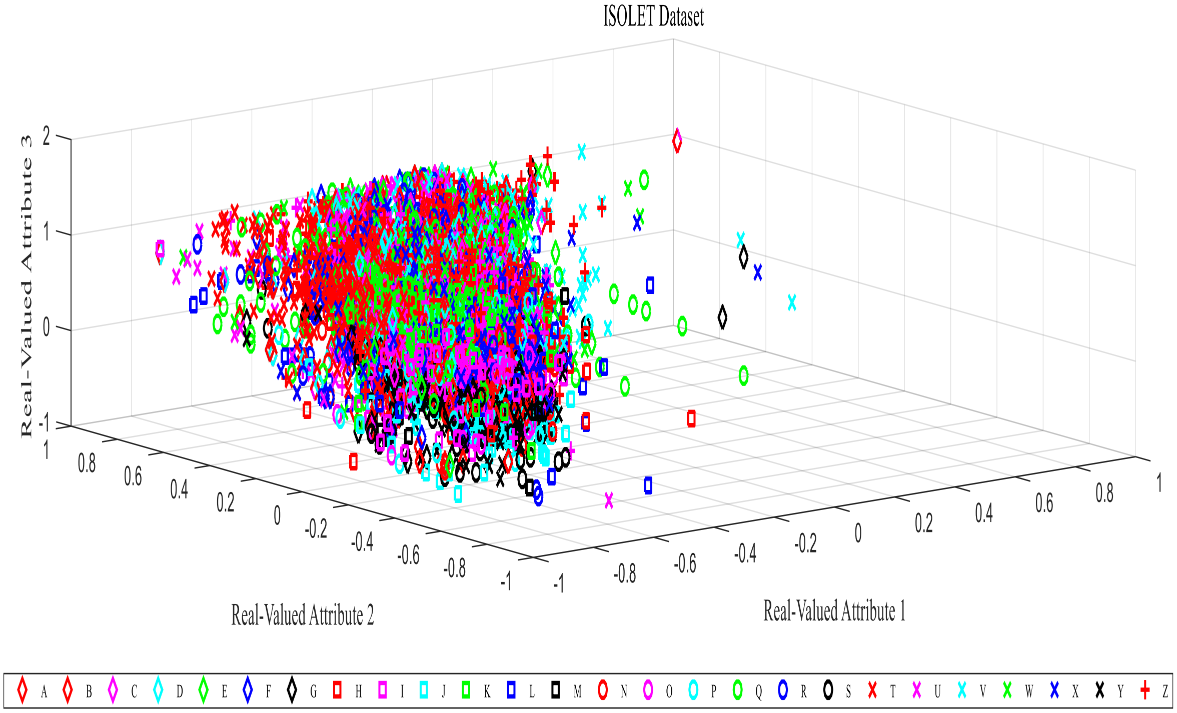

Figure 5.19: Clustering of ISOLET data objects using ICSO clustering algorithm

5.4.2 Statistical Test

This subsection reported the statistical results of Friedman test. The reason behind inclusion of statistical test is explained in subsection 3.4.2. Table 5.2 illustrates the Friedman test results using intra-cluster distance parameter. For Friedman test, two hypotheses are designed; hypothesis (H_0) stands the algorithm having similar performance, while, hypothesis (H_1) stands the algorithm exhibits dissimilar performance. Statistical results showed that ICSO algorithm attains the first rank (1.5) among all other algorithms. The critical and p-value of Friedman test are 12.591 and 0.001202. Hypothesis (H_0) is strongly rejected and ICSO exhibits dissimilar performance than other algorithms. It is sated that ICSO algorithm significantly differs from other algorithms. Table 5.3 discusses the Friedman test results using f-measure parameter. The proposed ICSO algorithm achieves first rank i.e. 1.81 rank among algorithms. Moreover, GA algorithm exhibits poor performance i.e. 5.75 rank among all algorithm using F-measure. The critical and p values are 12.591587 and 0.001136 respectively. Hence, hypothesis (H_0) is rejected and ICSO algorithm exhibits dissimilar performance than other. The ICSO algorithm significantly different than others.

