A Dissertation

On

Big Data Analysis Using Metaheuristic Algorithms

By

ASHISH KUMAR TRIPATHI

Roll No. 2k14/Ph.D/CO/03

Under the Joint Supervision of

Dr. Kapil Sharma
Professor, & Head
Department of Information Technology,
Delhi Technological University

Dr. Manju Bala
Assistant Professor,
Department of Computer Science
IP College of Women, Delhi University

Submitted in fulfillment of the requirements of the degree of
Doctor of Philosophy to the

Delhi Technological University
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi 110042
2018

DECLARATION

I, Ashish Kumar Tripathi, Ph.D. student (Roll No. 2k14/Ph.D/CO/03), hereby declare that the thesis entitled **"Big Data Analysis Using Metaheuristic Algorithms"** which is being submitted for the award of the degree of Doctor of Philosophy in Computer Science & Engineering, is a record of bonafide research work carried out by me in the Department of Computer Science & Engineering, Delhi Technological University. I further declare that this work is based on original research and has not been submitted to any university or institution for any degree or diploma.

Date: _____

Place: New Delhi

Ashish Kumar Tripathi

2k14/Ph.D/CO/03

Department of Computer Science & Engineering

Delhi Technological University (DTU)

New Delhi -110042

CERTIFICATE

DELHI TECHNOLOGICAL UNIVERSITY

(Govt. of National Capital Territory of Delhi)

BAWANA ROAD, DELHI - 110042

Date: _____

This is to certify that the work embodied in the thesis titled "BIG DATA ANALYSIS USING METAHEURISTIC ALGORITHMS" has been completed by **Ashish Kumar Tripathi** under our supervision and guidance towards fulfillment of the requirements for the degree of Doctor of Philosophy of Delhi Technological University, Delhi. This work is based on original research and has not been submitted in full or in part for any other diploma or degree of any university.

Dr. Kapil Sharma

Professor, & Head

Department of Information Technology,

Delhi Technological University

Dr. Manju Bala

Assistant Professor,

Department of Computer Science

IP College of Women, Delhi University

ACKNOWLEDGMENT

Sometimes it becomes difficult to express one's deep regards in words. An achievement of the work like this needs the support, blessings and motivation of others. I take this opportunity to express my sense of gratitude and great respect to all those who helped me through the duration of this Ph.D. work.

First and foremost, I would like to thank Dr. Kapil Sharma, Head of Department, Department of Information Technology, for his invaluable guidance, encouragement and support in carrying out my research work. He helped in creating a healthy environment for learning and gave me full academic freedom to pursue my research with confidence. He has been very kind with me and his polite attitude made me bridge all the shortcomings during continuance of this research work. This period has been an enriching experience of working under his guidance. He always gave priority to my needs and reposed faith in my abilities. I am really thankful for his endless support and encouragement throughout this work.

Further, I am thankful to Dr. Manju Bala, Assistant Professor, IP College Of Women, Delhi University, who has been very kind in shaping my research and without whom this work would not have seen the light of the day. Her wisdom, intelligence, constant support and endeavor to seek perfection transformed many aspects of my life, made me a better person and triggered in me the miraculous drive to work hard. Her academic and professional sharpness, dedication and strong commitment towards the work have always been a source of inspiration for me. Whenever I was in low confidence or in stress, she encouraged and motivated me. Due to her invaluable support and intelligent inputs, my research period passed very smoothly. Besides being my mentor, she has been a pillar of strength and I am deeply indebted to her for guiding me with her extensive knowledge and insightful discussions. I hope for and look forward to her continued guidance in future.

I am thankful to Prof. Yogesh Singh, Vice Chancellor, Delhi Technological University. He has been a constant source of inspiration for me throughout my research and teaching career. I am also thankful to my wife, Shikha Tripathi, Research Scholar, CSJM University Kanpur for all her help and support during this research work. She always encouraged me and kept my spirits high during thick and thin. I am thankful to her in helping me in formatting the thesis and reading the drafts of the chapters in the thesis. Her suggestions were useful and have been incorporated in the thesis. Her constant involvement and guidance helped a lot in bringing the thesis in its current form. I acknowledge with thanks the contribution of all my other teachers and friends for their direct and indirect help and support in this work.

A special thanks to my family. This work could not have been possible without the unrelenting support of my family, particularly my parents, Mr. Sarvesh Tripathi and Mrs. Rekha Tripathi. They supported me in every possible way to ease my research journey. I pay my sincere regards to my parents and parents in-law for their unconditional love and for inculcating confidence in me to pursue a career in academics. The work would not have been possible without the smiles of my son, Reyansh Tripathi who has been deprived from the love and care that he deserve during my Ph.D work.

Ashish Kumar Tripathi

ABSTRACT

Big Data has got the huge attention of the researchers from academia and industry for the decision and strategy making. Thus, efficient data analysis methods are required for managing the big data sets. Data clustering, a prominent analysis method of data mining, is being efficiently employed in big data analysis since it does not require labeled datasets, which is not easily available for the big data problems. K-means, one of the simplest and popular algorithm, has been employed for unfolding the various clustering problems. However, the results of K-means algorithm are highly dependent on initial cluster centroids and easily traps into local optima. To mitigate this issue, a novel metaheuristic algorithm named Military Dog Based Optimizer has been introduced and validated against 17 benchmark functions. The proposed algorithm has been also tested on 8 benchmark clustering datasets and compared with other 5 recent state-of-the-art algorithms. Though, the proposed algorithm witnessed better clustering in terms of accuracy as compared to the conventional methods. However, the algorithm fail to perform efficiently on the big datasets in terms of memory space and the time complexities, due to their sequential execution. To overcome this issue, four novel methods have been developed for the efficient clustering of the big datasets. The first method is a hybrid of K-means and bat algorithm which run in parallel over a cluster of computers. The proposed method outperformed K-means, PSO and bat algorithm on 5 benchmark datasets. The second method is a novel variant of the grey wolf optimizer for clustering the big data set, in which the exploration and exploitation ability

of the grey wolf optimizer is enhanced using the levy flight and binomial crossover. The proposed method performed efficiently on the 8 benchmark clustering datasets as compared to the conventional methods. Moreover, the parallel performance of the presented methods has been also analyzed using the speedup measure. Third, a hybrid method named K-BBO has been developed which utilizes the search ability of the biogeography based optimizer and K-means for better initial population. Fourth, a novel parallel method using MDBO is introduced and tested on four large scale datasets. Furthermore, to test the applicability of the proposed methods in real world scenarios, two real-world problems namely, Twitter sentiment analysis and fake review detection have been solved in the big data environment using the proposed methods.

Contents

List of Tables

List of Figures

List of Publications

Papers in International Journals

1. Tripathi, A. K., Sharma, K., & Bala, M. (2017). Dynamic frequency based parallel k-bat algorithm for massive data clustering (DFBPKBA). International Journal of System Assurance Engineering and Management, Vol 9, pp 866-874. **[Published, Scopus Indexed]**.

2. Tripathi, A. K., Sharma, K., & Bala, M. (2018). A Novel Clustering Method Using Enhanced Grey Wolf Optimizer and MapReduce. Big Data Research, Elsevier, Vol. (14), pp 93-100. **[Published, SCI Indexed]**.

3. Tripathi, A. K., Sharma, K., & Bala, M.(2018). Parallel Hybrid BBO search method for twitter sentiment analysis of large scale datasets using MapReduce. International Journal of Information Security and Privacy (IJISP), IGI Global, Vol (13). **[Published, Scopus Indexed]**.

4. Tripathi, A. K., Sharma, K., Bala, M. (2018) Military Dog Based Optimizer and its application to fake review detection. Journal of applied intelligence, Springer **[Communicated**].

Papers in International Conferences

5. Tripathi, A. K., Sharma, K., Bala, M. (2017) Parallel Bat Algorithm-Based Clustering Using MapReduce. In the Proc. of Data Knowledge Engineering. Springer, Singapore. **[Published]**

6. Tripathi, A. K., Sharma, K., Bala, M. (2018) Big data approach for fake review detection using parallel biogeography based algorithm. In the Proc. of Computers and Management, Elsevier, New Delhi, India. **[Published]**.

Abbreviations

ABC	**Artificial Bee Colony**
AIS	**Artificial Immune System Based Clustering**
ACO	**Ant Colony Optimization**
ASF	**Artificial Swarm fish**
ANN	**Artificial Neural Network**
BA	**Bat Algorithm**
BBO	**Biogeography Based Optimization**
BM	**Barking Movement**
CS	**Cuckoo Search**
CSO	**Curved Space Optimization (CSO)**
CUDA	**Compute Unified Device Architecture**
CHAID	**Chi-squared Automatic Interaction Detection**
CHI	**Chi-square**
DE	**Differential Evolution**
DFBPKBA	**Dynamic Frequency Based Parallel K-Bat Algorithm**
EA	**Evolutionary Algorithm**
IBBO	**Enhanced Bio-geography based optimization**
EGWO	**Enhanced Grey Wolf Optimizer**
ES	**Evolutionary Strategy**

ERCIM	**European Research Consortium for Informatics and Mathematics**
FA	**Firefly algorithm**
FCM	**Fuzzy C-Means**
FCS	**Fuzzy C- Shells**
FSV	**Feasibility Solution Vector**
FR	**Functional Requirements**
GA	**Genetic Algorithm**
GbSA	**Galaxy-based Search Algorithm**
GGA	**Grouping Genetic Algorithm**
GPU	**Graphics Processing unit**
GSA	**Gravitational Search Algorithm**
GWO	**Gray Wolf Optimizer**
GFS	**Google File System**
HS	**Harmony Search**
HSI	**Habitat Suitability Index**
HDFS	**Hadoop Distributed File System**
IDC	**International Data Corporation**
IWO	**Invasive Weed Optimization**
IOT	**Internet Of Things**
LR	**Logistic Regression**
KHA	**krill Heard Algorithm**
HKHA	**Hybrid krill heard algorithm**
MCS	**Magnetic charged system search**
MCSO	**Multiverse Cat Swarm Optimization**
MD	**Military Dog**

MDBO	**Military Dog Based Optimization**
MDS	**Military Dos Squad**
MDSI	**Military Dog Suitability Index**
MNC	**Multi National Company**
MVO	**Multi Verse Optimizer**
ML	**Machine Learning**
MR-KBBO	**Map Reduce based BBO**
NIST	**National Institute of Standards and Technology**
NLTK	**Natural Language Tool Kit**
NG	**NewsGroup**
NGL	**Ng-Goh-Low**
NLP	**Natural Language Processing**
NN	**Neural Network**
OS	**Operating System**
OSS	**Open Source Software**
PBILL	**Probability based Incremental Learning**
P K-Means	**Parallel k-Means**
PPI	**Protein Protein Interactions**
PSO	**Particle Swarm Optimization**
PCM	**Probabilistic C Means**
POS	**Part of Speech**
PSOC	**Particle Swarm Optimization Classifier**
PSVM	**Probabilistic Framework for SVM**
QPSO	**Quantum Behaved PSO**
SA	**Simulated Annealing**

SFL	**Shuffled Frog Leaping**
SMO	**Small-World Optimization**
SMOP	**Spider Monkey Optimization**
SM	**Sniffing Movement**
SMA	**Stock Market Analysis**
SOM	**Self Organizing Maps**
SVD	**Singular Value Decomposition**
SVM	**Support Vector Machine**
TF	**Term Frequency**
UCI	**University of California Irvine**
WM	**Web Mining**
WSN	**Wireless Sensor Network**

Chapter 1

Introduction

This chapter introduces the big data and its importance in today's world. The motivation and objective of the work are highlighted along with the overview of the work done. Chapter wise thesis organization is presented at the end of the chapter.

1.1 Overview

Last one decade has shown mountainous growth of the digital data. It is anticipated that the data produced from mobile and wireless devices will share 66% of the total IP traffic by the end of 2020 [1]. The social media sites, such as Instagram, Face-book, Twitter etc., has major contributions in the growth of the digital data due to their penetration in the day to day life of the people. According to the Howard [2], Facebook is used to organize the protest, Twitter is used to harmonize and YouTube to express the thoughts in the form of video. In these scenarios and availability of the data generated by the people, efficient analysis is inevitable which stimulates the requirement of parallel and distributed methods for the analysis of the big datasets. Big data is the popular term used to describe amounts of

data so large and complex that it becomes difficult to process or analyze using traditional methods. The increasing growth of the digital data has raised new challenges to the existing data analysis methods. Responding to these challenges, various tools and methods has been developed for analyzing the big datasets. The data analysis methods are broadly classified into supervised and unsupervised. The supervised models require labeled datasets to train themselves. On the contrary, unsupervised learning models work on unlabeled datasets to induce the learning model. The main source of the big datasets are the social media, sensors devises, satellites, and mobile devices and generally, the labels of such datasets are not available. Therefore, the applicability of the supervised models is limited for the analysis of the big datasets. Therefore, in this work the unsupervised learning based models have been explored. Clustering is one of the popular unsupervised data analysis technique that groups data items in multiple clusters with maximum intra-cluster similarity and minimum inter-cluster similarity [3]. Data clustering is widely used in the many domains like image segmentation, data mining, bio-medical and information retrieval. Over the past several years, a number of methods have been proposed for solving clustering problems. K-means, one of the simplest and popular algorithm, has been employed for unfolding the various clustering problems [4]. However, the results of K-means algorithm are highly dependent on initial cluster centroids and it's probability of trapping into local optima is high [5]. To mitigate this issue, metaheuritic based methods have been proposed in literature [6, 7, 8]. Maulik et al. [9] used the ability of genetic algorithm to find the best centroid in the feature space so that the compactness of the resulting clusters is optimized. Kumar et al. [10] introduced a clustering algorithm which imitate the hunting behavior of grey wolves. Zhang et al. [11] proposed a variant of grey wolf optimizer for optimizing the clustering process. Hatamlou et al. [8] introduced gravitational search algorithm based clustering method which used

K-means algorithm for initializing the center heads. Karaboga et al. [7] proposed a novel artificial bee colony based method for clustering the multivariate data. Cura et al. [12] developed particle swarm optimization based clustering method and solved the web application problems. However, the metaheuristic based computation is facing new challenges due to the increasing applications and complexity of the real-world optimization problems [13]. The application of the metaheuristic algorithms pertaining engineering and real world problems is strongly limited because of high computational cost. When dealing with the optimization related to big data sets, the computational cost of fitness evaluation increases rapidly. In such scenario the algorithms which runs sequentially, fail to give satisfactory results within the given time domain. Therefore, the improvements in the traditional sequential computational model is required to handle big data sets in reasonable amount of time.

For alleviating computation performance on big dataset, parallel and distributed computation has exhibited attractive solutions. With the years of progress in technology, Hadoop and MapReduce is a widely used parallel processing tool. Hadoop [14] is an open source platform, developed and managed by Apache for handling large datasets using distributed processing. Hadoop works with its own file system referred as HDFS (hadoop distributed file system) and, is capable of processing zeta bytes of data with commodity hardware [15]. MapReduce is a parallel programming model which works on the top of the Hadoop [16, 17, 18]. The proposed work has successfully utilized the architecture of Hadoop for the distributed processing and MapReduce model for the parallel programming. Moreover, the existing algorithms are modified and novel variants are proposed that fits into the big data environment. Further, the proposed parallel methods are successfully utilized to solve sentiment analysis and fake review detection problem. The remainder of the chapter gives

a brief introduction of the field and explains the basic concepts of the work carried out in this thesis.

1.2 Big Data

Big Data is defined as dataset, so large and complex that it can not be handled by the traditional database management systems and algorithms. Generally, big data is used to analyze unstructured data, means the data for which schema is not properly defined and it is heterogeneous in nature. However, contrary to this notion, the structured data can also be treated as big data if the traditional sequential algorithms are not able to process it within the reasonable time. A number of data scientists and organizations has defined the term big data in their own ways. A systematic literature of the same has been presented in the next chapter. Gartner and Laney [19] introduced the 3Vs concept defining big data, as high volume, velocity and variety information that require new forms of data processing. However, till date 8Vs of the big has been presented in the literature which as explained as follows.

1.2.1 The 8 Vs of Big Data

1. **Volume** Volume refers to the huge size of the data that becomes beyond the abilities of the conventional hardware and methods to handle. The size of data determine, whether it should be considered as the big data or not.

2. **Velocity** It refers to the speed with which the data is generated. For example, the streaming data such as tweets of the twitter, facebook comments, and customer reviews generated for e-commerce products. The traditional data base techniques are not able to handle such fast streaming real time data sets.

3. **Variety** It means the data can be in any format such as structured, unstructured or semi-structured. The unstructured data does not have proper schema and it may consists of text, audio or video. For example call records, sales records and spreadsheets are the examples of the unstructured data. Further, semi-structured data are like the structured data but it is not organized in RDBMS format rather separated by tags or with some other markers.

4. **Veracity** Veracity deals with the reliability or the trust of source of the dataset. It also refers that how meaningful it is to do analysis of the data with confidence.

5. **Volatility** It refers to the life of the stored data, means for how much time it is valid. For example, the one time password (OTP) of the banking applications is valid for few minutes.

6. **Value** It refers the importance of the data from the business perspective. The value of the big data deals with the revenue generation, customers satisfaction, profit and relationship of the business man with the customer.

7. **Visualization** Traditional data visualization tools face difficulty due to poor scalability and response time. Traditional graphs are not appropriate, for plotting billions of data points. Thus, contemporary tools and techniques are required for representing big data sets.

8. **Vulnerability** It refers that how vulnerable your data is, in terms of security. A data breach pertaining to the big data has a bigger impact, for example AshleyMadison hack in the year 2015. Thus, the security of the big data is also a prime concern, and should be handled carefully.

1.2.2 The Impact of Big Data on Human Social Life

The data revolution has made a broad impact on the humanity. The social impact of things are only as good or bad as the human intentions behind them are. John Battelle, the chief executive officer (CEO) of Federated Media, quoted that "The era of Big Data is an important inflection point in human history and represents a critical moment in our civilization's development". It will impact peoples interaction with business, government and each other. The decisions made during this era will decide the kind of world we will make for the future generations. The big question about the big data is that, how and who can control the generation of the data [20]. What access can be allowed to the citizens and the government?. What will be the social impact on the people, when each individual will be indulged in the social media like twitter and Facebook?. Today, each individual in the world's popular cities are exposed to the amount of information in a day, which is equal to the information of entire lifetime of our ancestors in 15th-century. The revolution of big data has changed the day today life of human beings. The mobile phones and other network devices plattering as the "on-ramp" for the millions of people to access the information via internet. Not only this, technology has basically influenced the way of interaction of the people with the surrounding. The data collected from the mobile phones, personal computers, tablets, homes appliances, and other multitude devices can be processed and analyzed on large scale, which can influence the social life of the humans. Moreover, it is said that, technology is never biased for any activity, it always remains neutral. For example, hammer can be utilized to build the shelter or murder some people. Thus, the big data has the positive as well as negative impacts on the social human life. In the positive side of the big data, the visualization of the film features, using the contemporary graphics and animations, the smart phones,

and sensors are enabling the human beings to sense and analyze vast information. People are able to have live track to their child's from their phones. Also, the big data analysis is used for tracking the crime, poverty, and political upheaval. The real time epidemics, such as flu trends may be analyzed using the big data analysis, which was quite time taking previously to track. Digitization has revolutionized the healthcare, as mentioned in the story of a daughter and mother, in which the 2 breast cancer gene was tested. The proliferation of the new sensors and the mobile devices used for monitoring the walk, blood pressure, temperature, and food intake has changed the day today life of human beings. The health issues are managed by the big data analysis such as malaria was identified by a non government organization in Kenya. The big data analysis is also being used for the disaster management such as awareness of the flood, how to respond the earthquake, Midwest tornados or Hurricane to the people using social media. Moreover, the big data has guided the daily traveling of human beings using the Maps. Apart from the positive side, there are also some negative impacts of big data on human social behavior. The explosion of social media has reduced the meeting and face to face communication among the humans. The social media is also used to organize the protest, in which sometimes the fake news is also used for provoking the people. E-commerce has changed the buying behavior of human beings. Generally, online reviews of the product or the services are used by the humans to buy the products. However, few companies are flooding the fake reviews to increase their sale, which is misguiding the people. The biggest matter of concern about is all about the security of the big data. In every action, we do left some information about us on the internet. What we search, where we go, what we buy and what we read. Everything of our day to day and social life is being recorded and saved in the internet. The personal privacy, ownership and civil liberties are becoming a matter of concern is today's digital

world, which is creating an ocean of opportunities and challenges in the field of big data.

1.3 Importance of Big Data Analysis

Industries are going alight with the explosion of data. None of the sectors, whether it is medical or engineering have remained away of this momentous change in the last decade. Technology has moved slowly inside each business platform. Thus, it has become part and parcel of every processing unit. Businesses are shifted towards more on innovation rather than stability. Therefore, adopting the big data technologies is boosting the companies to expand their business and remain up to date. Moreover, big data analysis has also allowed the industries and academia to stay updated and predict the future trends [21].