Table 5.2: Statistics of Friedman test over avg. intra-cluster distance parameter

Datasets	Clustering Algorithms						
	K-means	GA	PSO	ACO	CSO	BA	ICSO
Iris	6	7	5	4	3	2	1
Cancer	6	7	3	5	4	2	1
CMC	7	1	6	5	4	2.5	2.5
Wine	7	5	3	4	2	6	1
Glass	1	7	5	6	2	3.5	3.5
Statlog	7	6	4	5	3	2	1
LR	7	5	2	3	4	6	1
ISOLET	3	7	5	6	4	2	1
Sum	44	45	33	38	26	26	12
Rank	5.5	5.63	4.13	4.75	3.25	3.25	1.5

Number of observations: 56	Number of problems: 08	Number of algorithms: 7
Sum of squares of rank sums: 7990	Correction factor: 896	Friedman test statistic: 22.11
Degree of freedom: 6	p-value: 0.001202	Critical value: 12.5915

Table 5.3: Friedman statistical test performed on F-measure parameter

Datasets	Clustering Algorithms						
	K-means	GA	PSO	ACO	CSO	BA	ICSO
Iris	3.5	7	5	6	3.5	2	1
Cancer	3	7	6	5	4	1.5	1.5
CMC	1	7	5	6	4	2.5	2.5
Wine	5	7	6	4	3	1.5	1.5
Glass	2	7	5	6	4	1	3
Statlog	7	5	3	1.5	6	4	1.5
LR	2.5	1	7	5	6	4	2.5
ISOLET	4	5	2	7	6	3	1
Sum	28	46	39	40.5	36.5	19.5	14.5
Rank	3.5	5.75	4.88	5.06	4.56	2.44	1.81

Number of observations: 56	Number of problems: 08	Number of algorithms: 7
Sum of squares of rank sums: 7984	Correction factor: 896	Friedman test statistic: 22.15
Degree of freedom: 6	p-value: 0.001136	Critical value: 12.5915

5.5 Summary

This chapter discuss the improved CSO algorithm for optimizing the clustering problems. Several improvements are inculcated to address the shortcomings of CSO algorithm. These improvements are listed as updated velocity and position update equations for achieving better tradeoff between global and local searches, and convergence rate. A neighborhood strategy is also designed to prevent local optima. It is revealed that ICSO algorithm achieves lower intra cluster distance and higher f-measure rate than other algorithms. Morover, statistical test also confirms the efficacy of ICSO algorithm. It is concluded that ICSO is an efective algorithm for cluster analysis.

CHAPTER 6
CONCLUSION AND FUTURE SCOPE OF WORK

This thesis work addresses the shortcomings associated with the partitional clustering techniques. The aim of this work is to design new meta heuristic algorithms for partitional clustering problems. The primary objectives of this thesis are to develop new clustering algorithms for addressing the accuracy, convergence rate and local optima issues. To address the above mentioned issues, three clustering algorithms are developed in this thesis work. These algorithms are ACRO, IBB-BC and ICSO. The whole work is categorized into three chapters.

This thesis presents a new algorithm based on chemical reactions, called ACRO for clustering. Two operators are integrated into ACRO algorithm i.e. position based operator and neighborhood operator to generate promising solutions. The eight datasets are adopted for evaluating the simulation results of ACRO algorithm and these simulation results are compared with several existing algorithms. Two performance metrics i.e. f-measure and intra cluster distance are considered for performance evaluation of ACRO algorithm. It is stated that significant clustering results are obtained using ACRO algorithm. It is also seen that ACRO is capable to handle local optima problem and provide a better convergence speed.

In this thesis, an IBB-BC algorithm is developed for effective cluster analysis task. This algorithm addresses the convergence and diversity issues of BB-BC algorithm. These issues are handled through chaotic maps and cellular automata-based concepts. The well-known datasets are adopted for evaluating the simulation results of IBB-BC algorithm. The simulation results are compared with established algorithms. The proposed amendments successfully improve the convergence rate and also address the diversity issue of BB-BC algorithm.

This thesis also explores the local optima and tradeoff issues of meta-heuristic algorithm. This thesis also presents an improved CSO algorithm for cluster analysis. It is claimed that exploitation process of CSO algorithm is weak in comparison to exploration process. In turn, tradeoff issues are occurred and performance of CSO algorithm is affected. In this work, an improved velocity and position update mechanisms are developed for better tradeoff. Moreover, neighborhood strategy is designed for preventing local optima and generating feasible solutions. The simulation results are taken over eight datasets and compared with existing algorithms. It is stated that aforementioned amendments improve the simulation results of traditional CSO algorithm. It is observed that significant clustering results are obtained using ICSO algorithm.

In this thesis, three clustering algorithms are developed to address clustering problems effectively. The performance of these algorithms are also compared using f-measure and intra cluster distance parameters. It is noticed that IBB-BC algorithm outperforms than ACRO and ICSO clustering algorithms. It is noticed that size of dataset also affects the performance of clustering algorithm.

6.1 Future Scope

In future, more meta-heuristic algorithms will be explored to determine the optimal solution for clustering problems. Further, new method and neighborhood strategy will be designed for handling local optima problem. The initialization issues of clustering problems will be addressed. Apart from single objective clustering, multi objective clustering will be addressed.

REFERENCES

[1] U. Fayyad, G. S. Piatetsky and P. Smyth, From data mining to knowledge discovery in databases, AI magazine, Vol. 17, No. 3, pp. 37-54, 1996.

[2] J. Han, M. Kamber and J. Pei, Data mining: concepts and techniques: concepts and techniques, Elsevier, 2011.

[3] A. Jain, M. Murty and P. Flynn, "Data clustering", ACM Computing Surveys (CSUR), vol. 31, no. 3, pp. 264-323, 1999.

[4] R. Xu and D. WunschII, "Survey of Clustering Algorithms", IEEE Transactions on Neural Networks, vol. 16, no. 3, pp. 645-678, 2005.

[5] S. Nanda and G. Panda, "A survey on nature inspired metaheuristic algorithms for partitional clustering", Swarm and Evolutionary Computation, vol. 16, pp. 1-18, 2014.

[6] A. Bouguettaya, Q. Yu, X. Liu, X. Zhou and A. Song, "Efficient agglomerative hierarchical clustering", Expert Systems with Applications, vol. 42, no. 5, pp. 2785-2797, 2015.

[7] A. Amini, T. Wah and H. Saboohi, "On Density-Based Data Streams Clustering Algorithms: A Survey", Journal of Computer Science and Technology, vol. 29, no. 1, pp. 116-141, 2014.

[8] A. Ram, S. Jalal, A. Jalal and M. Kumar, "A Density Based Algorithm for Discovering Density Varied Clusters in Large Spatial Databases", International Journal of Computer Applications, vol. 3, no. 6, pp. 1-4, 2010.

[9] S. Nanda, B. Mahanty and M. Tiwari, "Clustering Indian stock market data for portfolio management", Expert Systems with Applications, vol. 37, no. 12, pp. 8793-8798, 2010.

[10] M. Prevezer, " The dynamics of industrial clustering in biotechnology", Small Business Economics, vol. 9, no. 3, pp. 255-271, 1997.

[11] P. Scheunders, "A genetic c-Means clustering algorithm applied to color image quantization", Pattern Recognition, vol. 30, no. 6, pp. 859-866, 1997.

[12] V. Gómez-Muñoz and M. Porta-Gándara, "Local wind patterns for modeling renewable energy systems by means of cluster analysis techniques", Renewable Energy, vol. 25, no. 2, pp. 171-182, 2002.

[13] S. Mitra and H. Banka, "Multi-objective evolutionary biclustering of gene expression data", Pattern Recognition, vol. 39, no. 12, pp. 2464-2477, 2006.

[14] M. Dorigo, M. Birattari and T. Stutzle, "Ant Colony Optimization", IEEE Computational Intelligence Magazine, vol. 1, no. 4, pp. 28-39, 2006.

[15] T. Cura, "A particle swarm optimization approach to clustering", Expert Systems with Applications, vol. 39, no. 1, pp. 1582-1588, 2012.

[16] Y. Kumar and G. Sahoo, "A charged system search approach for data clustering", Progress in Artificial Intelligence, vol. 2, no. 2-3, pp. 153-166, 2014.

[17] D. Karaboga and C. Ozturk, "A novel clustering approach: Artificial Bee Colony (ABC) algorithm", Applied Soft Computing, vol. 11, no. 1, pp. 652-657, 2011.

[18] N. Kushwaha, M. Pant, S. Kant and V. Jain, "Magnetic optimization algorithm for data clustering", Pattern Recognition Letters, vol. 115, pp. 59-65, 2018.

[19] A. Hatamlou, "Black hole: A new heuristic optimization approach for data clustering", Information Sciences, vol. 222, pp. 175-184, 2013.