Though big data is in its early stage, but it is producing more revenue for the industries which adopting the new trends. However, the advent of big data systems has allowed the companies to put the untouched data into use and extract meaningful insights from it. Much to everyone's amazement, the data which was left unrecognized or considered useless in the past has suddenly become a goldmine for the companies. The companies can accelerate their processes and thus reduce their operating costs at the end, all thanks to big data analytics [22]. We are currently in a data-driven economy where no organization can survive without analyzing the current and future trends. Whether it is a manufacturing firm or a retail chain, wrangling data has become a crucial job to be done before taking a single step further. In this era of fierce competition, everyone wants to stand out from the crowd. But the question is HOW?. How will the companies be perceived as unique despite having the same operations as others in the industry?. The answer lies in the practices adopted by the firms. In order to perform better than the competitors, the ability to make good and

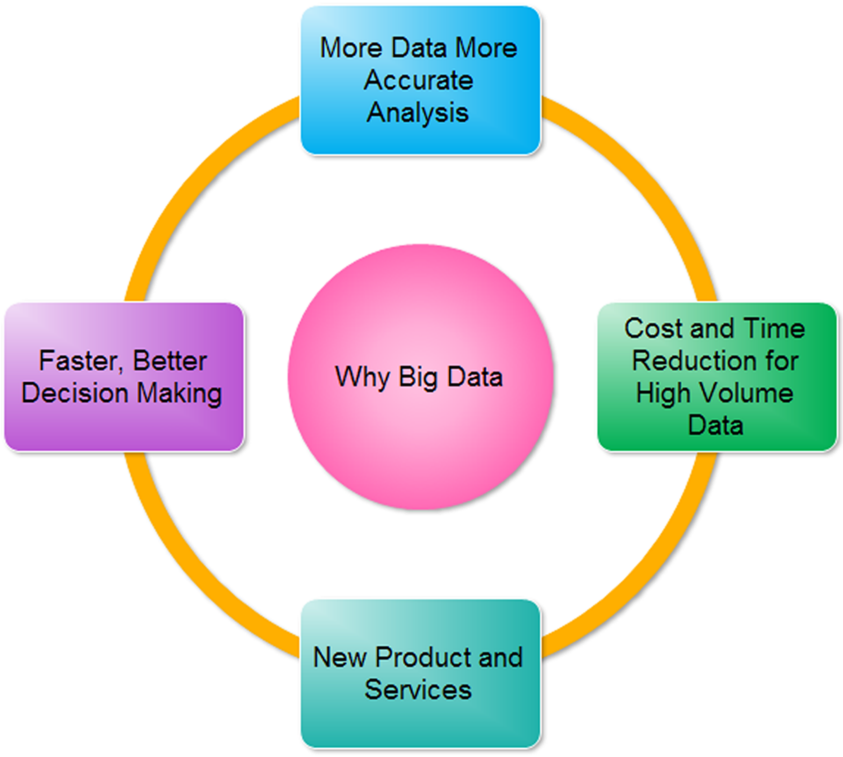


Figure 1.1: Big data importance

intelligent decisions play a pivotal role in every step. The decisions should not only be the good ones, but should be made smartly and as quickly as possible to allow the companies to remain proactive in their approach instead of being reactive.

The practice of implementing big data analytics into the process sheds light on the unstructured data in such a way that helps the managers to analyze their decisions in a systematic manner and take the alternative approach as and when needed. Customer-centricity is the new policy now a days which means that customers have the opportunity to shop anywhere and anytime, it has turned into a challenge for the companies to make every interaction better than the previous one with the help of relevant information. But how will the companies do it on a continuous basis?. The answer is “Big Data”. The customer dy-

namics are ever-changing and so marketers should police their strategies accordingly. The companies can become more responsive by incorporating the past as well as the real-time data to assess the taste and preferences of the customers.

For example, Amazon has grown from a product-based company to a big market player comprising of 152 million customers by leveraging the abilities of powerful big data engines. Amazon aims to delight customers by tracking their buying trend and providing marketers all the related information they need instantly. Moreover, Amazon successfully fulfills the needs of its customers by monitoring 1.5 billion products across the world in real-time and it generates values through leveraging data silos. The companies are getting bigger and hence different processes generate varied data. Many of the important information sulk in the data silos remain inaccessible. However, companies have been able to dig this mountain with the weapon called big data analytics that has let the analysts and engineers drill deep and come out with fresh and informative insights. After this discussion, one thing is for sure that this is just the beginning of a highly digitized and technology-driven era revolving around the powerful real-time big data analytics.

1.4 Research Problem and Goal of Thesis

The meta-heuristic algorithms provide an efficient solution to the data analysis techniques such as data clustering as compared to traditional algorithms such as K-means. However, when the size of the data is large, the metaheuristic algorithms becomes computation intensive in nature. In such environment, the traditional sequential algorithms are not able to provide satisfactory results within the reasonable amount of time. Thus, some parallel model of the meta-heuristic algorithms is required to perform efficient data analysis of the

large-scale data sets. Hence the research goal of the work is extracted as

Developing efficient data analysis methods using meta-heuristic algorithms and introducing their parallel and distributed models which can handle the computational complexity of big data sets. Therefore, to summarize, our research goal can be seen as consisting of the following subgoals:

1. **To, study the various meta-heuristic based computation methods and hence developing parallel meta-heuristic model to handle the complexities of large-scale datasets.** The first research goal is to study the meta-heuristic based methods and the distributed processing tools which can be utilized for the parallelization of the existing meta-heuristic algorithms. Also, to develop a parallel model which can handle the computational complexities of computation intensive problems.

2. **Development and implementation of hybrid methods for Big Data Analysis.** The meta-heuristic based methods may be hybridized with the traditional data analysis methods such as K-means to improve the performance of the existing methods for various big data applications. The objective of this goal is to explore the hybrid model of the metaheuristic algorithms with traditional methods and to investigate the efficiency in the big data environment.

3. **To, develop the novel variant of existing meta-heuristic based method for big data analysis and analyzing its effect on accuracy and computation time.** To, achieve the better accuracy and computation time, some enhancements in the existing methods are inevitable to accelerate the efficiency of the existing methods in the parallel and distributed environment. The proposed work is focused on the data analysis using meta-heuristic based methods. Hence, some variant of the existing

methods should be proposed, which can perform efficient data analysis for the large-scale datasets. Moreover, the efficiency and the computation time of the proposed variants are required to be analyzed and compared with the existing methods.

4. **To study and analyze the applicability of the proposed parallel methods for solving real-world big data analysis problems.** Finally, it is essential to investigate the applicability of the proposed methods. The proposed methods should be tested on some real-word problems in the big data environment. Thus, the fourth goal of the thesis is to validate the proposed methods on some real-world big data analysis problems.

1.5 Overview of the Work Done

This work focused on the development of the methods for the efficient analysis of the big datasets. The proposed work leverages the strength of meta-heuristic based methods for achieving the above stated objectives. Further, the ability of Hadoop MapReduce architecture is utilized for the parallelization of the proposed methods to handle big data sets. Moreover, the proposed methods are leveraged to untangle the real world problems such as twitter sentiment analysis and fake review detection. The following work is carried out to achieve each objective.

1. For achieving the first objective, Hadoop MapReduce architecture was studied. Further, a Hadoop cluster of 10 computers was designed. Moreover, a parallel method based on bat algorithm has been designed using the Hadoop and MapReduce for clustering large scale datasets. The proposed method was compared with the particle swarm optimization and the results were validated on 5 benchmark datasets. The ex-

perimental results demonstrated that the proposed method is well suited for the data analysis of the large-scale dataset.

2. In continuation of the above work to achieve the second objective, two hybrid methods have been developed namely, DFBPKBA and MR-KBBO. In the the first method, three contributions were incorporated in the bat algorithm to improve its efficiency. First, a novel variant of the bat algorithm was introduced, second the proposed variant was hybridized with K-means algorithm to speedup the convergence, third the hybrid algorithm was parallelized using the Hadoop MapReduce to handle the complexity of large-scale datasets. Furthermore, the clustering effect of the proposed method was first tested on 5 benchmark datasets and results were compared with K-means, particle swarm optimization and bat algorithm. Moreover, the speedup of the proposed method was tested by varying number of nodes on each run. Further, a novel hybrid method using biogeography based optimization and K-means has been introduced. The performance of the proposed method has been compared with the DFBPKBA and three other state-of-the-art methods.

3. To achieve the third objective, a novel variant grey wolf optimizer(GWO) named enhanced grey wolf optimizer was developed. The grey wolf optimizer has been recently introduced and widely used by the researchers. However, sometimes it falls into local optima due to lack of population diversity. Thus, a good attempt can be made to improve the abilities of the GWO and leverage its strength. Therefore, in this work, the exploration and exploitation ability of the GWO was improved by incorporating levy flights and binomial crossover. The contribution of this work has three folds: First, the clustering accuracy of the GWO was enhanced by improving the exploration and exploitation ability of the GWO. Second, the clustering accuracy of

the proposed method was validated on seven benchmark datasets. Third, the parallel model of the proposed method was designed to handle the big datasets. Furthermore, the applicability of the proposed method was tested on two large-scale datasets and compared with four other MapReduce based state-of-the-art.

4. Furthermore, a novel meta-heuristic algorithm based on military dogs is developed. The proposed algorithm mimics the searching capability of the trained military dogs. Military dogs have strong smell senses by which they are able to search the suspicious objects like bombs, wildlife scats, currency, or blood as well as they can communicate with each other by their barking. The proposed algorithm has been benchmarked on 17 test functions and compared with PSO, MVO, GA, and PBIL. The results show that the MDBO is outperforming other algorithms on the majority of test functions in terms of convergence, mean fitness value and standard deviation. The consistency of the results has also been statistically validated by the BoxPlots. Moreover, a parallel model of the proposed algorithm has been developed to handle large scale dtasets. Further, the performance of all the proposed methods has been compared in terms of accuracy and computation time.

5. Nowadays, tweets are widely used for analyzing the sentiments of the users, and utilized for decision-making purposes. Though clustering and classification methods are used for the twitter sentiment analysis, meta-heuristic based clustering methods have witnessed better performance due to the subjective nature of tweets. However, sequential metaheuristic based clustering methods are computation intensive for large-scale datasets. Therefore, to analyses the performance of proposed methods on some real-world problems, the problem of twitter sentiment analysis has been unfolded. Each method has been tested on four benchmark datasets and the results are com-

pared in terms of accuracy and and computation time.

6. Online reviews are increasingly used by the customers for taking the decision to purchase a product or service. E-commerce sites provide a good platform to the customers to express their experience of the product or service. However, to increase profit or publicity few companies hire spammers to produce synthetic reviews to promote their product or demote their rivals brand. Over the time, these fake reviews have seen to be grown in the market due to increasing competition. Thus, fake review detection is an open and challenging problem. However, the literature witnessed majority of the research using sequential algorithms which are not able to provide the satisfactory results for the big datasets. This thesis has made a successful attempt to solve the fake review detection problem by leveraging the strengths of the proposed methods. All the proposed methods have been tested on one benchmark and one real world dataset and the performance has been compared in terms of accuracy and computation time.

1.6 Contribution of the Research Work

1. A parallel bat algorithm based data clustering method is introduced for big data analysis.

2. A novel hybrid variant of bat algorithm for large-scale data clustering named dynamic frequency based parallel K-bat algorithm has been introduced to overcome the limitations of bat algorithm.

3. An enhanced grey wolf optimizer has been developed for the efficient data clustering

of large-scale datasets.

4. A new optimization algorithm named Military Dog Based Optimizer is introduced for solving the real-world optimization problems pertaining big datasets.

5. The problem of Twitter sentiment analysis for large-scale datasets has been unfolded using the proposed methods.

6. The fake review detection problem has been solved in the big data environment using the proposed methods.

1.7 Organization of the Thesis

The rest of the thesis is organized as follows:

Chapter 2 presents the background study in the field of big data analysis. The chapter starts with the evolution of big data and its social impact on today's human life. The key challenges involved in the big data analysis along with its applications are discussed. Further, a brief introduction of metaheuristic algorithms and their applications in the field of data analysis are also detailed. Then, the recent state-of-the-art review of various methods available for handling big data are explained.

Chapter 3 reveals the proposed parallel methods for the big data analysis. The chapter presents a dynamic frequency based parallel K-bat algorithm for the big data analysis. Further, an enhanced grey wolf optimizer is presented, which leverages the strengths of metaheuristic approach and MapReduce architecture. Also, a hybrid method, MR-KBBO is introduced which leverages the strengths of evolutionary approach for the big data analysis. The techniques used for validating the proposed methods are also explained. The

performance measures along with the statistical tests used to evaluate the performance of proposed methods are summarized.

Chapter 4 presents a novel metaheuristic algorithm named, "Military dog based optimizer". The proposed algorithm is validated on the 17 recent benchmark functions. Furthermore, the proposed algorithm has been investigated to solve the clustering problems and the results are validated 8 benchmark datasets. Moreover, a parallel version of the proposed algorithm is introduced for the big data clustering.

Chapter 5 presents the applications of the proposed methods for the solving the real world problems. Two real-world problems have been unfolded namely, twitter sentiment analysis and fake review detection. The chapter begins with the elaboration of the twitter sentiment problem and proposed approach for mining the sentiments from large scale datasets. Thereafter, the accuracy in the results of each method has been presented and the performance is compared on the basis of accuracy and computation time. Further, the proposed methods have been applied to unveil the fake review detection problem and the performance is compared.

Chapter 6 summarizes the results obtained from the previous chapters of the thesis. Also, the thesis is concluded and the future research possibilities and scope is discussed.

References This section details the references used in the thesis.

Chapter 2

Literature Survey

This chapter presents a systematic review of the work done in the area of data analysis using metaheuristic algorithms and big data. The chapter starts with the origin of big data and ends with its importance and state-of-the-art for the analysis of the big data. It also presents the work done in the field of clustering using the metaheuristic algorithms and the methods for the parallel processing to handle the big datasets.

2.1 Big Data Definitions

The 'Big Data' word was first used by John Mashey in 1998 when he was delivering a presentation on Silicon Graphics (SGI) slide titled "Big Data and the Next Wave of InfraStress" [22]. In 2001, Doug Laney, a reputed analyst elucidated the opportunities and challenges of the big data through a 3Vs model, i.e., Volume, Velocity, and Variety [23]. In the last one decade, the enterprises like IBM [24], Microsoft [25] also used the "3Vs" model to explain big data [26].

Apache foundation, in 2010 explained big data as datasets that can not be managed

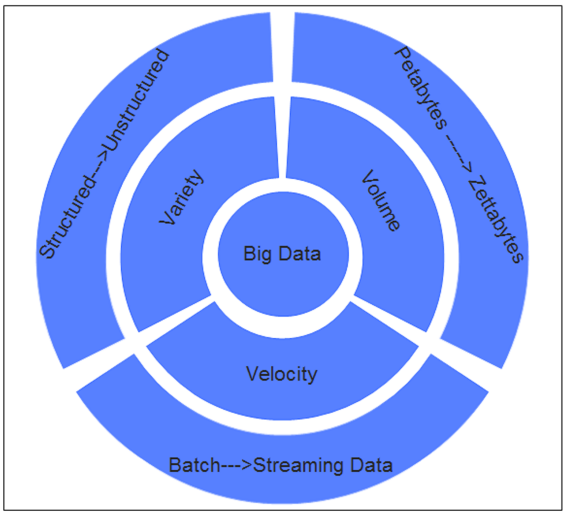


Figure 2.1: Big data Characterization

and processed by commodity computers within tolerable time. In continuation, McKinsey, which is a global consulting firm declared the ‘ ‘Big Data as the next frontier of productivity, innovation, and competition”. Big data means the datasets which could not be handled by classical database systems and tools. This definition has two implications: First, the volumes of the datasets are changing, and growing with the advancements in the technology. In today’s scenario, the data range of big data varies from TB to PB [21]. As per the definition of McKinsey, the volume could not be the only criterion of a data set to be called as big data.

The visual representation of the big data is shown in Fig. 2.1. However, IDC which is one of the leading organization of the world, in 2011, called big data as new generation

architecture, designed to extract value from huge volume and variety of data with the high velocity capture in an economic way [27]. This definition has given one more characteristic of big data i.e., Value. The 4Vs definition of big data has also got good recognition, as it describes the importance of the big data. The 4Vs definition elucidates an essential issue of the big data, which is how to extract the required information from such huge and heterogeneous data. Moreover, NIST illustrated that big data mean the data for which high volume or acquisition speed limits the use of traditional methods for efficient analysis. In other words, the data which can only be effectually analyzed with the help of latest technologies. Thus, it can be elucidated, that there is a need to develop efficient methods and technologies for the efficient processing of the big data. In 2012, the 8V model of the big data has been introduced namely, volume, variety, velocity, value, validity, vulnerability, volatility and visualization [19]. In today's scenario, the big data has gained its recognition, but there are still different standpoints for its definition.

2.2 Evolution of Big Data: 1970 to Current Scenario

The concept of a technology used for storage and analysis of data, called "database machine" was appeared in late 1970. With the time and growth of data size, the processing capacity of the mainframe system became insufficient. Further, in the 1980s, a parallel database system called, "share nothing," was introduced to manage the growing data [28]. The architecture of this system was cluster computing, where each computer has its storage, disk, and processor. In the same time, Teradata has launched its first parallel database system that has gained good popularity. In early 1986, Teradata introduced its first parallel database with 1TB storage capacity. It was used by K-mart, a retail company of America

to enlarge its data warehouse [29]. The database world has started recognizing the benefits of parallel database systems in the late 1990s. However, with increasing demand and value, several challenges came in picture specially for search engine companies. Thereafter, Google introduced GFS [30] and a parallel programming model was introduced named MapReduce. Moreover, users content, satellite data, sensors, and social media also given an equal contribution to the growth of the digital data. Jim Gray, a pioneer of database software, called this transformation as "The Fourth Paradigm" in the year 2007 [31]. He suggested the generation of new computing tools is the only solution to manage and analyze such high volumes of data. In 2011, IDC published a report "Extracting Values from Chaos" [27], to introduce the potentials and importance of big data, which gained popularity in the industry an well as academia. In the recent years, the majority of the MNCs like Google, Microsoft, Amazon, Facebook, and IBM are working actively in the field of big data.

Since 2005, IBM has spended 16 billion USD in the projects pertaning big data. In academia also, big data is current research hot-spot. Nature released a special issue in 2008 and Science released a special issue for the big data named key technologies of "data processing". ERCIM also launched an issue related to big data. In 2012, a report named big data, Forum of Switzerland, announced big data as an asset, like gold or money in the report titled Big Data, Big Impact. In 2012, the USA government announced 200 million USD investment for the big data research. In 2012, the UN used big data for development report, which indicates the importance of the big data. The detailed year wise market volume of big data in US dollars is presented in Figure 2.2.

Further, as per the report from IDC, the overall volume of the data across the glob was spotted as 1.8ZB, in 2011, which increased approximately 9 times in 2016 [27]. This figure is expected to be 2 times in every couple of years. The term big data is generally used

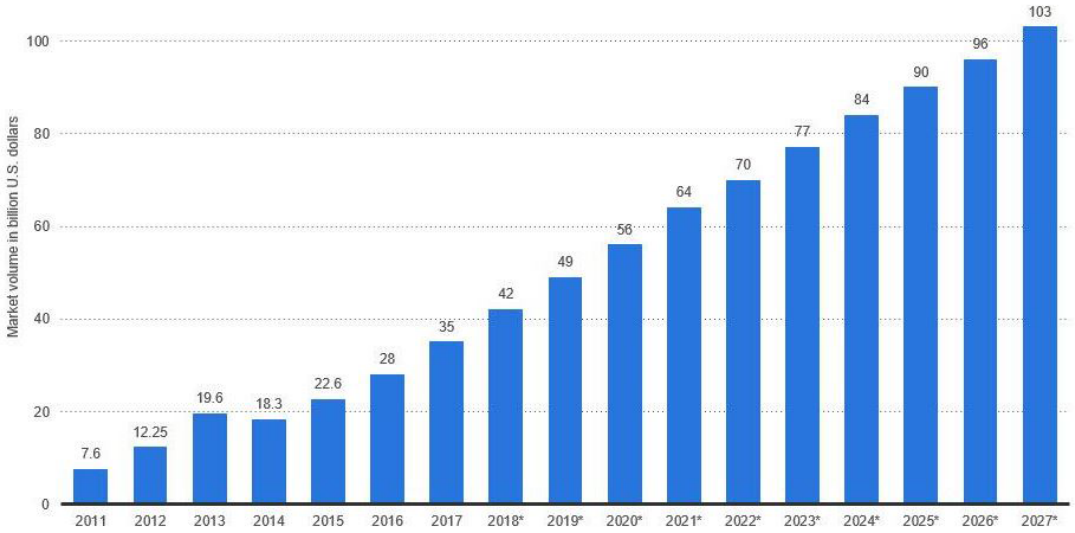


Figure 2.2: Market Volume of Big Data in US Dollars

to explain such enormous growth of datasets. Big datasets include raw and unstructured data, which need much reliable and robust real-time analysis. Moreover, big data has also brought new challenges and opportunities for efficiently organizing and managing such datasets. Recently, a number of industries and government agencies are planning to accelerate investments for the big data. The issues related to big data have acquired much space in the in public media, such as New York Times [32], the Economist [33]. Nature and Science journal also discussed the opportunities and impacts of big data in the upcoming generations [34].