[20] A. Hatamlou, S. Abdullah and M. Hatamlou, "Data clustering using big bang–big crunch algorithm.", in In International conference on innovative computing technology, Berlin, Heidelberg., pp. 383-388, 2011.

[21] J. Irani, N. Pise and M. Phatak, "Clustering Techniques and the Similarity Measures used in Clustering: A Survey", International Journal of Computer Applications, vol. 134, no. 7, pp. 9-14, 2016.

[22] P. Mohanty, S. Nayak, U. Mohapatra and D. Mishra, "A survey on partitional clustering using single-objective metaheuristic approach", International Journal of Innovative Computing and Applications, vol. 10, no. 34, p. 207, 2019.

[23] A. Huang, "Similarity measures for text document clustering.", in In Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand, Vol. 4, pp. 9-56, 2008.

[24] M. R. Anderberg, Cluster Analysis for Application, Academic Press, New York, 1973.

[25] E. Alpaydin, Introduction to machine learning. MIT press, 2004.

[26] J. A. Hartigan, Clustering Algorithms, Wiley, New York, 1975.

[27] A. K. Jain and R. C. Dubes, Algorithms for Clustering Data, Prentice-Hall, Englewood, 1988.

[28] A. Khandare and A. Alvi, "Survey of Improved k-means Clustering Algorithms: Improvements, Shortcomings and Scope for Further Enhancement and Scalability.", In Information Systems Design and Intelligent Applications, New Delhi, pp. 495-503, 2016.

[29] R. Qaddoura, H. Faris and I. Aljarah, "An efficient clustering algorithm based on the k-nearest neighbors with an indexing ratio", International Journal of Machine Learning and Cybernetics, vol. 11, no. 3, pp. 675-714, 2019.

[30] A. Rodríguez, E. Cuevas, D. Zaldívar, M. Pérez-Cisneros, G. García-Gil and B. Morales-Castañeda, "An improved clustering method based on biological visual models", Applied Mathematical Modelling, vol. 85, pp. 174-191, 2020.

[31] J. Bezdek, S. Boggavarapu, L. Hall and A. Bensaid, "Genetic algorithm guided clustering", In Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence, pp. 34-39, 1994.

[32] P, Shelokar, V. Jayaraman and B. Kulkarni, "An ant colony approach for clustering", Analytica Chimica Acta, vol. 509, no. 2, pp. 187-195, 2004.

[33] Y. Liu, Z. Yi, H. Wu, M. Ye and K. Chen, "A tabu search approach for the minimum sum-of-squares clustering problem", Information Sciences, vol. 178, no. 12, pp. 2680-2704, 2008.

[34] M. Mahdavi, M. Chehreghani, H. Abolhassani and R. Forsati, "Novel meta-heuristic algorithms for clustering web documents", Applied Mathematics and Computation, vol. 201, no. 1-2, pp. 441-451, 2008.

[35] B. Santosa and M. Ningrum, "Cat swarm optimization for clustering.", in In 2009 International Conference of Soft Computing and Pattern Recognition, 2009, pp. 54-59.

[36] C. Zhang, D. Ouyang and J. Ning, "An artificial bee colony approach for clustering", Expert Systems with Applications, vol. 37, no. 7, pp. 4761-4767, 2010.

[37] S. Satapathy and A. Naik, "Data clustering based on teaching-learning-based optimization.", In International conference on swarm, evolutionary, and memetic computing, Berlin, Heidelberg., pp. 148-156, 2011.

[38] J. Senthilnath, S. Omkar and V. Mani, "Clustering using firefly algorithm: Performance study", Swarm and Evolutionary Computation, vol. 1, no. 3, pp. 164-171, 2011.

[39] A. Hatamlou, "In search of optimal centroids on data clustering using a binary search algorithm", Pattern Recognition Letters, vol. 33, no. 13, pp. 1756-1760, 2012.

[40] M. Taherdangkoo, M. Hossein Shirzadi, M. Yazdi and M. Hadi Bagheri, "A robust clustering method based on blind, naked mole-rats (BNMR) algorithm", Swarm and Evolutionary Computation, vol. 10, pp. 1-11, 2013.

[41] M. Malinen, R. Mariescu-Istodor and P. Fränti, "K-means∗: Clustering by gradual data transformation", Pattern Recognition, vol. 47, no. 10, pp. 3376-3386, 2014.