Further, the data of the online service providing companies is growing very rapidly. Google stores an estimated 10 exabytes and it processes around 30 thousand queries per second, Facebook generates approximately 10 petabytes of data in a month, Taobao, a Chinese company, processes tens of terabyte data per day for online trading. An average of 72 hours videos is uploaded per minute to YouTube [20]. Thus, considering the complexity,

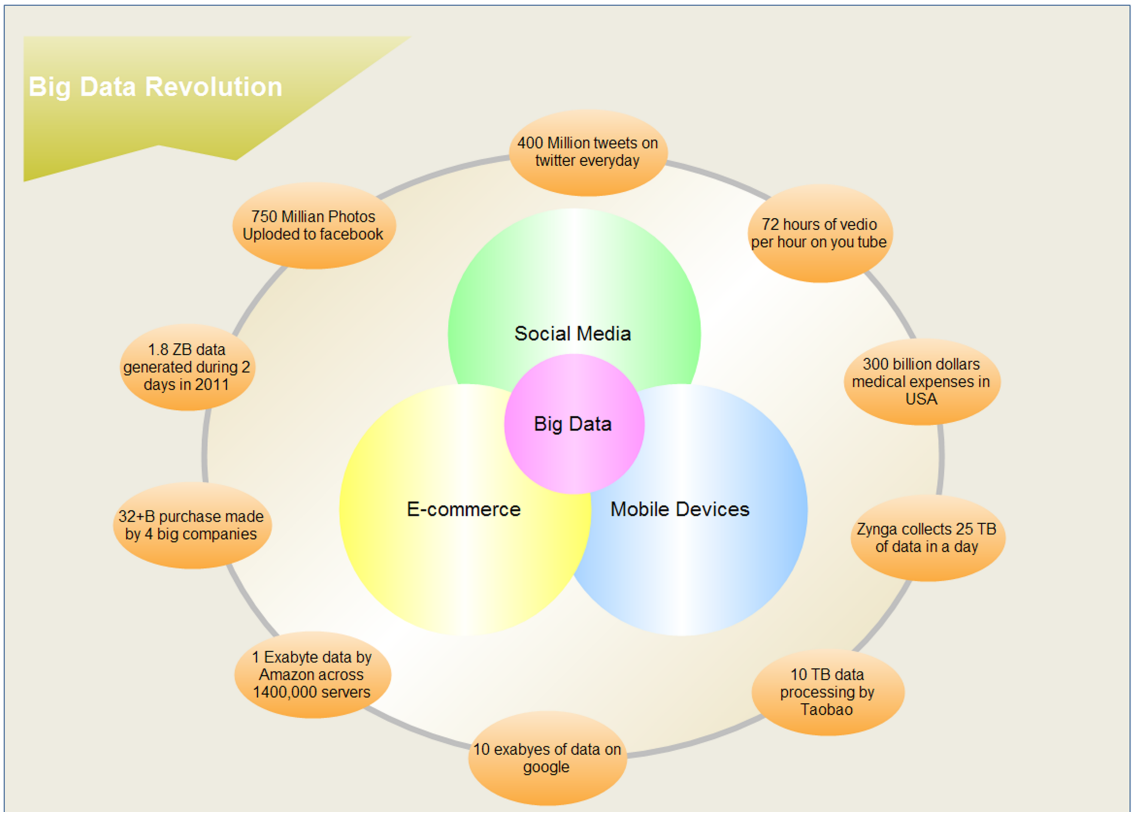


Figure 2.3: Big Data Revolution

heterogeneity, robustness, and security of big data, efficient mining methods are required for the for better decision making. The revolution of big data in terms of volume has been illustrated in Figure 2.3.

2.3 Big Data Challenges

The rapid growth of the digital data brought new challenges for the management and analysis. The conventional data processing systems such as RDBMS can handle only structured datasets. Apart from this, RDBMS based systems require reliable hardware and also they are not able to handle the computational complexity of large-scale datasets. Though cloud

computing may handle the infrastructural need of the big data, but for the analysis of such datasets, NoSQL [35] databases and distributed file systems [36] are required. In the recent years, these frameworks have achieved good popularity for handling the big datasets. However, there are some hurdles and challenges in the development of big data-based applications, as mentioned by Min Chen et al. [22], which are described as follows.

- **Data representation:** Data representation makes data more clear and meaningful for the user interpretation and analysis purpose. However, if the representation method is not efficient, some information may be loosed from the original data, which may lead to inefficient data analysis. Since big data sets are generally unstructured in nature. Thus, careful data representation is required.

- **Redundancy reduction and data compression:** Big data sets are high volume data, which may have data redundancy. Data redundancy may increase the processing cost and hence it should be removed without losing any important information.

- **Analytical mechanism:** The big data is heterogeneous in nature and it processes masses of high volume data. In contrary, RDBMS design suffers from scalability and expand-ability issues, which can not handle big data requirements. No SQL databases may process semi-structured and unstructured data. However, No SQL databases also suffer from some performance issues. Thus, more novel methods and techniques are required for the efficient analysis of the large-scale datasets.

- **Data confidentiality:** In the present, most of the big data owners or service providers are dependent on tools to analyze big datasets, which involves security and confidentiality risks. Therefore, sufficient preventive measures must be adopted to protect and ensure the security of the datasets.

- **Energy management:** Mainframe systems generally consume more energy. Thus, handling huge volume datasets will be expensive in terms of electric energy consumption. Therefore, energy efficient systems must be developed and managed to process huge masses of data.

- **Scalability:** This is a prime concern of the traditional database and analytical systems. The big data-based framework must support scalability to manage the increasing demand for data size.

- **Cooperation:** Big data analysis is an interdisciplinary research, which invites experts from heterogeneous fields to get maximum advantage of big data. Therefore, cooperation of the researchers and scientists of different filed is required to develop socially useful applications.

2.4 Domains of Big Data Analysis

Big data analysis is powerful in the today's world for the industries and academia. It is widely used for the management decision making in the different domains. However, big data analysis suffers from many challenges and complexities. There are several data analysis domains such as text analysis, multimedia analysis, mobile analysis, structured data analysis, network data analysis, and web data analysis. A brief discussion of the fields is presented in this section.

- **Text Data Analysis:** The text is the most popular data format used to store the data. Text data is stored in a number of areas such as business document data, emails, social media, users opinion, and comments. Thus, text analysis is a prominant widely used

technique, especially used for the business and health-based decision making. The text processing is a way of extracting information from the unstructured raw data. NLP is a key element of text mining. NLP is used to analyze the text data, which is used in combination with machine learning, data mining, and statics to extract knowledge. Parts of speech tagging, lexicon analysis, and word polarity extraction are some of the common tasks which are performed in the NLP [37]. The most common applications of the text mining are information extraction, clustering, and classification of text-based data, product review classification, and opinion mining.

- **Structured Data Analysis:**

  Structured data is generally generated by scientific research, application and business analytics. This type of data analysis is mostly performed by commercialized technologies such as OLAP, and RDBMS. However, when the data size is large, these traditional methods are not able to perform in a reasonable amount of time. In such scenarios, big data tools are required to perform efficient data analysis of the structured data. The protection and analysis of e-commerce, e-government, and health-related data may be structured and also large in volume. At present, to perform time efficient analysis of these data sets, big data analysis is actively used.

- **Web Data Analysis:**

  Web data analysis is used to extract and evaluate meaningful information from the web-based documents. It broadly involves information retrieval, text mining, and natural language processing. Web mining is basically of three types namely, Web content mining, Web usage mining and web structure mining [38]. Generally, web content mining deals with the extraction of information from the text, audio, image,

video, and hyperlink etc., available on the web pages. The research on the web data manly falls into the category of big data analysis since web related data is unstructured and generally hug in volume. Moreover, web structure mining basically deals with the discovering of web link structures. The topological structures of the hyperlinks are studied to develop models, which are used to classify web pages. Web usage mining deals with access log data. Browsers' history records, user queries, user profiles, time spend by the user on the website and click rate etc., are used to extract knowledge from the web-based data.

- **Multimedia data analysis**

  Multimedia data is gaining much popularity in the recent years. It is defined as the process of extracting meaningful information from the images, audio, and videos related data. Multimedia data contains plenty of useful information as compared to the simple text or structured data. The multimedia analysis includes multimedia event detection, multimedia summarization, video classification, multimedia annotation, multimedia recommendation, etc. For example, representation of video content sequence, audio summarization by extracting the important keywords are the important examples of the multimedia analysis which is used in many business application. The multimedia recommendation is an active and important field of the multimedia analysis, in which multimedia content is analyzed and recommended to the users as per their preferences.

- **Network Data Analysis:**

  Network data analysis is generally associated with sociological network based data [38]. Moreover, social network analysis is the most preferred area of the researchers

of the 21st century. Social media such as Instagram, Twitter, Snapchat, LinkedIn, and Facebook etc. are providing a fertile platform for the researchers. These platforms generate massive unstructured data, which contains valuable information hidden inside [39]. Social media is also influencing the daily life and behavior of the individuals. This change is dependent on the relation between the individuals, time effect and type of networks. Also, social media analysis have an influence on the e-commerce and marketing. Many companies are actively using network analysis for marketing, recommendation, and election advertisement [40]. Network related data is mostly unstructured, noisy and massive, thus big data analysis is performed to extract information from the network related data.

- **Mobile Data Analysis:** In the recent years, mobile devices have given a new platform for the big data analysis. By march 2018, more than 3.3 million applications were available on google play store and this figure was 1 million in July 2013 [41]. However, the data generated from the mobile devices have noise, redundancy, and massiveness. Mobiles are nowadays used for forming new communities, real-time monitoring of peoples health and google map etc., which provides a huge platform for the researchers working in the area of big data and data analytics. Delano et al. [20] introduced an application named iTren, which is used for the detection and monitoring of Parkinson disease patients. Furthermore, Norway university researchers developed the smartphone-based application for the safety of the system by monitoring the paces of the peoples walk [20].

2.5 Key Applications of Big Data Analysis

Big data analysis provides proficient values and support in the decision making of different fields. The big data-driven applications have made a revolution in the last decade. In the, early of 20K, business intelligence based on the big data analysis has prevailed in the various industries. In this section, some crucial and influential fields of big data applications are presented as follows.

- **Evolution of Commercial Applications:** In the earlier time, mostly the data related to business was structured, collected by companies and analyzed using RDBMS. In early 1990, simple analytical techniques such as predictive modeling, OLTP and search based business intelligence were used [42]. In 2011, the number of mobile phones and tablets became more than the personal computers [22]. Thereafter, in the early of 21st century, with the revolution of social media and e-commerce, new opportunities arrived because of the direct interaction of customers with the industries. Industries started mining the data of users search behavior, click-stream and product reviews by the customers. Social network analysis, sentiment mining and market analysis are the good weapons for the companies to spread their business by text and web mining.

- **Evolution of Scientific Applications:**

  Scientific research involves the data generated from the satellite, sensors and other instruments such as genomics astrophysics and oceanology. The data generated from such devices is complex and massive. Further, the national science foundation, a scientific firm of USA, proclaimed the big data program to enhance the efforts for min-

ing the complex data sets. Other, scientific disciplines such as iPlant have invested in network infrastructure, virtual machine, and software to provide researchers with a proper platform for big data analysis.

- **Evolution of Network Applications:** In the today's generation, the network-based application has prevailed in the WWW. The majority of the data in the universe belongs to the network data, which contains audio, videos, text and images. In contrary, the older generation of the Internet was moving around the mails and web-based services. The text and webpage mining was widely applied to the emails and search engine. Social media has provided a good platform to create, upload and share contents. Therefore, for current demands, novel methods and technologies are required to handle unstructured and semi-structured massive data sets. The network related data is prime for the today's business world, such as multimedia data can be used in military applications, advertisements and in the recommendation.

2.6 Meta-heuristic Algorithms

Over the last three decades, more than sixty meta-heuristic algorithms have been proposed by the various authors. Such algorithms are inspired from physical phenomena, animal behavior or evolutionary concepts. These algorithms have been widely used for solving the various real-world optimization problems. Researchers are continuously working to improve the existing algorithms and also proposing new algorithms that are giving competitive results as compared to the existing algorithms present in the literature.

Nature-inspired meta-heuristic algorithms mimic the optimization behavior of nature. Generally, these algorithms are population-based and start with a population of random solu-

tions to obtain the global best solution. In contrast to this, there exists single-solution based algorithms like hill climbing [43] and simulated annealing [44], which initiates the optimization process with a single solution. However, these algorithms suffer from the problem of the local trap and premature convergence as they do not share any kind of information. On the contrary, population-based algorithms improve the solution over the iterations by information sharing.

Two common aspects of the population-based algorithms are exploration and exploitation. Exploration represents the diversification in the search space, while exploitation corresponds to the intensification of the current solution. All population-based algorithm tries to attain an equilibrium between exploration and exploitation to achieve the global best solution. Every agent of the meta-heuristic tries to improve its performance by sharing its fitness value with other agents at each iteration. The meta-heuristic can be broadly classified into three categories namely, swarm-behavior based and evolutionary-based and other several phenomenon such as human behavior based and physics based. Figure 2.4 depicts the broad classification of metaheuristic algorithms.

Swarm-based algorithms behave like the swarm of agents such as fishes or birds to achieve optimization results. Eberhart et al. [45] proposed the particle swarm optimization (PSO) which was inspired by the swarming behavior of fish or birds in search of food. Gandomi [46] presented an algorithm based on the simulation of the krill individuals. Further, Wang et al. [47] proposed the hybrid krill heard algorithm to overcome the problem of poor exploitation capability of the krill herd algorithm. Ant colony optimization is another swarm-based algorithm, which imitates the path finding behavior of ants [48].

Bansal et al. [49] introduced and optimization which mimics the foraging behavior of spider monkeys. The proposed algorithm was found to be competitive with PSO, DE, ABC,

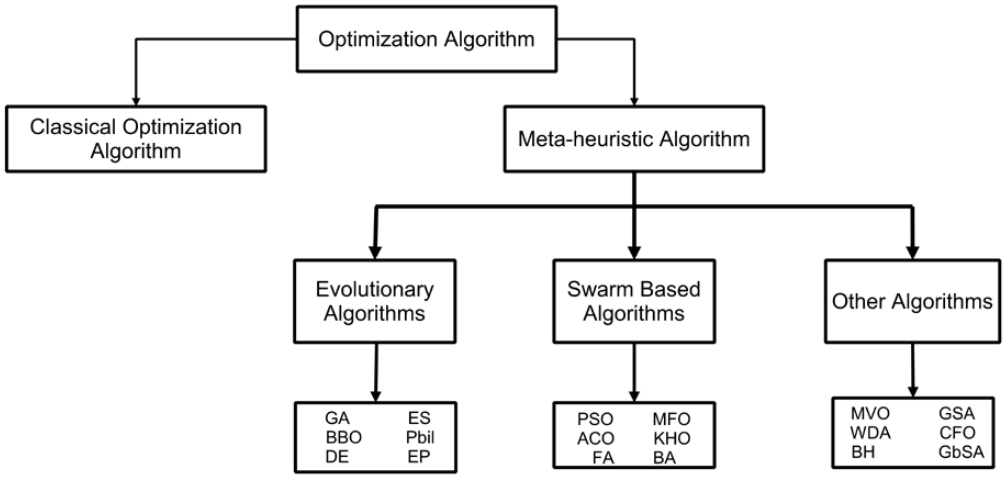


Figure 2.4: Classification of Meta-heuristic Algorithms

and CMA-ES in terms of reliability, accuracy and efficiency.

Tsai et al. [50] presented a variant of the bat algorithm to improve exploration and exploitation. The proposed approach was tested on three benchmark functions with ten, thirty, fifty and hundred dimensions and it was concluded that new variant is able to enhance up to forty-seven percent accuracy of the original bat algorithm.

Kavita et al. [51] developed a fitness based particle swarm optimization to improve the slow convergence rate and stagnation in the local optima of the PSO by incorporating a new position update operator. The new phase was inspired from the onlooker phase of the Artificial Bee Colony algorithm. The proposed algorithm has outperformed particle swarm optimization and artificial bee colony optimization in terms of mean fitness and convergence rate. Jadon et al. [52] improved the problem of poor exploitation and slow convergence rate of ABC. The proposed algorithm outperformed other state-of-the-art methods on 24

benchmark functions.

Evolution based algorithms are inspired by the biological evolution phenomena such as Darvin evolutionary theory. The evolutionary algorithms work on the principle of generating better individuals with the course of iterations by combining best individuals of the current generation. The popular genetic algorithm (GA) is an evolutionary algorithm based on the evolution of natural species. It maintains the balance between exploration and exploitation through the mutation and crossover operators. Another biological process based evolutionary algorithm is ES which gives almost equal importance to recombination and mutation, and it uses more than two parents to accord to an offspring. Baluja [53] proposed the probability-based incremental learning algorithm (PBIL) which manages only statics of the population rather than managing the complete population. Simon presented bio-geography based optimizer which is based on the immigration and emigration of the species between the islands of natural bio-geography. Differential evolution is another popular evolutionary algorithmic introduced by Storm et al. [54]. The physics-based algorithm optimizes the problem by imitating the physics-based phenomenon. Gravitational search algorithm, proposed by Rashedi et al. [55], is one such algorithm which is based on Newtonian laws of gravity and motion. Hosseini [56] proposed an intelligent water drop algorithm which was inspired by the flow of rivers, as rivers often follow the shortest path while flowing from source to destination. Further, Birbil [57] proposed an algorithm based on the concept of electromagnetism in which the properties of attraction and repulsion is used to attain a balanced trade-off between exploration and exploitation. Moreover, Mirjalili et al. [58] proposed multi-verse optimizer (MVO) in 2015, which is based on the notion of cosmology i.e white hole, black hole, and wormhole. Some other physics-based algorithms are Galaxy-based Search Algorithm (GbSA) [59], Black Hole (BH) [60] algo-

rithm, Small-World Optimization Algorithm (SWOA) [61], Ray Optimization (RO) [62], Curved Space Optimization (CSO) [63].

Further, some of the popular meta-heuristic algorithms used in the study are explained as follows.

2.6.1 Grey Wolf Optimizer

Grey wolf optimizer (GWO) is a meta-heuristic algorithm proposed by Mirjalili et al. [64], which imitate the hunting mechanism of the grey wolves. In GWO, grey wolves are grouped into $alpha(\alpha)$, $beta(\beta)$, $delta(\delta)$ and $omega(\omega)$ according to their social hierarchy. The best three grey wolves are considered as $alpha$, $beta$, $delta$ and renaming grey wolves are termed as $omega$. The $alpha$ wolves are the commanding one and all other grey wolves follow their instructions. The second category of the wolves belonging to the $Beta$ category are responsible for helping $alpha$ in their decision making. $Omega$ are the lowest ranked grey wolves. In the grey wolf algorithm, hunting is escorted by $alpha$, $beta$ and $delta$ while $omega$ wolves are responsible for encircling the prey to find better solution.

The GWO algorithm has been widely used in a number of applications in last three years. Emary et al. [65] used the binary version of GWO to perform optimal feature selection. Komaki and Kayvanfar [66] optimized the flow shop scheduling tasks by applying GWO. Moreover, GWO has also been applied for solving the power dispatch and mixed heat task for power systems [67]. Medjahed et al. [68] introduced a GWO-based method for selection of hyperspectral bands. Fergany and Hasanien [69], demonstrated the efficiency of GWO on optimal power flow (OPF) problem. On the same footnote, GWO was drafted to design the wide area power system stabilizer (WAPSS) [70]. To solve the economic dispatch problem, Jayabarathi et al. [71] introduced mutation and crossover mechanisms in GWO. Guha [72]

employed GWO in power systems for optimizing the load frequency control (LFC). Song et al. [73] utilized the strengths of GWO to find the optimal parameters of the surface waves. GWO optimizer is also used effectively for training the multi-layer perceptrons [74]. Amirsadri et al. [75] proposed a new variant of gwo using lévy flights in combination with back propagation for training the neural network.

Though GWO has gained quick popularity since it has been proposed. It has surpassed the other popular meta-heuristics, PSO, MVO, DE, GA on the benchmark functions and other standard problem [64]. **Despite of wide applicability, GWO has limitation of lack of population diversity. This results in slow convergence rate and risk of trapping into local optima [11].**

2.6.2 Bio-geography Based Optimization

Biogeography based optimization(BBO) is a meta-heuristic algorithm inspired by the theory of island bio-geography and proposed by Dan Simon [76] in 2008. The proposed algorithm is based on the distribution and equilibrium of species in different islands. The species move between various Island in the search of better one. In BBO, the quality of Island is mathematically represented by the habitat suitability index (HSI) which depends on the various factors called suitability index variables (SIVs). The Islands with high HSI value share the information of their features with low HSI Islands and the species migrate from low HSI island to high HSI island to attain equilibrium.

BBO had been widely used for the optimization purpose in various engineering domains such as power optimization [77], image classification[78], job scheduling [79] and data analysis [80]. Moreover, several variants of the BBO has been proposed by the researchers to achieve the best results in the specific optimization problem.

Gong et al. [81] introduced real coded BBO by incorporating new mutation operation. Ma et al. [82] modified the migration process of the BBO and presented blended BBO for the constrained optimization. In the same footprints, Li et al. [83] introduced, perturb bio-geography based optimization (PBBO) by incorporating perturbation based migration. Moreover, a multi-operator based BBO (MOBBO) was introduced by Li [84], by introducing multi-parent based migration. An accelerated BBO was introduced by Lohokare et al. [85] by improving the mutation operator of the BBO. Also, the local search ability of the BBO was enhanced by integrating the locality search operator.

Feng et al. [86] proposed orthogonal crossover operator based BBO to magnify the exploration ability and population diversity. Likewise, Xiong et al. [87] introduced polyphyletic operator based migration to avoid local optima. Feng et al. [88] revamped the mutation and migration operator to balance the exploration and exploitation of the BBO. Moreover, few researchers also proposed the hybrid BBO with other evolutionary based algorithms such as DE and EA. Gong et al. [89] introduced a novel variant of the BBO named DE/BBO. The exploration ability of the DE was combined with the exploitation of BBO. A novel, two stage BBO was presented by Boussaïdet al. [90], which update the population alternatively using DE and BBO. BBO has been proved to be an efficient evolutionary algorithm for solving various optimization problem. **However, due to single feature migration property BBO shows poor exploration capability and sometime stuck into local optima.**

2.6.3 Bat Algorithm

Bat Algorithm (BA) is a popular meta-heuristic, introduced by Yang in 2010. The basic inspiration behind the bat algorithm is the echolocation behavior of the micro-bats. Mi-

crobats have the ability to find their prey even in the darkness by using their echolocation behavior. The bat algorithm has witnessed its promising performance for various optimization problems. The strength of bat algorithm has been leveraged in the various domains such as electric power optimization [91], energy management systems [92], image processing [93], data classifications [94], job scheduling [95]. Gandomi et al. [96] improved the global search ability of the BA by incorporating chaos. Bahman et al. [97] introduced a novel self-adoptive approach by improving the velocity updation strategy of the bats. The author successfully utilized the proposed algorithm for optimizing the battery storage capacity micro-grid operation management. Jaddi et al. [98] enhanced the diversity in the bats by introducing new topologies, which made better consensus among individuals. The proposed variant has shown efficient for optimization of the artificial neural network model. On the same footprints, Khan et al. [99] introduced a novel a novel variant BA by leveraging the strengths of simulated annealing (SA), harmony search (HS) and PSO. The proposed algorithm has outperformed GA and PSO for training the feed-forward neural network model, which is used in the e-learning context. Moreover, in 2013, Wang et al. [100] hybridized BA with HS by incorporating a mutation operator to solve the numerical optimization problem. In continuation, He et al. [101] proposed a novel variant of BA by introducing Gaussian perturbations and SA. In 2014, Sadeghi et al. [102] improved the exploitation ability of the BA by introducing local search ability of the particle swarm optimization. The proposed algorithm was utilized for optimizing the bi-objective inventory model of supply chain management. Further, Yilmaz et al. [103] enhanced the exploitation ability of the BA by incorporating Invasive Weed Optimization. Xie et al. [104] introduced a novel variant of BA based on Levy flights and differential operator. All the proposed variants had been proved efficient on their proposed optimization problem. **However, BA**

lacks the self-adaptive ability of bats with the environment. Studies witnessed that bats are sometimes influenced by other insects, and results in the deviation from the target and fall into local optima.

2.7 Data Clustering

Clustering is the prominent approach of unsupervised learning and considered as part and parcel of data engineering applications, such as image segmentation, data mining, information retrieval system, anomaly detection, medicine, computer vision, and construction management [105, 106, 107]. Over the past years, several algorithms for data clustering have been introduced in the literature to handle the diversity in data and different sets of application requirements. Data clustering is defined as the process grouping up of data elements, which have a maximum resemblance. The aim is to develop a method, which classifies an unlabeled data elements into homogeneous groups. Data clustering is broadly classified into three categories: hierarchical, overlapping and partitional. The hierarchical based clustering produces a tree type structure, known as dendrogram plot, representing the nested type groups [108][109]. The prior knowledge about the number of clusters is not required in this type of clustering [110].

The overlapping based clustering algorithms are generally fuzzy based [111][112]. The fuzzy c-means (FCM) [113] and fuzzy c-shells are the most popular algorithm (FCS) [114] as present in the literature. In the fuzzy-based clustering, each data element is a part of all the clusters with some members. The partition-based clustering is defined as the process of dividing N data objects into K clusters with the aim of minimizing the distance of each data object from its cluster centroid. The distance of each data object from its centroid can

be calculated by using a distance metric such as Euclidean distance [115] or cosine measure [116]. The partition-based methods are well liked in the different research areas, such as image segmentation [117], robotics [118], wireless sensor network [119], web mining [120], business management [121] and medical sciences [122]. For each problem, the distribution of data elements is different and complex. Thus, a single method of clustering cannot be fit for all types of problems. Therefore, based on the nature of the problem and dataset, the user has to select the suitable clustering technique. This work is also based on the partitional clustering, hence a systematic review of the partitional clustering is provided in the following section. Though the work is mainly focused on the meta-heuristic based clustering, K-means algorithm is also included, which is the most popular and oldest partition based algorithm.

2.7.1 50 years: beyond K-means

The K-means was introduced by Lloyd in the Bell laboratories in 1957 [123]. The K-means algorithm is still widely used, because of lower computational cost and easy implementation. Moreover, the main motivation of the K-means algorithm is to partition the data in such a way, that minimizes the squared error between data objects and their centroid. After the success of K-means, other partition based clustering algorithms were introduced by researchers such as bisecting K-means [124], sort-means [125], K-harmonic means [126], K-modes algorithm [127], and Kernel K-means[128]. All these algorithms are, computationally economic in nature and simple. However, these algorithms follow hill-climbing strategy and hence usually trap into local optima . In contrary, the meta-heuristic methods ensure better solution due to their ability to find the global best solution. Further, in the following section, a detailed survey of metaheuristic based data clustering is presented.

2.7.2 Meta-heuristic Shift for Data Clustering

In 1991, the basic approach to emerge meta-heuristics based clustering algorithm was introduced by Selim and Alsultan [129], using simulated annealing. Thereafter, in 1994, Bezdek et al. [130] presented genetic algorithm based data clustering method, that was basically the first evolutionary based method of data clustering. Further, Sarkar et al. [131] presented the main challenges, encountered in meta-heuristic programming based clustering algorithms. The first, swarm based clustering algorithm was introduced by Lumer et al. [132] using ant colony optimization. In the following section, a systematic review of meta-heuristics based clustering algorithm proposed in the literature is presented.

2.7.3 Evolutionary based Metaheuristics in Partitional Clustering

Maulik et al. [9] introduced the first evolutionary based data clustering method using genetic algorithm in 2001. Subsequently, K-means based genetic algorithm was explored by Krishna et al. [133], in which the crossover operation of genetic algorithm was performed by K-means. In continuation, Lu et al. presented an incremental genetic K-means [134] and fast genetic K-means for the clustering of gene expression based data. Likewise, Sheng et al. [135] introduced K-medoid and genetic based method for the paritioning the large-scale dataset. However, it has been observed that hybrid evolutionary based algorithms, which are formed by merging the good features of two or more algorithms, had outperformed the parent algorithms. Cao and Lin [136] proposed a PSO and GA based hybrid algorithm, which was used for optimizing the SMT (surface mount technology) set up time. In continuation, Ye et al. [137] presented PSO and GA based application for image segmentation, which has been used for investigating the faults in the transmission lines. Further, Kwong et

al. [138] introduced a hybrid algorithm using ensemble learning steady-state genetic algorithm for performing efficient data clustering. Lorenab et al. [139] combined local search heuristic with GA and introduced a novel hybrid algorithm named 'Clustering Search'. The proposed algorithm was used to unfold capacitated centered problem. Moreover, He et al. [140] developed a 2 stage algorithm, in which mutation and selection were integrated to improve the search ability of the parent algorithm. The proposed algorithm surpassed the performance of genetic K-means and standard K-means algorithms. Falkenauer [141] explored, an algorithm named GGA (grouping genetic algorithm) to solve grouping related problems. The proposed algorithm performed well on UCI benchmark datasets in [142]. In continuation, Tan et al. [143] tested the efficiency of GGA for enhancing the spectral efficiency of multi-cast systems.

The evolutionary strategy is also one of the popular evolutionary algorithms that have been explored for optimizing the task of data clustering. In 1994, Murty et al. [144] introduced fuzzy and partition based clustering using ES. The summation of intra-cluster distance was used as the fitness function for performing the partition based clustering, while fuzzy C-mean (FCM) was the fitness function for the fuzzy-based clustering. Moreover, Schwefel [145] presented the recent works done in the area of clustering using ES. Ling and Ping [146] proposed hybrid clustering algorithm using K-means and ES. It has been perceived that the hybrid ES based algorithms performed better as compared to standard ES on the clustering problem on the majority of the UCI benchmark datasets.

2.7.4 Swarm based Meta-heuristics in Partitional Clustering

In 2003, Merve et al. [147] proposed the first swarm-based algorithm for partitional clustering using PSO. Thereafter, a number of PSO based clustering algorithms have been pro-

posed in the literature using the different variants of the conventional PSO. Cohen and Castro [148] introduced a novel variant of the PSO based clustering algorithm in 2006. Chuang et al. [149] presented an algorithm named, CPSO (combinatorial particle swarm optimization) to unfold the project scheduling problem. Cura et al. [12] developed particle swarm optimization based clustering method and solved the web application problems. Chuang et al. [150] presented a chaotic PSO based clustering algorithm, in which the conventional parameters of the PSO was replaced with chaotic operators. On the same footprints, Tsai and Cao [151] presented a PSO based clustering algorithm with selective particle regeneration. Likewise, Sun et al. [152] introduced yet another variant named QPSO (quantum-behaved PSO) for the efficient clustering of gene expression database.

Eusuff et al. [153] introduced meta-heuristic based clustering algorithm inspired by frog-leaping algorithm named shuffled frog leaping (SFL), which produced better results as compared to SA, ACO genetic K-means on the real life and synthetic datasets. Further, the SFL based clustering algorithm was successfully utilized for image segmentation and web text analysis [154]. Moreover, hybrid variants of PSO with K-means [147], K-harmonic means [155] and rough set theory [156] has been also introduced. Further, Du et al. [157] developed a hybrid PSO based clustering using K-means and particle-pair optimizer (PPO) for the analysis of microarray data. Zhang et al. [158] introduced possibilistic C-means(PCM) and PSO based variant for image segmentation. Niknam and Amiri [155] combined PSO, K-means, and ACO for the efficient data clustering.

Furthermore, some researchers proposed swarm and evolutionary based hybrid algorithms for the effective data clustering. Xu et al. [159] combined DE with PSO, and presented efficient results. Similarly, PSO was integrated with genetic algorithm [160] and simulated annealing [161] for the improvements in the clustering as compared to the

conventional PSO. The clustering algorithms based on the PSO and their hybrid variants has been successfully utilized for solving several real life applications, such as network anomaly detection [162], image clustering [163], color image segmentation [164], WSN [165], stock market analysis [166], gene expression [167], clustering for manufacturing cell design [168], and document clustering [169].

Furthermore, Cheng et al. [170] introduced fish swarm based clustering algorithm for the cluster analysis. The artificial bee colony optimization (ABC) is yet another popular meta-heuristic algorithm, that has been explored by the various researchers for optimizing the clustering task. Karaboga et al. [7] proposed a novel artificial bee colony based method for clustering the multivariate data. Zou et al. [171] introduced cooperative ABC based clustering.

Ant colony optimization (ACO) is another widely used optimization algorithm. A number of clustering methods based on ACO have been proposed in the literature. Chu et al. [172] introduced constrained ACO based clustering algorithm for the efficient analysis of arbitrarily shaped datasets. In continuation, and an adaptive ACO based clustering algorithm was proposed, which witnessed faster convergence as compared to conventional ACO [173][174]. Ghosh et al. [175] presented a novel variant of ACO, named aggregation pheromone density-based clustering (APC) and the same was applied in the image segmentation in [176]. Further, Yang et al. [177][178] introduced multi-ant colonies based algorithm for the effective data clustering, which worked on the concept of multiple ant colonies. Similarly, Handl et al. [179] proposed an improved ACO based clustering, which integrated adaptive and heterogeneous ants to make a better balance between exploration and exploitation. The proposed algorithm was successfully utilized for the document retrieval and topological mapping [180]. Moreover, hybrid algorithms of ANT colony opti-

mization have been also proposed by a number of researchers and utilized for unfolding several real-life applications. Kuo et al. [181] combined K-means and ACO algorithm, which was further improved by incorporating SOM (self-organizing maps)[182]. Subsequently, Jiang et al. [183] in 2017 presented a novel clustering algorithm using K-harmonic means, ACO and DBSCAN.

The strengths of the proposed ant colony based clustering algorithms have been appreciated and applied by various researchers in the several domains such as, intrusion detection[183], texture segmentation [184], web mining [185], high dimensional data analysis [186], gene expression data analysis [187] and test mining.

Recently, Kumar et al. [10] introduced a clustering algorithm which imitates the hunting behavior of grey wolves. In 2017, Ebrahimi et al. [188] introduced an adaptive meta-heuristic search-based method to cluster the sensors, deployed in the environment of IOT. The proposed algorithm has witnessed better performance as compared to the traditional clustering algorithms. Pandey et al. [2] proposed hybrid cuckoo search method for clustering twitter data for the sentiment analysis of the users. Pal et al. [80] proposed enhanced bio-geography based data clustering algorithm. Further, some other metaheurtics such as physics based and human behavior based are highlighted in the following subsection.

2.7.5 Other Meta-heuristics in Partitional Clustering

The physics based algorithms have been also explored by the various researchers for solving the partitional clustering problem. Alsultan et al. [129] leveraged the strength of simulated annealing (SA) for the efficient data clustering in 1991. Thereafter, in 1992 the SA based clustering was utilized by Huntley et al. [189] for untangling the multi-sensor fusion problem. Sun et al. [190] combined K-means, K-harmonic means and SA [191] and

introduced a hybrid algorithms, which outperformed the conventional SA on the majority benchmark UCI datasets. Likewise, Jin and Baoyu [192] developed a genetic annealing hybrid algorithm for partition based clustering, which was used to handle energy conservation problem of mobile ad-hoc network. Further, Lu et al. [193] combined multiple clustering using various measures of the partitions and proposed a fast simulated annealing algorithm for efficient clustering. Che [194] has successfully utilized the SA based clustering in supply chain management to efficiently manage the customer demand. Further, Merz [195] introduced memetic based clustering algorithm for the analysis of gene expression profiles. Similarly, Salehpour et al. [196] efficiently applied the memetic algorithm in wireless sensor networks for energy efficient clustering. Jiao et al. [197] applied memetic based clustering algorithm for the image segmentation of remote sensing images. Hatamlou et al. [8] introduced gravitational search algorithm based clustering algorithm, which used K-means for initializing the center heads. Recently, an exponential K-best gravitation search algorithm was introduced by Mittal and Saraswat to find the optimum threshold to perform multilevel image segmentation [198]. Nasraoui et al. [199] introduced an artificial immune system based clustering algorithm. Lue et al. [200], proposed Gene transposition based algorithm for the efficient clustering, in which immune cells (population) have been initialized with a vector of K cluster centroids. Likewise, Tan et al. [201] combined AIS with SVM (support vector machine) and presented a novel hybrid clustering algorithm. Thereafter, Nanda et al. [202] cloned global best particles of PSO and mutated them after updating position and velocity, the proposed variant was named as Immunized PSO (IPSO). Moreover, a clustering algorithm based on bacterial foraging optimization (BFO) was introduced by Lie et al. [203]. Subsequently, the proposed algorithm was successfully applied for clustering PPI based data, and for enlarging the coverage area of the wireless sensor net-

work (WSN)[204]. Further, Santosa et al. [82] introduced cat swarm optimization (CSO) based clustering approach and tested it on benchmark UCI datasets. Moreover, Senthilnath et al. [205] developed the firefly based algorithm for the efficient analysis of the clusters and tested in on the UCI datasets. Besides this, IWO (Invasive Weed Optimization) based automatic clustering algorithm was introduced by Chowdhury et al. [206]. Subsequently, Liu et al. [207] proposed multi-objective IWO based algorithm for the efficient clustering of the dataset.

2.8 Challenges of Meta-heuristic based Clustering

Producing acceptable solutions in a reasonable time is also one of the key features of meta-heuristic algorithms. Resources like memory have always been a concern, an algorithm is efficient if it uses the resources in an optimized way and always maintains a trade-off between time and space complexity.

At present, due to the tremendous growth in digital data, the complexity of real-world problems related to optimization has been also increased. The conventional metaheuristic computation is facing new challenges due to computation intensive objective function. To mitigate these concerns, researchers are working with parallel and distributed evolutionary computation nowadays.

A number of platforms are available for solving the computation-intensive problems using distributed evolutionary computation, such as CUDA, GPU and MapReduce based programming. Gong et al. [13] studied different distributed evolutionary models and appreciated the simplicity of Hadoop/MapReduce model for solving various computation-intensive problems. Hadoop [14] is an open source platform, developed and managed by Apache for

handling large datasets using distributed processing. Hadoop works with its own file system referred as HDFS (Hadoop distributed file system) and, is capable of processing zeta bytes of data with commodity hardware [15].

MapReduce provides the parallel computation platform and has successfully leveraged the strengths of metaheuristics algorithms for the analyses of large-scale datasets [16, 17, 18]. In the last one decade, Hadoop and MapReduce based model for parallel processing have been used by a number of researchers for solving complex real-world problems. In [208], the authors analyzed the techniques used for imbalanced big data processing using Random forest classifier. The problem of oversampling, undersampling and cost-sensitive learning was adopted using MapReduce to handle the large data sets and to correctly identify the unrepresented class. Zhu et al. [209] in their work proposed a parallel support vector machine (SVM) model for the prediction of large-scale protein-protein interactions (PPI). Since the process of finding protein-protein interaction is computation intensive and complex, the author employed the MapReduce based parallel architecture for training the SVM. Ma et al. [210] proposed a model based on the neural network for identifying rumors in the large data sets like microblog. Zhao et al. [211] presented parallel K-means using the MapReduce model to cluster the large-scale datasets.

Wang et al. [17] developed hybrid K-PSO method to handle complex problems. Subsequently, Pang et. al [212] introduced parallel genetic algorithm by leveraging the strength of MapReduce model. Meena [5] introduced an enhanced ACO based method for the feature selection of the text-based data. The proposed method was parallelized with the MapReduce model to handle the increased computational complexities.

Bhawani et al. [213] developed a parallel and hybrid method using DE, ACO, and K-means to analyze the useful species from their genome. Aljarah [214] introduced an intrusion de-

tection system using the parallel PSO. The proposed method was successfully utilized for analyzing high traffic data.

Wu et al. [215] presented a parallel ACO based method for the combinatorial optimization problem. Recently, Banharnsakun proposed MapReduce based artificial bee colony algorithm (MR-ABC) for handling large datasets.

2.9 Inferences from the Critical Review

- In the last two decades, more than hundred meta-heuristic algorithms have been proposed and utilized across the variety of domains. However, there does not exist a meta-heuristic based method which can handle the computational complexities of computation intensive problems.
- The meta-heuristic algorithms are able to provide promising results for unfolding the clustering problems. However, the big data clustering using these algorithms has not been explored in the literature.
- Literature witnessed that hybrid meta-heuristic algorithms outperformed the parent algorithms. Unfortunately, no hybrid model of the meta-heuristic approach has been studied for the clustering of massive datasets.
- The enhanced version of existing meta-heuristic algorithms have given prominent solution for solving various engineering problems. However, for analysis of big data sets these methods have not been explored.
- Literature discussed much work on the text data analysis problems such as twitter sentiment analysis and customer review analysis. However, the majority of the work

has focused on supervised machine learning based methods, which requires the labeled datasets. In addition, no method have been developed for big data analysis of the social media data or product review classification.

Chapter 3

Meta-heuristic Algorithms for Big Data Analysis

This chapter deals with development of parallel and distributed models of the meta-heuristic based clustering methods to handle big datasets. Three novel methods have been developed for the efficient clustering of the big datasets namely, dynamic frequency based parallel K-bat algorithm $(DFBPKBA)$*, bio-geography and K-means based clustering using MpReduce* $(MR-KBBO)$*, and enhanced grey wolf optimization* $(MR-EGWO)$*. The performance of the proposed methods have been analyzed on different parameters and the simulation results are compared with the other existing algorithms.*

3.1 Overview

With the progress of technology, there has been a significant increase in the growth of the digital data. Data mining techniques have automated the task of deriving the meaningful conclusions from large datasets in a short time frame. Clustering is the prominent approach

of unsupervised learning and considered as part and parcel of data engineering applications, such as image segmentation, data mining, information retrieval system, anomaly detection, medicine, computer vision and construction management [105, 106, 107]. Over the past years, several algorithms for data clustering have been introduced in the literature to handle the diversity in data and different sets of application requirements. K-means, one of the simplest and popular algorithm, has been employed for unfolding various clustering problems [4]. However, the results of K-means algorithm are highly biased on initial cluster centroids and it's probability of trapping into local optima is high [216].

To overcome the above concern, three novel methods are proposed in this chapter. Two methods namely, DFBPKBA and MR-EGWO are inspired from swarm intelligence while one method i.e MR-KBBO mimics the evolutionary approach. Further, the DFBPKBA and MR-KBBO are the hybrid methods while MR-EGWO is the the extension of the GWO based optimization.

The performance of the proposed methods have been vindicated in terms of the clustering quality and parallel performance effectiveness. The chapter is organized as follows: Section 3.2 briefs the preliminaries including data clustering approach, K-means algorithms and Hadoop MapReduce which have been used in the proposed methods. Section 3.3 presents the first hybrid method using bat algorithm and K-means. The second hybrid method named MR-KBBO is discussed in section 3.4. Section 3.5 presents the third method which is extension of GWO. Finally, section 3.7 summarizes the work done.

3.2 Preliminary

3.2.1 Data Clustering Approach

Clustering of a dataset in t-dimensional space is the process of assembling of N data objects into K groups on the basis of resemblance. Clustering partitions the data objects iteratively into k groups(clusters) in such a way, that the data objects within the same group have maximum resemblance. Further, data clustering is a type of unsupervised learning approach, that means data objects are grouped on the basis of structure of the data, without any training. Whereas, in supervised learning like classifications, data objects are classified based on the training set using labeled data. The proposed methods performs clustering with the known number of clusters. The summation of intra-cluster distance of K clusters is chosen as the criterion for the evaluation of the quality of the clustering. Let $Z = (z_1, z_2, z_3, \cdots, z_N)$ is a collection of N data objects where all the data object are represented in t dimensional space. The data objects are represented by a matrix of $Z_{n\times t}$ having n rows and t columns where each row vector describes one data object. The clustering process allocates the set of N data objects to K clusters and find a set of cluster centroids, $C = \{C_1, C_2, \cdots, C_K\}$ with the aim of minimizing the sum of squired euclidean distance between each data object Z_i and its centroids C_i to which it belongs. Generally, clustering process satisfies the following properties:

- Each and every cluster must have at least one data object, i.e., $C_i \neq \phi, \forall i \in \{1, 2, 3, \cdots, k\}$.
- Each data object certainly be part of a cluster.
- No data object can be part of more than one cluster, i.e., $C_q \cap C_r = \phi, \forall q \neq r$ and $q, r \in \{1, 2, 3, \cdots, k\}$.