[42] Y. Kumar, S. Gupta, D. Kumar and G. Sahoo, A clustering approach based on charged particles, Optimization Algorithms-Methods and Applications, pp. 245-263, 2016.

[43] S. Łukasik, P. Kowalski, M. Charytanowicz and P. Kulczycki, "Data clustering with grasshopper optimization algorithm", in Federated Conference on Computer Science and Information Systems (FedCSIS), pp. 71-74, 2017.

[44] S. Deb, Z. Tian, S. Fong, R. Wong, R. Millham and K. Wong, "Elephant search algorithm applied to data clustering", Soft Computing, vol. 22, no. 18, pp. 6035-6046, 2018.

[45] J. Nasiri and F. Khiyabani, "A whale optimization algorithm (WOA) approach for clustering", Cogent Mathematics & Statistics, vol. 5, no. 1, 2018.

[46] M. Alswaitti, M. Ishak and N. Isa, "Optimized gravitational-based data clustering algorithm", Engineering Applications of Artificial Intelligence, vol. 73, pp. 126-148, 2018.

[47] F. Kuwil, Ü. Atila, R. Abu-Issa and F. Murtagh, "A novel data clustering algorithm based on gravity center methodology", Expert Systems with Applications, vol. 156, p. 113435, 2020.

[48] J. Hartigan and M. Wong, "Algorithm AS 136: A K-Means Clustering Algorithm", Applied Statistics, vol. 28, no. 1, p. 100, 1979.

[49] B. Zhang, M. Hsu and U. Dayal, "K-Harmonic Means -A Spatial Clustering Algorithm with Boosting", In International Workshop on Temporal, Spatial, and Spatio-Temporal Data Mining, Springer, Berlin, Heidelberg, pp. 31-45, 2000.

[50] K. Žalik, "An efficient k′-means clustering algorithm", Pattern Recognition Letters, vol. 29, no. 9, pp. 1385-1391, 2008.

[51] F. Cao, L. Jiye and J. Guang, "An initialization method for the K-Means algorithm using neighborhood model.", Computers & Mathematics with Applications, vol. 58, no. 3, pp. 474-483, 2009.

[52] J. Xiao, Y. Yan, J. Zhang and Y. Tang, "A quantum-inspired genetic algorithm for k-means clustering", Expert Systems with Applications, vol. 37, no. 7, pp. 4966-4973, 2010.

[53] H. Jiang, S. Yi, J. Li, F. Yang and X. Hu, "Ant clustering algorithm with K-harmonic means clustering", Expert Systems with Applications, vol. 37, no. 12, pp. 8679-8684, 2010.

[54] T. Niknam, E. Fard, S. Ehrampoosh and A. Rousta, "A new hybrid imperialist competitive algorithm on data clustering", Sadhana, vol. 36, no. 3, pp. 293-315, 2011.

[55] M. Erisoglu, N. Calis and S. Sakallioglu, "A new algorithm for initial cluster centers in k-means algorithm", Pattern Recognition Letters, vol. 32, no. 14, pp. 1701-1705, 2011.

[56] H. He and Y. Tan, "A two-stage genetic algorithm for automatic clustering", Neurocomputing, vol. 81, pp. 49-59, 2012.

[57] R. Liu, L. Jiao, X. Zhang and Y. Li, "Gene transposon based clone selection algorithm for automatic clustering", Information Sciences, vol. 204, pp. 1-22, 2012.

[58] D. Chang, Y. Zhao, C. Zheng and X. Zhang, "A genetic clustering algorithm using a message-based similarity measure", Expert Systems with Applications, vol. 39, no. 2, pp. 2194-2202, 2012.

[59] S. Khan and A. Ahmad, "Cluster center initialization algorithm for K-modes clustering", Expert Systems with Applications, vol. 40, no. 18, pp. 7444-7456, 2013.

[60] G. Tzortzis and A. Likas, "The MinMax k-Means clustering algorithm", Pattern Recognition, vol. 47, no. 7, pp. 2505-2516, 2014.

[61] H. Peng, J. Wang, P. Shi, A. Riscos-Núñez and M. Pérez-Jiménez, "An automatic clustering algorithm inspired by membrane computing", Pattern Recognition Letters, vol. 68, pp. 34-40, 2015.