A dataset is grouped based on the above three conditions and the quality of clustering is evaluated in terms of the fitness value. The sum of squared Euclidean distance [217] is one of the famous function used for the evaluating the quality of the clustering, which is calculated using Eq. (3.1).

$$MinD(Z, C) = \sum_{i=1}^{N} \sum_{j=1}^{k} w_{ij} \mid z_i - c_j \mid \tag{3.1}$$

Where N represents the number of data objects, $\mid z_i - c_j \mid$ is the ecludian distance of I^{th} data object from the J^{th} centroid which is defined in Eq. (3.2) . Further, w_{ij} represents the association weight of i^{th} review vector in the j^{th} cluster, i.e. the value of w_{ij} is 1 if the data object i is allocated to the cluster j otherwise 0.

$$d(Z_i, Z_j) = \sqrt{\sum_{t=1}^{t} (z_i^t - z_j^t)^2} \tag{3.2}$$

3.2.2 K-Means Algorithm

K-means [218] is popular data clustering algorithm which divides N data objects into in K clusters with the aim of minimizing the distance of each data object from its cluster centroid. The distance of each data object from its centroid can be calculate by using a distance metric such as euclidean distance [115] or cosine measure [116]. Further, the main motivation of the K-means algorithm is to partition the data in such a way, that minimizes the squared error between data objects and their centroid. The steps of K-means algorithm are presented in Algorithm 1.

Algorithm 1 :K-Means

1. Initialize the k cluster-heads by randomly chosen data points
2. Compute the distance of each data point with the center heads and assign it to closest the cluster
3. Update the cluster-heads c_k defined by the following formula: $c_k = \frac{1}{n_i} \sum_{\forall d_i \in c_i} d_i$ where d_i denotes the data points that belong to the cluster c_i and n_i is the number of data points in cluster c_i
4. Repeat steps 2 and 3 for maximum iterations or while convergence is not reached

3.2.3 Hadoop MapReduce

Hadoop [219] is a framework which provides distributed processing platform for large scale data processing. It distributes giant datasets across a set of computers which are connected in the hadoop cluster. Hadoop cluster is a collection of computers interconnected through local area network. It can easily scale up from single server to hundreds of machines for handling large datasets, concurrency control and failure recovery. Hadoop works on its own file system known as Hadoop Distributed File System (HDFS), to process the zeta-bytes of data. Furthermore, hadoop uses MapReduce [220] parallel programming model, which is used for processing large data on multiple machines. Fig. 3.1 depicts the architecture of the MapReduce programming model for the parallel computation. As shown in the Fig. 3.1 MapReduce splits the data into equal size small chunks called input splits. Further, MapReduce processes the data in the form of key/value pairs. The complete cycle of MapReduce consists of two main phases, namely Map and Reduce. In the Map phase, map function works on each key/value pair, processes it, and produces output again on the form of key/-value pair. Reduce phase, starts followed by the Map phase in which Reduce function is invoked on the output generated by the Map phase. The Map phase is basically designed for task decomposition while Reduce is responsible for the amalgamation of final results.

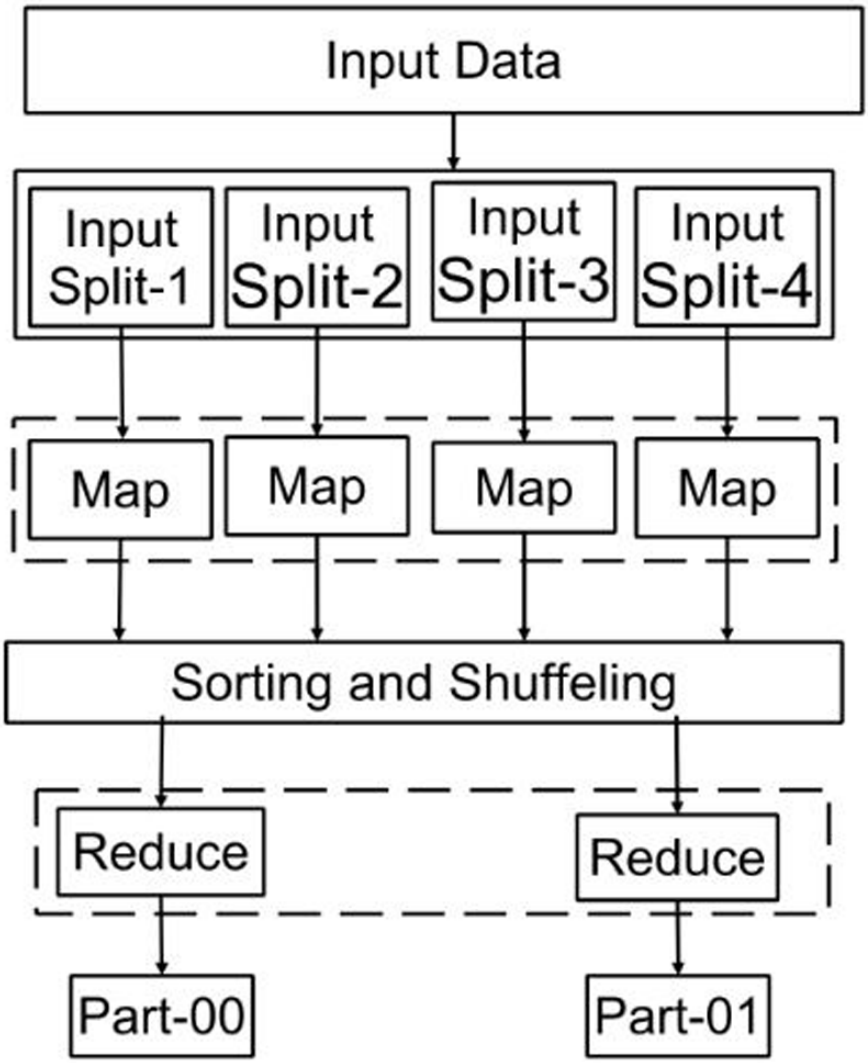


Figure 3.1: MapReduce parallel programming model

3.3 Dynamic Frequency based Parallel K-bat Algorithm

As discussed in the literature review, the hybrid methods have performed better than their parents on the majority of the clustering problems. This section presents a hybrid and parallel method for the efficient clustering of the big dataset. In the proposed method three contributions have been made. First, a novel variant of the bat algorithm based on the dynamic frequency is introduced. Second, the proposed variant is hybridized with K-means for the better initial population and faster convergence. Third, the proposed hybrid method is parallelized using the MapReduce model in the Hadoop framework. The experimental results show that the proposed method has outperformed K-means, PSO and bat algorithm on eighty percent of the benchmark datasets in terms of intra-cluster distance. Further, to test the parallel performance, the proposed method is run on a Hadoop cluster by increasing the number of nodes on each run. The results demonstrates that the proposed method is not only able to provide better clustering quality but also able to handle large scale datasets with a significant speedup.

3.3.1 DFBPKB based Clustering

In this section the proposed hybrid method of the clustering has been explained. The idea of the proposed method is to use the strength of bat algorithm for searching the best cluster centers and K-means algorithm for better population initialization. Bat is a recently introduced metaheuristic algorithm which has been widely used in the literature in the recent years as presented in section 2.6.3 of chapter 2. Bat is a simple, quick and easy to implement algorithm. However, the trade off between the exploration and exploitation of bat

algorithm is not proper as discussed in the literature. Moreover, the convergence of this algorithm becomes slow when clustering the large scale datasets due to increase in search space. To mitigate this, the bat algorithm is incorporated with the following capabilities.

- K-means is used to initialize the population, which gives a near by solution and makes the convergence faster.
- The frequency parameter of the bat algorithm has been tuned with course of iterations to make better trade off between exploration and exploitation.

The proposed hybrid model has improved the generated quality of clusters. However, the metaheuristic based hybrid methods fail to perform on big datasets due to high computation cost of the fitness function. Therefore, the proposed algorithm is parallelized using the MapReduce, which run in parallel over a cluster of commodity computers to handle the computational complexities of big datasets. Further, the detailed bat algorithm is defined as follows.

3.3.1.1 Bat Algorithm

Bat algorithm is popular metahueristic inspired by the echolocation behavior of the micro-bats and introduced by Yang in 2013[221]. Microbats fly in the air and finds the prey by utilizing their echolocation power. All bats uses echolocation power to detect the distance from the object. Moreover, each bat also have ability to differentiate the food and background barriers. They emits sound pulses according to their hunting strategies and than wait for the echo which is bounced back from the surrounding objects [217]. Each bat at position X_i flies randomly in the search space with some velocity V_i to find the prey which depends on the loudness A_0 and frequency F_i. The loudness factor is a variable which ranges from

A_{max} to value A_{min}. Each bat update its position defined by the Eq. (3.3).

$$pos_i^t = pos_i^{(t-1)} + vel_i^t \tag{3.3}$$

where pos_i^t, vel_i^t represents the position and velocity of the i^{th} bat in t^{th} iteration. The velocity vel_i^t is calculated using the Eq.(3.4).

$$vel_i^t = vel_i^{t-1} + (pos_i^{(t-1)} - pos^*) \times freq_i \tag{3.4}$$

Here, pos^* is the position of the bat having the best fitness value among all the bats. The frequency, $freq_i$ of the i^{th} bat is determined by Eq. (3.13).

$$freq_i = freq_{min} + (freq_{max} - freq_{min}) \times rand \tag{3.5}$$

The equation of local search around the best for the exploitation is defined as follows:

$$pos_{new} = pos^* + \epsilon * 0.001 \tag{3.6}$$

The equations for update of loudness and pulse rate are:

$$L_i^{t+1} = \alpha * L_i^t \tag{3.7}$$

$$rate_i^{t+1} = rate_0(1 - e^{-\gamma * t}) \tag{3.8}$$

Algorithm 2 describes the complete pseudo code of the bat algorithm.

Algorithm 2 : Bat Algorithm

Input: Objective function
Ouput: The position of best with best fitness value.
Generate initial population of N bats.
Initialize parameters α, λ, r_0, initial velocities V_i, current iteration i, maximum number of iteration $MaxItr$.
Evaluate the fitness of each bat using the objective function.
while $(i < MaxItr)$ **do**
  for each bat **do**
    Update the velocity of each bat defined by Eq. (3.4)
    Determine the new position of each bat using by Eq. (3.3).
    if (if $(rand > ri)$ **then**
      Find a local solution nearby the selected best solution.
    end if
    if $(rand < L_i$ & f(xi) < f(pos^*)) **then**
      Accept the new position
      Increase $rate_i$ and decrease L_i
    end if
  end for
  $i = i + 1$;
end while
Find the position of the bat with best fitness value

3.3.2 Dynamic Frequency based K-Bat Clustering

The efficiency of any metaheuristic algorithm lies on its ability to explore and exploit the search space intelligently. Generally, these algorithms explore the search space well in the initial phase to avoid local optima followed by exhaustive exploitation in the later stage to improve the search accuracy. However, the trade between exploration and exploitation of the bat algorithm is not proper due to which it sometimes falls into the local optima. Moreover, for data clustering, the search space is generally wide. Thus, good initial population is required for the proper convergence. To achieve the same, two contributions has been made in the proposed method for improving the clustering accuracy. First, the $freq_{max}$ parameter of the bat algorithm is improved. In the proposed method the value of $freq_{max}$ decreases with with the course of iterations as defined in Eq. (3.9), which was constant in the conventional bat algorithm.

$$freq_{max}(i) = (freq_{max} - itr_i * (freq_{max}/maxit)) \tag{3.9}$$

where $freq_{max}$ is the initial maximum frequency, $freq_{max}(i)$ is the current maximum frequency, itr_i is the current iteration number and $maxit$ is the maximum number of iterations for which the algorithm is run. Second, the population of the bat algorithm was initialized using 10 iterations of the K-means which gives a near by solution that is being optimizing by the proposed variant of the bat algorithm. For the data clustering, each bat in the proposed method is represented by a one dimensional array of $D \times K$, where D represents the number of dimensions in the dataset and K is the predefined number of cluster. Let us consider $bat_i = \{C_1, C_2, C_3, \cdots C_K\}$ as the i_{th} bat id (each bat represent one candidate solution), where $C_J = \{C_J^1, C_J^2, C_J^3, \cdots C_J^D\}$ represents J^{th} cluster centroid and $J \in (1, 2, 3...D)$.

The clustering process starts with the initialization of bat position and the sum of intracluster distance is minimized with the course of iterations. The cost function and the clustering approach is same as defined in section 3.2.1.

The complete pseudo-code of the proposed clustering method is described in Algorithm 3.

Algorithm 3 Dynamic frequency based K-bat clustering

Input: Data file having N data points with D dimensions and K Number of clusters.
Ouput: Final cluster centeroids. /* The position of bat after termination of algorithm represents centroids position*/

1: Create initial population of N bats using 10 iterations of K-means. /*centeroids generated by 10 iterations of K-means are the initial position of bats*/
2: Initialize parameters α, ϵ, f_{max}, f_{min}, number of iterations Itr .
3: Compute the fitness of each bat using Eq. (3.1).
4: Set the position of the best bat as pos^*.
5: **while** (Itr) or $(centroid\ stops\ moving)$ **do**
6: **for** each bat **do**
7: Update the velocity of each bat defined by Eq. (3.4)
8: Determine the new position of each bat using by Eq. (3.3).
9: **if** (if $(rand > ri)$ **then**
10: Find a local solution nearby the selected best solution.
11: **end if**
12: **if** ($rand < L_i$ & f(xi) < f(pos^*)) **then**
13: Accept the new position
14: Increase $rate_i$ and decrease L_i
15: **end if**
16: **end for**
17: **end while**
18: $i = i + 1$;
19: Return the position of bat with best fitness.

3.3.3 Parallelization of the Proposed Method using MapReduce

To handle the big data sets, a parallel version of the proposed clustering method (dynamic frequency based K-bat clustering) using Hadoop MapReduce framework is introduced,

termed as dynamic frequency based parallel K-bat algorithm (DFBPKBA). The proposed method works in two phases namely, DFBPKBA-Map and DFBPKBA-Reduce. For the data clustering using the proposed method, two main operations are performed. First, the fitness value for each bat is calculated as defined in (3.1). Second, the position of each bat is updated (bat position represents the centroids) which is defined by the new fitness value.

For the clustering of big data sets, computing the fitness value requires $N \times K$ number of operations. Therefore, in the proposed MapReduce based approach the large data set is broken into smaller chunks and distributed among the nodes of the Hadoop cluster. On each node of the cluster, the Map phase of the MapReduce job runs in parallel as described in section 3.2.3. Thereafter, each data point is converted in the form of key/value pairs by the record reader. The map phase, then reads the the data points in the form of key/value pairs and finds the centroid index of each data point by calculating the distance. The pseudo-code of the Map phase is detailed in Algorithm 4. The output of the Map phase is further a set of key/value pair in which key is a set of $\{batID, centroidID\}$, while the value is the distance of data point with the respective centroid-ID.

Thereafter, in the reduce phase, all the values with the same key are grouped and merged to compute the sum of the distances which represents the corresponding fitness value for each bat. The pseudo-code of the reduce function is presented in Algorithm 5. The velocity and position of each bat is updated according to Algorithm 3 as described in section 3.3.2. The completion of Map and Reduce phase makes one iteration of the proposed method. This process is continued for the maximum number of iteration or until the stopping criterion is not reached. Figure. 3.2 demonstrates complete architecture of the DFBPKBA.

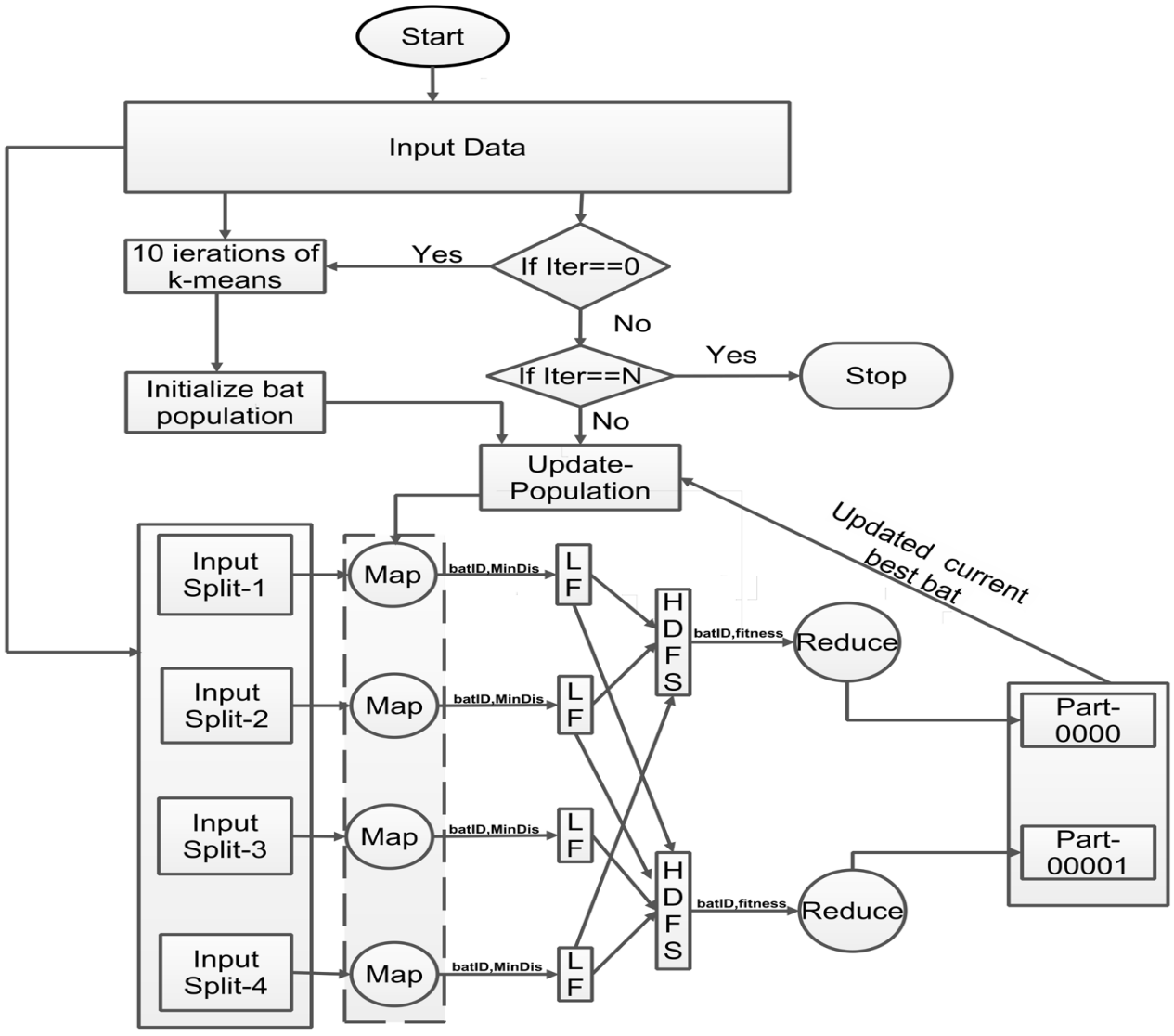


Figure 3.2: The procedure of DFBPKBA based on MapReduce

Algorithm 4 Map Function

Input: $Key - data, Value - Dataobject.$
Output: .Key-BatId, Value-Minimum distance
Map (Key : dataId, Value : data)
Bat-list = read (file);
For each bat;
$Minimun - distance$= getMinDistance (bat, data);
$Min - Distance$=$Minimun - distance + centroid - index$
write (badID, Min-Distance);

Algorithm 5 Reduce Function

Input: $Key - batId, Value - Min - Distance$
Output: .Key-Bat, Value-bat fitness
Reduce (Key: batID, Value: Min)
For each dist in Min-distance list
fitness=fitness+dist
if $(newfitness > oldfitness)$
$updateoldfitness = newfitness$
update position vector
decrease loudness
increase pulse rate
writeToFile (batID, fitness)

Table 3.1: Parameter values

Parameter Name	BAT	PSO	DFBPKBA
Population size (pop)	40	40	40
$freq_{min}$	0	–	0
$freq_{max}$	10	–	10
alpha	0.9	–	.9
gamma	0.01	–	0.01
r0	0.5	–	0.5
Inertia Weight	–	0.5	–
C1	–	0.5	–
C2	–	0.5	–

3.3.4 Performance Analysis

The performance of the proposed method is analyzed in two folds. First, the clustering quality is evaluated in terms of intra-cluster distance. Second the parallel performance is measured in terms of speedup measure obtained by running the proposed method by increasing the number of nodes in each run.

Table 3.2: Simulation results for the clustering Algorithm

Dataset Name	Criteria	K-Means	PSO	BAT	DFBPKBA
Iris	Best	97.5674	96.9087	104.786	**96.5555**
	Average	105.129	98.8976	118.9807	**96.6767**
Glass	Best	214.5467	223.7685	341.3211	**198.8769**
	Average	227.9778	230.49328	380.9874	**201.7654**
Wine	Best	16,566.77	**16,304.48**	16,768.66	16,396.03
	Average	16,963.05	**16,316.27**	17,094.89	16,461.38
Magic	Best	1,650,422.68	1,659,260.50	2,205,689.82	**1,645,851.75**
	Average	1,660,311.11	1,659,210.50	2,635,125.55	**1,647,410.30**
Pokerhand	Best	6,652,657.44	6,750,253.48	6,675,069.58	**6,031,523.30**
	Average	6,669,935.77	6,887,355.11	6,698,433.00	**6,055,335.71**

3.3.4.1 Clustering Quality Analysis

For analyzing the clustering quality, the proposed method is validated on five benchmark datasets taken from UCI machine learning repository [222] and the results are compared with three other methods, namely K-means, PSO and Bat. Table 3.1 presents the details of population size and the parameter values taken for the proposed and the considered methods. The number of iterations for each method was fixed as 500. Table 3.2 contains the best and average fitness value of each method obtained by 30 independent runs.