[62] H. Menéndez, F. Otero and D. Camacho, "Medoid-based clustering using ant colony optimization", Swarm Intelligence, vol. 10, no. 2, pp. 123-145, 2016.

[63] M. Zhao, H. Tang, J. Guo and Y. Sun, "A data clustering algorithm using cuckoo search", in In Frontier Computing, Singapore, pp. 225-230, 2016.

[64] F. Huang, X. Li, S. Zhang and J. Zhang, "Harmonious Genetic Clustering", IEEE Transactions on Cybernetics, vol. 48, no. 1, pp. 199-214, 2018.

[65] S. Pal and S. Pal, "Black Hole and k-Means Hybrid Clustering Algorithm", In Computational Intelligence in Data Mining, Springer, Singapore, pp. 403-413, 2020.

[66] F. Yang, T. Sun and C. Zhang, "An efficient hybrid data clustering method based on K-harmonic means and Particle Swarm Optimization", Expert Systems with Applications, vol. 36, no. 6, pp. 9847-9852, 2009.

[67] M. Yin, Y. Hu, F. Yang, X. Li and W. Gu, "A novel hybrid K-harmonic means and gravitational search algorithm approach for clustering", Expert Systems with Applications, vol. 38, no. 8, pp. 9319-9324, 2011.

[68] X. Yan, Y. Zhu, W. Zou and L. Wang, "A new approach for data clustering using hybrid artificial bee colony algorithm", Neurocomputing, vol. 97, pp. 241-250, 2012.

[69] C. Huang, W. Huang, H. Chang, Y. Yeh and C. Tsai, "Hybridization strategies for continuous ant colony optimization and particle swarm optimization applied to data clustering", Applied Soft Computing, vol. 13, no. 9, pp. 3864-3872, 2013.

[70] A. Hatamlou and M. Hatamlou, "PSOHS: an efficient two-stage approach for data clustering", Memetic Computing, vol. 5, no. 2, pp. 155-161, 2013.

[71] B. Jiang and N. Wang, "Cooperative bare-bone particle swarm optimization for data clustering", Soft Computing, vol. 18, no. 6, pp. 1079-1091, 2013.

[72] Y. Kumar and G. Sahoo, "A hybrid data clustering approach based on improved cat swarm optimization and K-harmonic mean algorithm", AI Communications, vol. 28, no. 4, pp. 751-764, 2015.

[73] R. Wang, Y. Zhou, S. Qiao and K. Huang, "Flower Pollination Algorithm with Bee Pollinator for cluster analysis", Information Processing Letters, vol. 116, no. 1, pp. 1-14, 2016.

[74] A. Pakrashi and B. Chaudhuri, "A Kalman filtering induced heuristic optimization based partitional data clustering", Information Sciences, vol. 369, pp. 704-717, 2016.

[75] Y. kumar and G. Sahoo, "A two-step artificial bee colony algorithm for clustering", Neural Computing and Applications, vol. 28, no. 3, pp. 537-551, 2015.

[76] X. Han, L. Quan, X. Xiong, M. Almeter, J. Xiang and Y. Lan, "A novel data clustering algorithm based on modified gravitational search algorithm", Engineering Applications of Artificial Intelligence, vol. 61, pp. 1-7, 2017.

[77] L. Abualigah, A. Khader, E. Hanandeh and A. Gandomi, "A novel hybridization strategy for krill herd algorithm applied to clustering techniques", Applied Soft Computing, vol. 60, pp. 423-435, 2017.

[78] Y. Zhou, Y. Zhou, Q. Luo and M. Abdel-Basset, "A simplex method-based social spider optimization algorithm for clustering analysis", Engineering Applications of Artificial Intelligence, vol. 64, pp. 67-82, 2017.

[79] S. Ishak Boushaki, N. Kamel and O. Bendjeghaba, "A new quantum chaotic cuckoo search algorithm for data clustering", Expert Systems with Applications, vol. 96, pp. 358-372, 2018.

[80] K. Lakshmi, N. Visalakshi and S. Shanthi, "Data clustering using K-Means based on Crow Search Algorithm", Sādhanā, vol. 43, no. 11, 2018.

[81] A. Bouyer and A. Hatamlou, "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.

[82] K. Bijari, H. Zare, H. Veisi and H. Bobarshad, "Memory-enriched big bang–big crunch optimization algorithm for data clustering", Neural Computing and Applications, vol. 29, no. 6, pp. 111-121, 2016.