It is depicted from the table 3.2 that the proposed method has outperformed K-Means, PSO and Bat on the four datasets out of five in the terms of best and average fitness value. However for Wine dataset PSO has given competitive performance as compared to the other methods. Further, the K-means algorithm has given the least performance, since its results are highly dependent on the initial cluster centers. Thus, it is elucidated from experimental study that the proposed method can serve as an alternative tool for performing the clustering tasks.

3.3.4.2 Speedup Analysis

To test the parallel performance of the proposed method, a Hadoop cluster of 4 nodes was designed. Four benchmark datasets from UCI repository were used. Table 3.3 contains the number of data points in each dataset. To test the applicability of the method in the different sizes of datasets, 2 small size datasets namely, wine and magic, 2 large scale datasets i.e pokerhand and replicated wine were considered. The replicated wine dataset is formed by replicating each record of wine several times. Further, the speedup measure is defined by Eq. (3.10).

$$Speedup = T_{base}/T_N \tag{3.10}$$

where T_{base} is the running time when a method is run on 1 node while T_N is running time with N number of nodes. It is observed from the table 3.3, that the speedup measure for the wine dataset is minimum while it is maximum for the replicated wine which is the dataset with highest number of records. Also the running time graph of all the datasets was plotted as shown in Fig. 3.3 in which Y-axis represents running time in seconds and X-axis represents the number of nodes. It is depicted from the Fig. (3.3 c, d), that the running time of the DFBPKBA for pokerhand and replicated wine datasets decreases almost linearly with increasing number of nodes. However, the performance is dropped for the smaller datasets i.e wine and magic as visualized from Fig. (3.3a, b) due to hidden input output cost. Thus is concluded that the parallel performance of the proposed method increases with the increase in data size. Thus it is elucidated that the proposed method is an efficient alternative for clustering the large scale dataset.

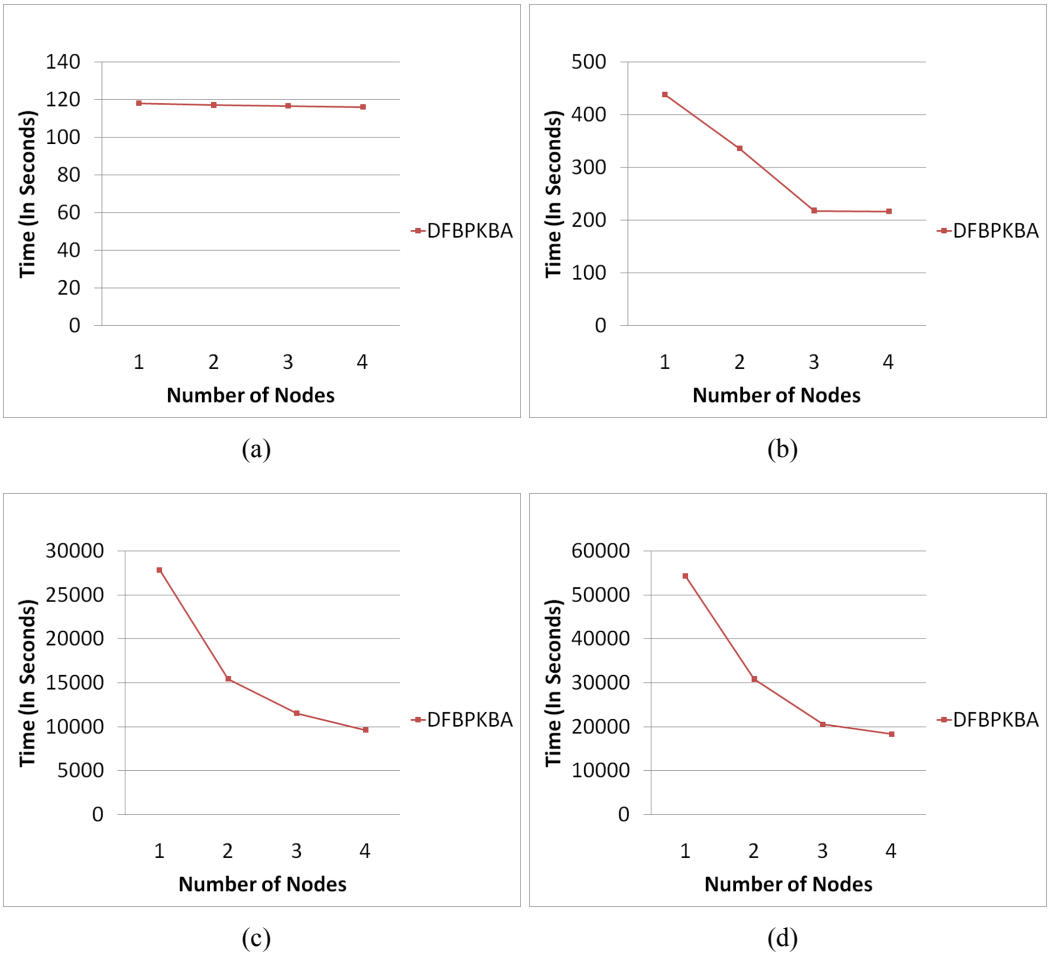


Figure 3.3: The speedup graph of (a) Wine (b) Magic (c) Pokerhand (d) Replicated Wine

Table 3.3: Speedup Values of DFBKBA on diffrent Nodes

Dataset Name	No of Records	2 Nodes	3 Nodes	4 Nodes
Wine	178	1.008547009	1.012875536	1.017241379
Magic	19020	1.305125149	2.011019284	2.023094688
PokerHand	1,000,000	1.800334185	2.41022842	2.887202766
Replicated Wine	1,780,000	1.757780133	2.637304901	2.95502798

3.4 Hybrid Clustering for Big Data using BBO and K-means

This section demonstrate a new hybrid method for clustering big datasets using the biogeography based optimization and K-means. The motivation behind the work is the absence of K-means and evolutionary algorithm based hybrid method for the big data analysis. The proposed method leverages a the strength of the popular biogeography based optimization which has witnessed promising performance over a number of real world problems in the recent years. The results demonstrate that the proposed method outperformed the DF-BPKBA presented in the previous section. The remainder of this section is organized as follows: Section 3.4.1 briefs the basics of BBO. The hybrid clustering using the BBO and K-means is detailed in section 3.4.2. Section 3.5.4 the proposed method for clustering big data.

3.4.1 Bio-geography Based Optimization (BBO)

Biogeography based optimization(BBO) is a meta-heuristic algorithm inspired by theory of island biogeography and proposed by Dan simon in 2008. The BBO algorithm is based on the distribution and equilibrium of species in the different islands. The species move across the Islands in the search of better one. In BBO, the quality of Island is mathematically represented by habitat suitability index (HSI) which depends on the various factors called suitability index variables (SIVs). The Islands with high HSI value share the information of their features with low HSI Islands. Further, the species migrate from low HSI island to high HSI island to attain equilibrium. The emigration and immigration rates of the species

are defined by Eqs. (3.11)-(3.12).

$$\mu_i = E \times (i/N) \tag{3.11}$$

$$\lambda_i = I \times (1 - i/N) \tag{3.12}$$

Where μ_i and λ_i are the emigration and immigration rates for the i_{th} island(individual) and N represents the population size. Algorithm 6 defines the main steps of the BBO.

Algorithm 6 :BBO

Initialize parameters $Pmig$, $Pmut$, N and $MaxItr$ as the migration probability, mutation probability, number of Islands and maximum number of iteration.
Initialize the value of suitability Index variable SIV of each Island(candidate solution) randomly in the search space
Calculate the value of HSI (fitness) of each island
while $(i < MaxIte)$ or (stopping criteria is not reached) **do**
  Sort the islands according to their HSI value
  Calculate the value of μ_i and λ_i based on the fitness of each island
  Apply the migration and mutation on the selected islands
  i=i+1;
end while
Return the best solution

3.4.2 Hybrid clustering Model (K-BBO)

The proposed clustering method leverages the strength of BBO for improving the quality of clustering and K-means algorithm for finding the initial cluster heads. K-means is a simple and popular clustering method, but it often fall into local optimum in the presence of noise. In the proposed method, first K-means algorithm is run for ten iteration which provide some near by solution and then BBO is used to optimize the clustering process. For any meta-heuristic algorithm, the results highly depends on the quality of initial candidate

solution. Since in BBO, population is initialized randomly which may increase the convergence time. Therefore, this method utilizes the K-means for generating good candidates of BBO which gives precise solution and better convergence rate. In the clustering process using the proposed method, the position SIV of each island represents a set of cluster centroids $(C_1, C_2, C_3, \cdots, C_K)$ for K clusters. The minimization of intra-cluster distance is considered as HSI(fitness) of each island which is calculated using Eq. (3.1). The optimal clusters corresponds SIV value of the island with the best fitness value returned after the completion of the algorithm. Further, the proposed MapReduce based method (MR-KBBO) is described in the following section.

3.4.3 Parallel K-BBO using MapReduce

In the proposed method, the main computation intensive task is to assign each data point a cluster head, which requires the calculation of distance for each data point from their center heads. For the data set of N input samples with D dimensions and K clusters, $K \times N$ distances has to be calculated for each iteration. Therefore, when the data size becomes large, traditional sequential algorithm are not able provide the results in the given time domain. In the MapReduce programming model this task of distance computation can be executed in parallel by distributing the task on multiple machines. For the first iteration, center heads are computed by running the K-means algorithm for 10 iterations and in the rest of iterations they are updated using BBO as described in section 3.4.1. The intra-cluster variance, is evaluated at the end of each MapReduce cycle, which represents the HSI (fitness) value each island(candidate solution). The value of λ_i and μ_i is calculate using the HSI value. Further, the islands(center heads) are updated according to λ_i and μ_i, by applying mutation and migration operation. This process is repeated till the maximum

number of iterations are reached or desired solution is not found. The complete MR-KBBO cycle consists of two phases namely; K-BBO-map and K-BBO-reduce. The K-BBO Map phase starts with retrieving the initial cluster heads and the data points from the HDFS. In this phase, the map function is invoked for each data point which is inputed to it in the form of key/value pair as described in section 3.2.3. The map function than computes the distance of a feature vector (data point) from each center-head and returns the minimum distance with its centroidID. Thereafter, Map functions emits the output in the form of key/value pair. The output key contains islandID with the centroidID of cluster to which data point belongs. Further, the output value contains the distance of data point with the respective centroidID. The complete pseudo code of the Map function is shown as follows.

Algorithm 7 :MR-KBBO Map

```
Map (Key: recordId, Value: Record)
Initialization:
key=record-ID
value=record
read(Biogeography) // this file contains population
for each island in biogeography ;
islandID = ExtractislandID(biogeograpgy)
centroidsArray = ExtractCentroids(biogeograpgy) // the SIV of Islands
minDist = returnMinDistance(record; centroidArray)
centroidID = i // i is the index of centroid with min distance
mapkey = (islandID+centroidID)
mapvalue = (minDist)
end for
write (mapkey, mapvalue);
```

Further, MR-KBBO-reduce phase starts followed by the MR-KBBO-map phase. The Reduce function gets the data grouped by the key, which is produced by the map phase. The reduce function key contains the islandID (candidate solution) and the centroidID while the $value$ contains the corresponding minimum distance. The Reduce function aggregates all the values with same key to find the total distance which serves as the HSI (fitness)

value of the island. The pseudo code of the Reduce phase is shown as follows. Further, the population is updated as per the new HSI value. This completes one iteration of the proposed method. This process continues unless the movement of centroids is not stopped or maximum iterations are reached.

Algorithm 8 :MR-KBBO Reduce

```
Reduce (Key:(gwoID, centroidID), value-list: minDistance)
Initialization
total-Distance = 0;
for each value in value-list
minDistance=retrieve minDistance(value)
total-Distance=total-Distance+minDistance
end for
write(key, total-Distance)
end function
```

3.4.4 Performance analysis

In section, the performance of the MR-KBBO, is studied on four large-scale synthetic datasets, formed by duplicating each record of the original dataset 10^7 times. Table 3.4 contains the number of clusters, number of dimensions, and number of data-points represented by ($\#C$), ($\#D$), and ($\#N$).

3.4.4.1 Clustering Quality Analysis

The clustering performance of the proposed method is validated in terms of the F-measure and the results are compared with the hybrid method presented in the previous section and 4 other MapReduce based state-of-the-art methods namely, K-means, ABC and MR-KPSO. Table 3.5 shows the mean of F-measure and computation time for 4 large scale synthetic datasets obtained by running each method on a cluster of 5 machines. The experimental

results confirms that the proposed MR-KBBO outperformed all the considered methods in terms of F-measure, while K-means has given the least performance. However, the computation time of K-means is less as compared to the meta-heuristics based clustering methods. Moreover, it is also pertinent from the results that DFBPKBA is the second best performer among the considered methods in terms of F-measure.mThus, it can be concluded that the proposed method can be used for efficient clustering of large datasets.

Further, a statistical comparison is performed to test the significant difference of the MR-KBBO and the other compared methods. For this, a non parametric statistical test, Wilcoxon rank sum test is conducted with 5% significance level. The p value has been computed for all the considered datasets using the fitness value of the proposed and compared methods. In Wilcoxon rank sum test, if the value of $p < 05(for 95\% confidence)$, the NULL hypothesis is rejected and symbolized by '+' or '-' other wise it is accepted and represented as '='. However, the '+' symbol indicates superiority and '-' indicate that inferiority of the proposed method with the considered methods. Table 3.6 demonstrates pair wise comparison of the significant levels(SGFT) of MR-KBBO with other compared algorithms. The value of $SGFT$ is '+' if $p < 05$ and mean fitness of the proposed method is better than the compared algorithm. Further, if $p < 05$ and mean fitness of the MR-KBBO is poor than the considered algorithm, the value of $SGFT$ is represented by '-'. It can be depicted from the Table 3.6 that $p - value < 0.5$ on all four considered datasets. Thus, it is concluded that the proposed method is significantly different from the considered methods on all the datsets.

Table 3.4: Large Datasets

Dataset	**#C**	**#D**	**#N**
Replicated Iris	3	7	10,000,050
Replicated CMC	3	9	10,000,197
Replicated Wine	2	18	5000000
Replicated Vovel	10	10	1025010

Table 3.5: Mean of F-measure and computation time of MR-KBBO over 30 runs

S.No	**Dataset**	**Criteria**	**parallel K-means**	**parallel K-PSO**	**MR-ABC**	**DFBPKBA**	**MR-KBBO**
1	Replicated Iris	F-Measure	0.667	0.785	0.842	0.790	**0.824**
		Computation time	**8.05E+04**	9.23E+04	9.26E+04	9.24E+04	9.23E+04
2	Replicated CMC	F-Measure	0.298	0.324	0.387	0.378	**0.390**
		Computation time	**8.24E+E04**	10.33E+E04	10.33E+E04	10.34E+E04	10.38E+E04
3	Replicated Wine	F-Measure	0.482	0.517	0.718	0.719	**0.715**
		Computation time	**11.20E+04**	16.14E+04	16.23E+04	19.24E+04	17.28E+04
4	Replicated Vovel	F-Measure	0.586	0.627	0.634	0.622	**0.631**
		Computation time	**10.50E+04**	13.22E+04	12.21E+04	13.23E+04	16.10E+04

Table 3.6: Results of Wilcoxon test for statistically significance level at $\alpha = 0.05$

S.No	**Dataset Name**	**MR-KBBO-PK-Means**		**MR-KBBO-parallel KPSO**		**MR-KBBO-DFBPKBA**		**MR-KBBO-MR-ABC**	
		P-Value	**SGFT**	**P-Value**	**SGFT**	**P-Value**	**SGFT**	**P-Value**	**SGFT**
1	$Iris$	3.35E-05	+	7.67E-05	+	3.40E-08	+	4.11E-06	+
2	CMC	1.45E-07	+	2.24E-11	+	3.55E-07	+	3.01E-09	+
3	$Wine$	5.45E-06	+	4.45E-65	+	4.03E-07	+	1.36E-07	+
4	$Vovel$	8.60E-09	+	7.09E-45	+	8.02E-05	+	6.08E-09	+

3.5 Optimized Big Data Clustering using Enhanced Grey Wolf Optimizer

Enhanced versions of meta-heuristic have given a prominent solution for solving numerous engineering problems. However, for the analysis of big data sets the above mentioned domain has not been explored. The absence of big data analysis as a contributory domain to metaheuristic based methods has been the stimulating factor for the motivation of the work. This section demonstrates a new clustering method named, MapReduce based enhanced grey wolf optimizer (MR-EGWO), for clustering big data sets.

The proposed method introduced a novel variant of grey wolf optimizer, Enhanced grey wolf optimizer (EGWO), where the hunting strategy of grey wolf is hybridized with binomial crossover and lévy flight steps are inducted to enhance the searching capability for pray. Further, the proposed variant is used for optimizing the clustering process. The clustering efficiency of the EGWO is tested on seven UCI benchmark datasets and compared with the five existing clustering methods namely K-Means, PSO, GSA, BA, and GWO in terms of intra-cluster distance. The convergence behavior and consistency in the results of EGWO has been validated through the convergence graph and boxplots. Further, the proposed EGWO is parallelized on the MapReduce model in the Hadoop framework and named MR-EGWO to handle the large-scale datasets. Moreover, the clustering quality of the MR-EGWO is also validated in terms of F-measure and compared with five MapReduce based state-of-the-art big data clustering methods namely, K-Means, K-PSO, MR-ABC, DFBPKBA, and MR-KBBO. Experimental results affirm that the proposed method outperformed the all the considered methods including the hybrid metaheuristic based methods $DFBPKBA$, and MR-KBBO presented in the previous section.

GWO is a popular swarm based metahueritic as discussed in the literature and widely used in the recent years. Recently, GWO based clustering has been also introduced in the literature which has given promising results. However, the this algorithm has not been explored fir the big data analysis. Therefore, in this thesis the strength of the GWO has been leveraged for the efficient clustering of the big data. Though, GWO has been proved to be efficient for the number of applications, still it has limitation of lack of population diversity. This results in slow convergence rate and risk of trapping into local optima. To improves its search precision, a novel variant of GWO, Enhanced grey wolf optimizer (EGWO), is proposed in this thesis by incorporating the following capabilities.

- Lévy Flight steps: To magnify search for prey.
- Binomial crossover: To inflate the attack to pray.

The overall contribution of this work has been divided into three folds. First, a novel clustering method is proposed based on the new variant of GWO. Second, the efficiency of proposed variant is studied on clustering problem. Third, the proposed method is parallelized using MapReduce architecture and named MR-EGWO for efficacious clustering of large datasets. The empirical analysis of EGWO has been done on seven UCI datasets and validated against five clustering algorithms namely K-Means [4], PSO, GSA, BA [223] and GWO in terms of Mean and Best of intra-cluster distance. The convergence behavior of EGWO is discussed along with the box plots to visualize its consistency. Moreover, the clustering performance of MR-EGWO is also validated in terms of F-measure by comparing with four MapReduce based parallel state-of-the-art namely, K-PSO, MR-ABC, DFBP-KBA, and MR-KBBO. To demonstrate the parallel computation performance, the proposed method (MR-EGWO) has been analyzed on four large scale datasets and asserted through

speedup graphs.

The rest of this section is organized as follows. Section 3.5.1 briefs the GWO algorithm. Section 3.5.2 describes the proposed variant i.e. EGWO. The parallelization of the EGWO is detailed in section 3.5.4. Section 3.5.5 demonstrates the experimental results.

3.5.1 Grey Wolf Optimizer

Grey wolf optimizer (GWO) is a swarm based metaheuristic introduced by Mirjalili et al., which imitate the hunting mechanism of the grey wolves. In GWO, grey wolves are grouped into $alpha(\alpha)$, $beta(\beta)$, $delta(\delta)$ and $omega(\omega)$ according to their social hierarchy. The best three grey wolves are considered as $alpha$, $beta$, $delta$ and renaming of the grey wolves are known as $omega$. The $alpha$ wolves are the commanding one and all other wolves follows their instructions. The second category of the wolves belonging to the $Beta$ category are responsible for helping $alpha$ in their decision making. $Omega$ are the lowest ranked grey wolves.

In GWO, the hunting is lead by $alpha$, $beta$ and $delta$ while $omega$ wolves are responsible for encircling the prey to find better solution. The encircling operation performed by the grey wolves is mathematically defined by Eq. (3.13) and (3.14):

$$\overrightarrow{D} = |\overrightarrow{C}.\overrightarrow{X}_p(i) - \overrightarrow{X}(i)| \tag{3.13}$$

$$\overrightarrow{X}(i+1) = \overrightarrow{X}_p(i) + \overrightarrow{A}.\overrightarrow{D} \tag{3.14}$$

where X_p is the location of the prey, $X(i)$ is the location of the grey wolf at i^{th} iteration.

$\overrightarrow{A}$, and $\overrightarrow{C}$ are coefficient vectors and computed using Eq. (3.15) and (3.16) respectively.

$$\overrightarrow{A} = 2\overrightarrow{a}.\overrightarrow{r_1} - a. \tag{3.15}$$

$$\overrightarrow{C} = 2.\overrightarrow{r_2}. \tag{3.16}$$

where $\overrightarrow{a}$ is coefficient vector which is reduced linearly from 2 to 0 with the increasing number of iterations and r_1, r_2 are the random numbers between [0,1].

$$\overrightarrow{D}_\alpha = |\overrightarrow{C}_1.\overrightarrow{X}_\alpha - \overrightarrow{X}| \tag{3.17}$$

$$\overrightarrow{D}_\beta = |\overrightarrow{C}_2.\overrightarrow{X}_\beta - \overrightarrow{X}| \tag{3.18}$$

$$\overrightarrow{D}_\delta = |\overrightarrow{C}_3.\overrightarrow{X}_\delta - \overrightarrow{X}| \tag{3.19}$$

Further, Eqs. (3.17), (3.18) and (3.19) define the estimated span around the current position and $alpha$, $beta$ and $delta$, respectively. After estimating of distances, the final position of the ω wolves is determined by Eq. (3.20). Where, $\overrightarrow{A}_1$, $\overrightarrow{A}_2$, $\overrightarrow{A}_3$ represents the random vectors, i shows the current iteration number and the vectors $\overrightarrow{X}_1$, $\overrightarrow{X}_2$, $\overrightarrow{X}3$, are defined by Eq. (3.21), (3.22), and (3.23) respectively.

$$\overrightarrow{X}(i+1) = \left[\frac{\overrightarrow{X}_1 + \overrightarrow{X}_2 + \overrightarrow{X}_3}{3}\right] \tag{3.20}$$

$$\overrightarrow{X}_1 = \overrightarrow{X}_\alpha - A_1.D_\alpha \tag{3.21}$$

$$\overrightarrow{X}_2 = \overrightarrow{X}_\beta - \overrightarrow{A}_2.\overrightarrow{D}_\beta \tag{3.22}$$

$$\overrightarrow{X}_3 = \overrightarrow{X}_\delta - \overrightarrow{A}_3.\overrightarrow{D}_\delta \tag{3.23}$$

3.5.2 Enhanced Grey Wolf Optimizer (EGWO)

The success of any metaheuristic depends upon the equilibrium between exploration and exploitation. The GWO algorithm has limitations of slow convergence rate and risk of trapping into local optima due to the lack of diversity in the wolves for certain cases . These limitations can be overcome by the increase of diversification and intensification of the search space. Therefore, in this thesis a novel variant of GWO named enhanced grey wolf optimizer (EGWO) is proposed. The proposed method is empowered with the advantages of lévy [217] flights and binomial crossover to improve the exploration and exploitation capabilities. The EGWO introduces two new phases to relieve the above mentioned problem.

3.5.2.1 Inflated Attack to Pray using Binomial Crossover

As the intensification of the population around the current best solution inflates the generation of optimal solutions. The exploitation in the proposed variant is enhanced by including one of the popular and widely used binomial crossover operator present in the literature. As $alpha$ wolf defines the current best position, its position can be used to define the better position of other wolfs. Hence, binomial crossover operator is performed between between the $alpha$ and the $X(i)$ to inflate the attack to the pray. The updated position (UP) of grey wolves is defined in Eq. (3.24).

$$UP_i^j = \begin{cases} \overrightarrow{X}_\alpha^j & (K \leq C) \\ \overrightarrow{X}_i^j & (K > C) \end{cases} \tag{3.24}$$

Where UP_i^j is the position of the i_{th} grey wolf in j^{th} dimension, K is the random number between [0,1]. $C \in [0, 1]$ is crossover constant.

3.5.2.2 Magnified Search for Pray using on Lévy flight

In GWO, the problem of stagnation still prevails in some cases, since the position updation of a wolf is determined solely by the positions of leader wolves namely, alpha, beta, and delta. Correspondingly, GWO results in immature convergence. To enhance the exploration capability, the proposed EGWO uses the concept of lévy flight to update the position of each wolf. As lévy flight defines steps of random lengths drawn from the lévy distribution [224], the chances of exploring the search space increases. In this thesis the Mantegna algorithm [217] is used to generate steps of random length. The Eq. (3.25) depicts the formulation of step length z defined by Mantega's algorithm.

$$z = \left[\frac{r}{\mid s \mid^{1/\beta}}\right] \tag{3.25}$$

where, $\beta \in (0, 2]$ is lévy index and r and s are variables following normal distribution of $N(0, \sigma_r^2)$ and $N(0, \sigma_s^2)$, respectively. The σ_r is calculated by Eq. (3.26) while σ_s is always 1.

$$\sigma_r = \left[\frac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\beta\Gamma[(1+\beta)/2]2^{(\beta-1)/2}}\right]^{1/\beta} \tag{3.26}$$

where, $\Gamma()$ is called Gamma function and defined by Eq. (3.27).

$$\Gamma(1+\beta) = \int_0^\infty t^\beta e^{-t} dt \tag{3.27}$$

In the proposed phase, each grey wolf takes lévy flight for the search of the prey and updates its position using Eq. (3.28).

$$\overrightarrow{X}_{t+1} = \overrightarrow{X}_t + estep_t \tag{3.28}$$

where, $\overrightarrow{X}_t$ is the position of the grey wolf at t^{th} iteration, $\overrightarrow{X}_\alpha$ represents the position of $alpha$ wolf and $estep_t$ at a particular iteration t defines the lévy flight step size and calculated by Eq. (3.29).

$$estep_t = 0.01 \times z \times (\overrightarrow{X}_t - \overrightarrow{X}_\alpha); \tag{3.29}$$

3.5.3 EGWO based Clustering

Furthermore, the proposed enhanced grey wolf optimizer (EGWO) is elucidated for the clustering problem. In EGWO based clustering, the position X of each grey wolf represents a set of cluster centroids $(C_1, C_2, C_3, \cdots, C_K)$ for K clusters. The minimization of intra-cluster distance is considered as the cost function and formulated in Eq. (3.1). The rest of the process is same as described in section 3.2.1. The optimal clusters corresponds to the position of the $Alpha$ wolf. The complete pseudo-code of the EGWO based clustering method is described in Algorithm 9.

The computation time of the EGWO based clustering is proportional to the size and the number of clusters in the dataset. In this thesis, EGWO generates the optimal cluster centroids with O$(N \times K \times t)$ operations for t iterations, where N is the number of data objects and K corresponds to the required number of clusters. Therefore, for P population size, the total time complexity of the proposed clustering method is O$(P \times N \times K \times t)$.

Algorithm 9 : Enhanced Grey Wolf Optimizer based Clustering

Input: Data file having Z data objects with t dimensions and K Number of clusters.
Ouput: Final centroids position. /* The location of α after termination of algorithm represents centroids position*/

1: Generate initial population of N grey wolves.
2: Initialize parameters a, i, A, C, maximum number of iteration $MaxItr$.
3: Evaluate the fitness of each grey wolf using Eq. (3.1).
4: Set top three grey wolves according to the fitness as $\overrightarrow{X}_{\alpha}$, $\overrightarrow{X}_{\beta}$ and $\overrightarrow{X}_{\delta}$.
5: **while** $(MaxItr)$ *or* $(centroid\ movement\ becomes\ zero)$ **do**
6: **for** each grey wolf **do**
7: Update the position of each grey wolf defined by Eq. (3.20)
8: Perform binomial cross over determined by Eq. (3.24).
9: Determine the new position of each grey wolf using lévy flight defined by Eq. (3.28).
10: Upgrade the values of a, A, C.
11: Calculate the fitness of each grey wolf.
12: Update $\overrightarrow{X}_{\alpha}$, $\overrightarrow{X}_{\beta}$, and $\overrightarrow{X}_{\delta}$.
13: **end for**
14: $i = i + 1$;
15: **end while**
16: Return $\overrightarrow{X}\alpha$ /*the position of alpha is the final centroid position*/

3.5.4 Parallelization of the EGWO

To demonstrate the applicability of EGWO on large dataset, a parallel version of EGWO algorithm using Hadoop MapReduce framework, MapReduce based EGWO (MR-EGWO), is presented. MR-EGWO works in two phases; EGWO-Map and EGWO-Reduce. Initially, MapReduce framework divides the large datasets into smaller chunks and distribute them uniformly among the hadoop nodes. Further, each data sample is converted into key/value pairs by the record reader. The MR-EGWO map phase, then, processes the input key/value pairs with cluster centroids in parallel and finds the centroid index of each data object. The pseudo-code of the MR-EGWO Map phase is presented in Algorithm 10. The output of this phase is the another set of key/value pair where key consists of $\{gwoID, centroidID\}$, while the distance of data object with the respective centroid-ID defines the value component. Further, the reduce function of the MR-EGWO reduce phase merges all the computed values with identical key's and computes the corresponding fitness value for each grey-wolf. Algorithm 11 presents the pseudo-code of the EGWO-reduce function. The $alpha$, $beta$, and $delta$ wolves are updated along with the position of each grey wolf according to the Algo 9. This marks one iteration of the MR-EGWO and this process is continued until the stopping criterion is reached. The complete architecture of the MR-EGWO for data clustering is shown in Fig. 3.4.

3.5.5 Performance Analysis

The proposed work is evaluated in two folds. First, EGWO is validated for clustering in terms of intra-cluster distance and convergence behavior. The comparison is made with k-means and four meta-heuristic algorithms for clustering namely; GSA, PSO, BA, and GWO.

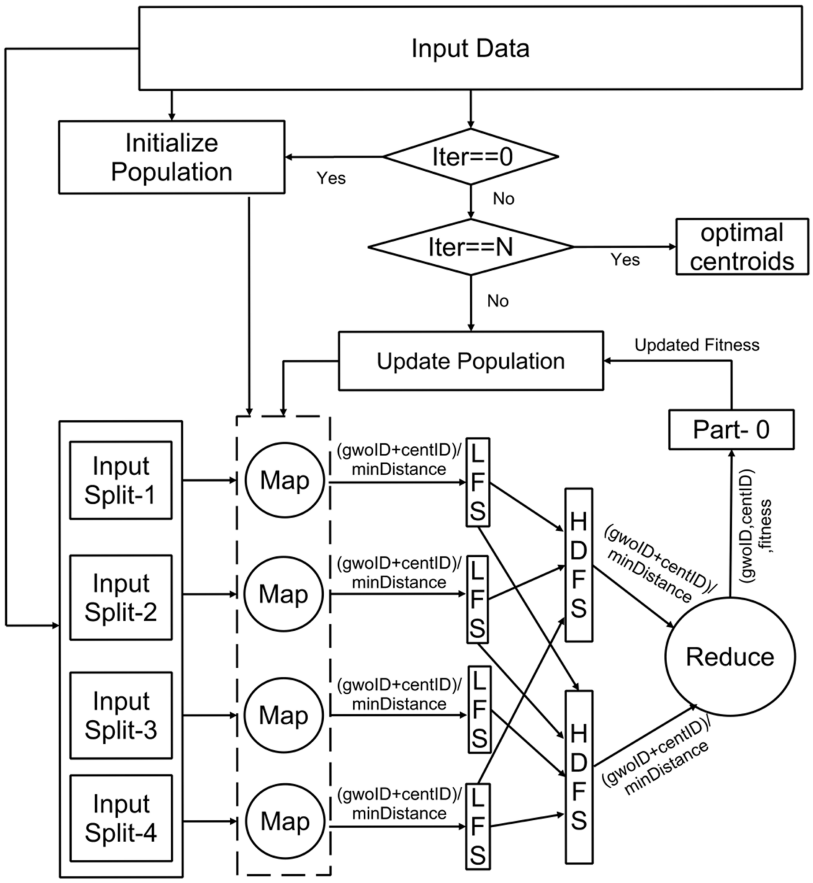


Figure 3.4: MapReduce architecture MR-EGWO for data clustering

Algorithm 10 :MR-EGWO Map

```
Map (Key: recordId, Value: Record)
Initialization:
key=record-ID
value=record
read(gwoPopulation);
for each wolf in gwo-population;
gwoID =retrieve-gwoID(gwoPopulation)
centroidArray =retrieve-centroids(gwoPopulation) /* wolf position represents centroids */
minDistance= getMinD(record, centroidArray);
/* The getMinD() function to get minimum distance is described below */
for each centroid do
   distance = get distance of i^th centroid from record
   if (distance < minDistance) then
      minDistance=distance
      centroid-ID = i / i represents index of the centroid array having minimum distance
   end if
end for
updated-key= gwoID+centroidID;
end for
write (updated-key, minDistance);
```

Algorithm 11 :MR-EGWO Reduce

```
Reduce (Key:(gwoID, centroidID), value-list: minDistance)
Initialization
fitness=0;
for each value in minDistance list do
   minDistance=retrieve-minDistance(value-list)
   fitness=+minDistance
   end for
   write(key, fitness)
end for
```

Table 3.7: Dataset Description

Dataset	NOC	NOF	NOI
Iris	3	4	150
Wine	3	13	178
Seeds	3	7	210
Glass	6	9	214
Cancer	2	9	638
Balance	3	4	625
Haberman	2	3	306

NOC:Number of Clusters
NOF:Number of Features
NOI:Number of Instances

Second, the effectiveness of the MapReduce based MR-EGWO is vindicated in terms of F-measure against the the two hybrid methods presented in the previous section and and three other state-of-the-art MapReduce based clustering methods namely, K-means, K-PSO, MR-ABC. Moreover, the speedup behavior of MR-EGWO is also studied by increasing number of nodes in each run.

3.5.5.1 Clustering Quality Analysis

The proposed EGWO is tested on seven benchmark datasets taken from UCI repository and results are compared with K-means, PSO, GSA, BA and GWO. Table 5.2 summarizes the seven considered benchmark datasets. The simulation is carried out for 30 runs on a system with Matlab 2015a, intel core i3 processor, $2.80GH_z$ frequency, 4 GB of RAM and 500 GB hard-disk. Table 3.8 details the parameter setting of the experimentation.

Table 3.9 defines the best and mean fitness values attained by the proposed and considered methods over 30 runs. It can be observed from the Table 3.9 that, EGWO outperformed all five methods on all the datasets in terms of best fitness value. For mean fitness value, EGWO has surpassed results for wine, seeds, glass and cancer. However, GWO has competitive results on Iris and Balance datasets while PSO performed well on Haberman dataset.

Table 3.8: Parameter values of the proposed and considered algorithms

Parameter Name	K-Means	PSO	GSA	BAT	GWO	EGWO	EGWO
Population size (pop)	–	40	40	40	40	40	40
Number of Iterations (itr)	500	500	500	500	500	500	500
Inertial Constant (w)	–	0.5	–	–	–	–	–
Cognitive Constant (c1)	–	1	–	–	–	–	–
Social Constant (c2)	–	1	–	–	–	–	–
G-constant (G0)	–	–	20	–	–	–	–
Loudness (α) –	–	–	.9	–	–	–	
Minimum frequency	–	–	–	0	–	–	–
Maximum frequency	–	–	–	2	–	–	–
Emission rate (γ)	–	–	–	.9	–	–	–
Pulse rate	–	–	–	.9	–	–	–
a	–	–	–	–	2	2	–
Crossover rate(C)	–	–	–	–	–	–	.1
Mutation rate(C)	–	–	–	–	–	–	.1

Moreover, to validate the performance difference in the proposed and tested methods, a non parametric statistical test, Wilcoxon rank sum test, is conducted at 5% level of significance. Table 3.10 contains the $p - value$ and SGFT($significance$) of each method. The null hypothesis is rejected if $p - value < 0.05$ and symbolized by $'+'$ or $'-'$, else, it is accepted and represented by $' ='$ symbol. The $'+'$ indicates that the method is different and significantly good while $'-'$ shows that it is different and significantly poor. It can be observed from table 3.10 that $p - value < 0.5$ on all the datasets. Correspondingly, it is assured that the EGWO is significantly different from the considered methods except GSA for balance dataset.

To demonstrate the improvement in exploration and exploitation trade-off, convergence behavior of the EGWO and considered methods are illustrated on two datasets, namely wine and seeds as shown in Fig. 3.5. Horizontal axis represents the iteration numbers while corresponding fitness values are aligned along the vertical axis. It can be visualized from Fig. 3.5 that EGWO prefers exploration at early stage of iterations and then lessen its exploration rate to perform the exploitation. In the later stage, this decline exploits the search space well for finding the optimal solution. Hence, it is pertinent from the convergence graphs that EGWO improves the exploration and exploitation abilities contrary to GWO.

Table 3.9: Best and mean fitness value over 30 runs

Dataset	Criteria	K-Means	PSO	GSA	BA	GWO	EGWO
Iris	Best	97.34084	96.78998	96.65548	96.65552	96.65826	**96.65548**
	Mean	106.33437	97.13691	96.67516	99.53097	**99.12574**	99.55645
Seeds	Best	587.31957	312.68370	311.79804	311.79816	311.88200	**311.79804**
	Mean	588.10457	313.85971	311.79804	315.41951	312.09220	**311.79804**
Glass	Best	292.75724	238.51144	286.11855	243.70331	265.81420	**214.44399**
	Mean	325.54765	257.06514	316.71044	264.10417	302.04114	**242.68894**
Cancer	Best	19323.17382	2969.23958	2970.17834	2964.38718	2964.390179	**2964.38697**
	Mean	19323.17693	2976.15128	2994.77937	3032.42259	2964.39495	**2964.38697**
Balance	Best	3472.32142	1423.96787	1423.82042	1424.04307	1423.82106	**1423.82040**
	Mean	3493.80000	1424.62818	1424.51503	1426.28547	**1423.82963**	1424.20479
Haberman	Best	30507.02076	2566.99548	2566.98989	2566.98889	2567.02562	**2566.98889**
	Mean	32271.96242	**2567.12294**	2582.08625	2648.88585	2590.77309	2637.34900
Wine	Best	2370689.68700	16298.98906	17038.59226	16371.05448	16307.09242	**16292.18465**
	Mean	2484626.08700	16305.11720	17709.43544	16865.72325	16318.41351	**16292.35069**

Table 3.10: Results of Wilcoxon Test for Statistically Significance level at $\alpha = 0.05$

Dataset	EGWO-K-Means		EGWO-PSO		EHGWO-GSA		EGWO-BAT		EGWO-GWO	
	P-Value	SGFT	P-Value	SGFT	P-Value	SGFT	P-Value	SGFT	P-Value	SGFT
$Iris$	4.45E-08	+	6.76E-05	+	2.40E-09	+	4.11E-06	+	1.42E-05	+
$Seeds$	2.78E-09	+	3.11E-11	+	2.55E-11	+	3.01E-09	+	3.11E-10	+
$Glass$	4.41E-08	+	6.28E-06	+	3.02E-11	+	1.36E-07	+	4.50E-11	+
$Cancer$	9.60E-10	+	3.01E-11	+	3.02E-11	+	3.02E-11	+	3.02E-11	+
$Balance$	5.02E-11	+	0.01E-0	+	0.10E-0	-	8.09E-10	+	0.00E-0	+
$Haberman$	2.01E-11	+	1.07E-07	+	5.18E-07	+	1.11E-06	+	6.52E-07	+
$Wine$	5.16E-08	+	3.01E-11	+	3.02E-11	+	3.12E-11	+	3.12E-10	+

Further, box-plots in Fig. 3.6 represent the consistency of the clustering results reported by the EGWO and other considered methods. Vertical lines of the boxes indicate variability of the best-so-far fitness value over 30 runs. Fig. 3.6a, 3.6b clearly illustrates that the degree of dispersion in EGWO is minimum, compared to PSO, GSA, BA, and GWO. Thus, it can be concluded from experimental analysis that EGWO is an efficient alternative for performing clustering tasks.

3.5.5.2 Speedup Analysis

Furthermore, the speedup performance of the MR-EGWO is analyzed on iris and CMC datasets. To measure the speedup performance of MR-EGWO, one machine is increased in the cluster on each run. The speedup performance of MR-EGWO is illustrated in Fig.

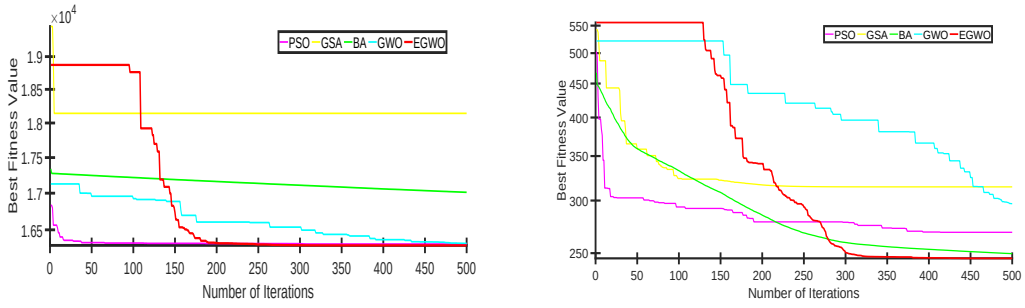


Figure 3.5: The convergence graphs of (a) Wine and (b) Glass

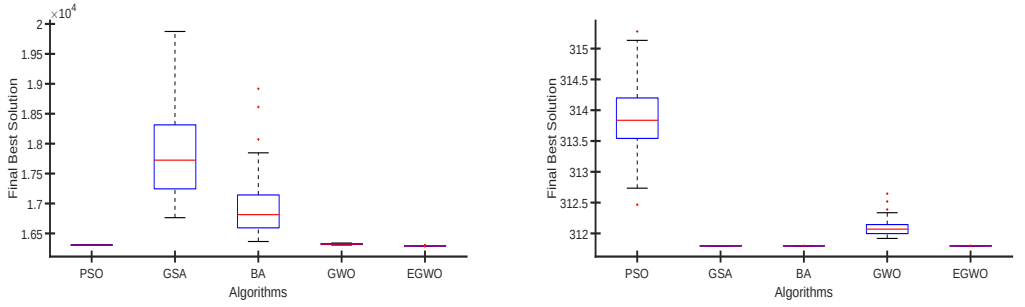
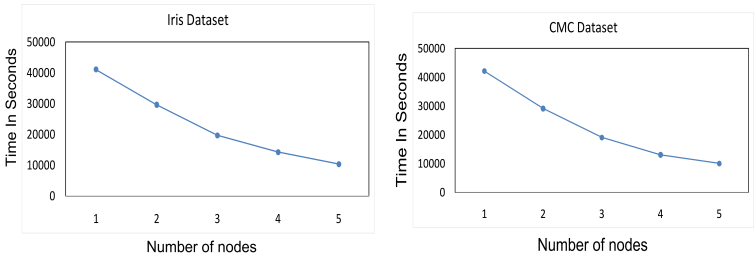


Figure 3.6: The box-plot graphs for (a) Wine and (b) Glass



(a) (b)

Figure 3.7: The speedup graph of (a) Iris (b) CMC

3.7. It can be concluded from the Fig. 3.7 that the running time of MR-EGWO decreases gradually with the increase of machines in the Hadoop cluster. The proposed method has achieved up to 4.6754, 4.3457 speedup measure on dataset 1 and 2 respectively with the 5 machine cluster. Therefore, it is affirmed that the proposed MR-EGWO is advantageous for large-scale data.