[83] Y. Kumar and P. Singh, "Improved cat swarm optimization algorithm for solving global optimization problems and its application to clustering", Applied Intelligence, vol. 48, no. 9, pp. 2681-2697, 2017.

[84] Y. Kumar and P. Singh, "A chaotic teaching learning based optimization algorithm for clustering problems", Applied Intelligence, vol. 49, no. 3, pp. 1036-1062, 2018.

[85] Abdulwahab, H. A., Noraziah, A., Alsewari, A. A., & Salih, S. Q. (2019). An enhanced version of black hole algorithm via levy flight for optimization and data clustering problems. IEEE Access, 7, 142085-142096.

[86] A. Kaur, S. Pal and A. Singh, "Hybridization of Chaos and Flower Pollination Algorithm over K-Means for data clustering", Applied Soft Computing, p. 105523, 2019.

[87] H. Ismkhan, "I-k-means−+: An iterative clustering algorithm based on an enhanced version of the k-means", *Pattern Recognition*, vol. 79, pp. 402-413, 2018.

[88] A. Rezaee Jordehi, "Chaotic bat swarm optimisation (CBSO)", *Applied Soft Computing*, vol. 26, pp. 523-530, 2015.

[89] Y. Aboubi, H. Drias and N. Kamel, "BAT-CLARA: BAT-inspired algorithm for Clustering LARge Applications", *IFAC-PapersOnLine*, vol. 49, no. 12, pp. 243-248, 2016.

[90] B. Alatas, "ACROA: Artificial Chemical Reaction Optimization Algorithm for global optimization", Expert Systems with Applications, vol. 38, no. 10, pp. 13170-13180, 2011.

[91] B. Alatas, "A novel chemistry based metaheuristic optimization method for mining of classification rules", Expert Systems with Applications, vol. 39, no. 12, pp. 11080-11088, 2012.

[92] O. Erol and I. Eksin, "A new optimization method: Big Bang–Big Crunch", Advances in Engineering Software, vol. 37, no. 2, pp. 106-111, 2006.

[93] A. Uzun, T. Usta, E. Dündar and E. Korkmaz, "A solution to the classification problem with cellular automata", Pattern Recognition Letters, vol. 116, pp. 114-120, 2018.

[94] S. Chu, P. Tsai and J. Pan, "Cat swarm optimization", In Pacific Rim international conference on artificial intelligence Berlin, Heidelberg, 2006, pp. 854-858.

[95] Z. Wang, C. Chang and M. Li, "Optimizing least-significant-bit substitution using cat swarm optimization strategy", Information Sciences, vol. 192, pp. 98-108, 2012.

PUBLICATIONS FROM THESIS

Journals

Accepted

1. H. Singh, Y. Kumar and S. Kumar, "A new meta-heuristic algorithm based on chemical reactions for partitional clustering problems", Evolutionary Intelligence, vol. 12, no. 2, pp. 241-252, 2019. **(Published, Scopus)**
2. H. Singh and Y. Kumar, "Cellular Automata Based Model for E-Healthcare Data Analysis", International Journal of Information System Modeling and Design, vol. 10, no. 3, pp. 1-18, 2019. **(Published, Scopus)**
3. H. Singh and Y. Kumar, "A neighborhood search based cat swarm optimization algorithm for clustering problems", Evolutionary Intelligence, pp.1-17, 2020. **(Published, Scopus)**
4. H. Singh and Y. Kumar, "Enhanced Cat Swarm Optimization Algorithm for cluster analysis ", International Journal of Applied Metaheuristic Computing (IJAMC), IGI Global. vol. 13, no. 2, 2020. **(Published, Scopus)**

Communicated

1. H. Singh and Y. Kumar, "Improved BAT Algorithm for cluster analysis ", Applied Intelligence. **(SCI, Communicated)**

Conferences

1. H. Singh and Y. Kumar, "Hybrid Artificial Chemical Reaction Optimization Algorithm for Cluster Analysis", Procedia Computer Science, vol. 167, pp. 531-540, 2020. **(Published, Scopus)**
2. H. Singh and Y. Kumar, "Hybrid Big Bang-Big Crunch Algorithm for Cluster Analysis", Futuristic Trends in Networks and Computing Technologies, pp. 648-661, 2020. **(Published, Scopus)**