3.6 Performance Comparison of Proposed Methods

In section 3.5.5.1, EGWO has shown to be an efficient alternative for clustering task. Thus, the performance of the parallelized EGWO, (MR-EGWO), is analyzed. The datasets used and the execution environment is same as detailed in section 3.4.4. Table 3.11 shows the mean of F-measure and computation time for 4 large scale synthetic datasets obtained by running each method on a cluster of 5 machines. The F-measure comparison of four MapRedcue based methods as given in Table 3.11 confirms that the proposed MR-EGWO outperformed all the methods under comparison. MR-KBBO has shown second best performance on three datasets namely iris, wine, and CMC while DFBPKBA has been runner method in one dataset i.e wine. Further, K-means has given the least performance among all the considered methods in terms of F-measure. However, the computation time of K-means

Table 3.11: Mean of F-measure and computation time over 30 runs

S.No	Dataset	Criteria	parallel K-means	parallel K-PSO	MR-ABC	DFBPKBA	MR-KBBO	MR-EGWO
1	Replicated Iris	F-Measure	0.667	0.785	0.842	0.790	0.824	**0.846**
		Computation time	**8.05E+04**	9.23E+04	9.26E+04	9.24E+04	9.23E+04	9.22E+04
2	Replicated CMC	F-Measure	0.298	0.324	0.387	0.378	0.390	**0.391**
		Computation time	**8.24E+E04**	10.33E+E04	10.33E+E04	10.34E+E04	10.38E+04	10.32E+E04
3	Replicated Wine	F-Measure	0.482	0.517	0.718	0.719	0.715	**0.733**
		Computation time	**11.20E+04**	16.14E+04	16.23E+04	19.24E+04	17.28E+04	16.11E+04
4	Replicated Vovel	F-Measure	0.586	0.627	0.634	0.622	0.631	**0.635**
		Computation time	**10.50E+04**	13.22E+04	12.21E+04	13.23E+04	16.11E+04	13.21E+04

is lest as compared to the meta-heuristics based clustering methods. Moreover, the computation time of MR-EGWO is least among all the metaheuristic based methods on three datasets namely, iris, CMC, wine. However, the computation time of MR-EGWO is least among all the metaheuristic based methods. Thus, it can be concluded that the proposed method can be used for efficient clustering of large datasets.

3.7 Summery

This chapter presents the hybrid and extended models of metahueristic based clustering for the big data, in the parallel and distributed environment. Three novel methods have been introduced for the efficient clustering of the big data. First, a novel swarm based hybrid clustering method is presented and termed as DFBPKBA. The proposed method takes the advantage of the bat algorithm to achieve global optimal solution and MapReduce architecture to handle large scale datasets. The DFBPKBA changes the max frequency parameter of original bat algorithm at each iteration to leverage the exploration at the starting stage followed by the exploitation at the later stage. The results shows that the proposed algorithm has outperformed PSO, K-means and bat algorithm in the terms of quality of the clustering.

The speedup performance results show that the DFBPKBA algorithm is able to handle large scale datasets efficiently in reasonable amount of time.

Second, a novel evolutionary based hybrid clustering method (MR-KBBO) is introduced for analyzing large scale datasets. The proposed method leverages the strengths of the BBO for optimizing the clustering task and MapReduce model to handle large scale datasets. The random initialization process of BBO has been modified by the solutions obtained from 10 iterations of K-means to speedup the convergence. Further, the proposed method has been adopted on MepReduce architecture to handle big data sets. The results are compared with the DFBPKBA and three other methods in terms of F-measure. It is vindicated from the experimental results that the MR-KBBO has outperformed the DFBPKBA and the other methods in terms of F-measure. Third, a novel extended swam based clustering method using GWO is developed. The proposed method has three folds, (i) An efficient variant of grey-wolf optimizer called enhanced grey-wolf optimizer (EGWO) has been introduced for improving the quality of clustering (ii) The performance of the proposed variant (EGWO) is validated on seven benchmark datasets for clustering problem. The proposed method has outperformed five clustering methods namely: K-means, PSO, GSA, BA and GWO in terms of mean and best fitness values. The exploration and exploitation capabilities of the proposed variant are also analyzed using convergence graph. Boxplots are drawn to study the consistency of the results over the 30 runs. Third, a novel method named, MR-EGWO is proposed by parallelizing the EGWO using MapReduce for clustering large scale data sets. The proposed method, MR-EGWO, takes the advantage of EGWO to alleviate the clustering quality and MapReduce architecture to cope with large scale datasets.

Furthermore, to ascertain the efficiency of the MR-EGWO in the parallel environment, the proposed method is run on the Hadoop cluster of five nodes for four large scale syn-

thetic datasets namely, iris, CMC, wine, and vovel. The simulation results outperformed DFBPKBA and K-BBO in terms of F-measure. Moreover, the speedup efficiency of the MR-EGWO is studied on two synthetic datasets (iris and CMC) by varying the number of nodes of the Hadoop cluster. The speedup results show that MR-EGWO is well suited for analyzing large datasets with significant speedup performance and better clustering quality. Thus, it is concluded that MR-EGWO is a competitive method for large scale clustering problems.

Furthermore, MR-EGWO is a swarm based algorithm, thus is is concluded that swarm based methods are more powerful on the benchmark and real world datasets. This makes a motivating factor for developing a novel swarm based algorithm which can be be utilized for clustering the large scale datasets. In the next chapter, a new algorithm which mimics the military dogs behavior is introduced.

Chapter 4

Military Dog Optimizer for Big Data Clustering

It is vindicated from the previous chapter that the swarm based metaheuritics are performing better than evolutionary and physics based algorithms for the clustering of big data. In this chapter a novel meta heuristic algorithm based on military dogs squad is introduced for the big data clustering. The proposed algorithm is validated on 17 benchmark functions and compared with five other meta-heuristics namely particle swarm optimization (PSO), multi-verse optimizer (MVO), genetic algorithm (GA), probability based learning (PBIL) and evolutionary strategy (ES). The results are validated in terms of mean and standard deviation of the fitness value. The convergence behavior and consistency of the results have been also validated by plotting convergence graphs and BoxPlots. Further, the clustering effectiveness of the proposed algorithm validated on 7 benchmark datasets. Finally, the proposed algorithm is adopted on MapReduce architecture for clustering the big datasets.

4.1 Overview

Over the last three decades more then sixty meta-heuristic algorithms have been proposed by the various authors. Such algorithms are inspired from physical phenomena, animal behavior or evolutionary concepts. These algorithms have been widely used for solving the various real world optimization problems. Researchers are continuously working to improve the existing algorithms and also proposing new algorithms that are giving competitive results as compared to the existing algorithms present in the literature.

Moreover, these algorithm are gaining more and more popularity in the engineering domain due to their ability to bypass local optima and applicability across different disciplines, whereas the classical optimization algorithms are not able to provide a suitable solution for solving the optimization problems of high dimensionality. Since, the search space increases exponentially with the problem size, therefore solving these problems with the techniques like exhaustive search is impractical. Various heuristic approaches have been developed by the researchers to solve the global optimization problems such as Genetic algorithm (GA) [225], Particle swarm Optimization (PSO) [226], Gravitational search algorithm (GSA) [55], central force optimization (CFO) [227], Colliding Bodies optimization (CBO) [228], Magnetic charged system search (MCS) [228], Ray optimization [229], cuckoo optimization (CO) [230], Firefly algorithm (FA), etc. Meta-heuristics are the population based algorithms inspired from the nature. Each algorithms starts with the random set of solution called population. What makes the difference is the way of movements of population towards the global optima during the optimization process. These algorithms are tested and analyzed in the different domains of engineering. As No Free Lunch theorem clearly obviates the claim of an optimization algorithm for all optimization problems [231].

Table 4.1: Number of Scent Receptors for different Species

Species	Number of Scent Receptors
Humans	5 million
Dachshund	125 million
Fox Terrier	147 million
Beagle	225 million
German Shepherd	225 million
Bloodhound	300 million

Thus, the urge of new meta-heuristic algorithm is standstill. Therefore, in this thesis, a new meta-heuristic algorithm is proposed which leverages the searching ability of the trained military dogs.

Dogs are trained by the humans for object detecting and tracking purposes. They train them especially as military dogs, sniffer dogs, hunting dogs, police dogs, search dogs, and detector dogs. Military dogs are the category of dogs, especially trained for detecting substances like bombs, illegal drugs, wildlife scats, currency, or blood [232]. Mostly, military dogs work in groups called military dog squad to detect the object. They use the barking sound to locate or signal other dogs. Coren and Hodgson [233] studied that each sound of the military dog have some meaning associated with it. Military dogs have strong smell senses by which they are able to search the suspicious objects like bombs, wildlife scats, currency, or blood as well as they can communicate with each other by their barking. For example, loud sound of dog indicates insecurity. Baying sound indicates a call from the military dog to assure that his mates are alerted [233]. Generally, the smelling power of the dog is 1,000 to 10,000 times more than the humans or other species [233]. Table 4.1 shows the number of scent receptors in the various species. Moreover, the military dogs have the capability of deducing the direction of smell by moving their nostrils. Also, they have ability of storing meaningful information about the object in the form of scent while searching, which helps them in reaching to the desired object.

Furthermore, united states war dogs association studies stated that the smelling power

of dogs is effected by the wind. A dog may detect the suspected object up to 200 meters by the smell power if there is no wind. However, with the greater wind factor, the same can detect up to 1000 meters. Moreover, the factors like smoke and heavy vegetation are the confusing factors for a dog, as it confuse them in sensing the direction of actual smell or sound. This paper mimics the searching process of trained military dog squad to introduce a novel military dog based optimizer for finding the global optima. The overall contribution of this chapter has three folds. First, a new military dog based optimization has been presented. Second, the mathematical model of the proposed algorithm has been detailed. The validation of the proposed algorithm has been done against 17 benchmark functions and performance is measured in term of 4 parameters namely, fitness value, standard deviation, convergence behavior, and consistency in the results. The efficiency of the algorithm is compared with 5 existing meta-heuristics. Third, the clustering efficiency of the proposed algorithm is validated against 7 benchmark datasets. Fourth, the parallel model of the proposed algorithm has been introduced for handling big datasets.

Rest of the chapter is organized as follows. Section 4.2 discusses the related work. Section 4.3 presents the mathematical model of the military dog based optimizer. Section 4.4 provides the performance analysis. Section 4.5details the parallel model of the MDBO. The observations of the work done are elucidated in section 4.6.

4.2 Background Study

Nature-inspired meta-heuristic algorithms mimics the optimization behavior of the nature. Generally, these algorithms are population-based and start with a population of random solutions to obtain the global best solution. In contrast to this, there exists single-solution

based algorithms like hill climbing [43] and simulated annealing [44], which initiates the optimization process with a single solution. However, these algorithms suffer with the problem of local trap and premature convergence as they do not share any kind of information. On the contrary, population-based algorithms improve the solution over the iterations by information sharing. Two common aspects of the population-based algorithms are exploration and exploitation. Exploration represents the diversification in the search space, while exploitation corresponds to the intensification of the current solution. All population based algorithm tries to attain an equilibrium between exploration and exploitation to achieve the global best solution. Every agent of the meta-heuristic tries to improve its performance by sharing its fitness value with other agents at each iteration. The meta-heuristic can be broadly classified into three categories namely; physics-based, swarm-behavior based and evolutionary-based.

The physics-based algorithm optimizes the problem by imitating the physics based phenomenon. Gravitational search algorithm, proposed by Rashedi et al. [55], is one such algorithm which is based on Newtonian laws of gravity and motion. Hosseini [56] proposed an intelligent water drop algorithm which was inspired from the flow of rivers, as rivers often follow shortest path while flowing from source to destination. Further, Birbil [57] proposed an algorithm based on the concept of electromagnetism in which the properties of attraction and repulsion is used to attain a balanced trade-off between exploration and exploitation. Moreover, Mirjalili et al. [58] proposed multi-verse optimizer (MVO) in 2015, which is based on the notion of cosmology i.e white hole, black hole and wormhole. Some other physics based algorithms are Galaxy-based Search Algorithm (GbSA) [59], Black Hole (BH) algorithm, Small-World Optimization Algorithm (SWOA) [61], Ray Optimization (RO) [62], Curved Space Optimization (CSO) [63].

Swarm-based algorithms behave like the swarm of agents such as fishes or birds to achieve optimization results. Eberhart et al. [45] proposed the particle swarm optimization (PSO) which was inspired from the swarming behavior of fish or birds in search of food. Gandomi [46] presented an algorithm based on the simulation of the krill individuals. Mirjalili [74] proposed an ant-lion based optimizer that mimics the hunting mechanism of ant-lions. Moreover, Mirjalili [234] also introduced the moth-flame optimization, which simulates the death behavior of moths, in which the movement of agent is based on the transverse orientation based navigation of moths. Further, Wang et al. [47] proposed the hybrid krill heard algorithm to overcome the problem of poor exploitation capability of the krill herd algorithm. Ant colony optimization is another swarm based algorithm, which imitates the path finding behavior of ants [48]. Some other swarm based algorithm proposed in the literature are Cuckko search, Bat algorithm, Firefly optimization, Spider monkey optimization and Artificial bee colony optimization [235].

Evolution based algorithm are inspired from the biological evolution phenomena such as Darvin evolutionary theory. The evolutionary algorithms work on the principle of generating better individuals with the course of iterations by combining best individuals of the current generation. The popular genetic algorithm (GA) is an evolutionary algorithm based on the evolution of natural species. It maintains the balance between exploration and exploitation through the mutation and crossover operators. Another biological process based evolutionary algorithm is ES which gives almost equal importance to recombination and mutation, and it uses more than two parents to accord to an offspring. Baluja [53] proposed the probability-based incremental learning algorithm (PBIL) which manages only statics of the population rather than managing the complete population. Simon pre-

sented bio-geography based optimizer which is based on the immigration and emigration of the species between the islands of natural bio-geography. Differential evolution is another popular evolutionary algorithmic introduced by storm et al. [54].

4.3 Military Dog Optimizer

In this section, a new optimization algorithm based on the behavior of military dog's is introduced. Military dogs are the special trained dogs who go through a special training to search any specific type object, where they learn to identify thousands of scents. Moreover, military dogs undergo intense one on one training where they learn to work as a team to find the particular suspicious object for which they are trained. All these military dogs can communicate with each other by passing their message via barking. Hence dogs can cooperate with each other directly by passing message via the way of barking and its loudness. The loudness of barking indicates its closeness with the target object. When, a group of military dogs are left out for searching of a target object hidden in an open ground. The military dogs randomly start searching the area. With smelling sensation, the military dogs analyze a particular location and they define the fitness of the location in terms of loudness. The highest loudness indicates the best location among them. Military dog takes small step based on the scent smell in a particular location to exploit the local search area and it moves to explore the search space based on the loudness of barking. The smelling sensation analysis of the military dogs help them to take a move closer towards the target object and exploit the current location. Military dogs diverge from each other to search for the target object and converge to indicate that the target object is close.

4.3.1 Mathematical Model of MDBO

In this subsection the behavior of military dogs is mathematically simulated and explained. Some definitions for formalizing the MDBO algorithm are explained. Thereafter, the whole procedure of the MDBO is outline. In the given definitions R is used to refer the set of real numbers, ϕ is used to refer to empty set, while Z is used to denote the set of integers.

$Definition1$: A military dog squad MDS^m is a set of m trained military dogs. The size m of the military dog squad remains constant. Future work could allow variable size military dog squad.

$Definition2$: The feasible solution vector FSV^d, represents the position of a military dog in MDS. $FSV \in R^d$ is a set of all real numbers that represents the urine marking of a MDS.

$Definition3$: A military dog smell index $MDSI : MD \rightarrow R$ is a measure of goodness of the solution that is represented by a MD. In most of the population based algorithm this, $MDSI$ is called fitness of the individual.

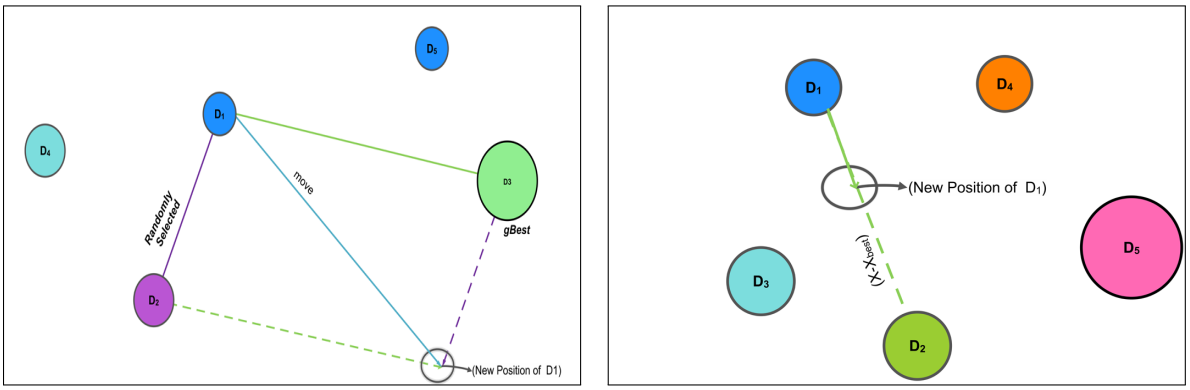
$Definition4$: Sniffing movement $\delta(p, \alpha) : MD \rightarrow MD$ is a probabilistic operator that randomly modifies the military dogs FSV^d based on the fitness of the loudest barking MD and movement probability P_m. Sniffing movement takes place using the following equation.

$$FSV_i^j(t+1) = \begin{cases} FSV_{lodest}^j, \ p \leq P_m \\ FSV_i^j + R(0,1) \times step(i), \ p > P_m \end{cases} \tag{4.1}$$

where,

$step(i) = \alpha \times K(0,1) \times (FSV_i^j - FSV_{loudest}^j)$

$Definition5$: Barking movement $\omega(p, q, K) : MD^n \rightarrow$ MD is a probabilistic operator



(a) (b)

Figure 4.1: The conceptual diagrams of MDBO for (a) Exploration Phase and (b) Exploitation Phase

that adjusts position of a military dog based on FSV^d of loudest barking and any randomly chosen Military Dog. The probability p, that position of MD is modified is constant and $q \in (1, 2, 3..., d)$ is the randomly chosen index. K is the wind factor that effects the sound coming from the other military dog.

The feasible solution vector (FSV^d) modification of a military dog is defined by:

$$FSV_i^j(t+1) = \begin{cases} FSV_i^j(t), \ p \leq P_m \\ FSV_i^j(t) + B_m \times K, \ p > P_m \end{cases} \tag{4.2}$$

where $B_m = (FSV_{loudest} - FSV_q)$ and $K \in R(0, 1)$ is any random number between (0,1).

$Definition6$: The MDS transition function $\phi = (m, d, \delta, \omega, P_m) : MD^m \rightarrow MD^m$ is a 5-tuple that modifies the MDS from one iteration to the next iteration. The MDS transition function begins by computing the feasible solution vector FSV^d and military dog smell index $MDSI$. Further, the MDS modification is performed on each military

dog MD followed by $MDSI$ recalculation for each military dog.

$Definition7$: A MDBO algorithm $MDBO = (H, \phi, T)$ is a three tuple that finds the solution for an optimization problem. $H :\rightarrow \{MD^n, MDSI^n\}$ is a function that creates an initial MDS and computes the corresponding $MDSI$. ϕ is a MDS transition function defined earlier. H is implemented using the random number generators inside the urine marking area of the military dog. $T : MD^n \rightarrow \{true, false\}$ is a termination criterion.

The MDBO algorithm can be informally described as follows:

1. The MDBO algorithm starts with the initialization of the MDBO parameters. In this step the method is derived for mapping the problem solution to FSV^d and MDS as described in definition 1 and 2, which are problem dependent. Also the maximum number of military dogs, sniff movement probability P_m, smell rate α, wind factor K are initialized according to the nature of the optimization problem.

2. Initialize the position of each military dog in the search space corresponding to the potential solution given in the problem. This is defined by the H operator described in definition 7.

3. Sniffing around current area (exploitation step): In this step, each MD modifies its FSV based on the information got from the loudest barking dog. While searching, the dogs take a random walk and steer around the new location. MD searches around the target object with a smell rate α and may either move directly towards the military dog at best position with movement probability P_m or they may take random movements according to its own position and the position of the MD nearest to the target object as described in definition 4.

   The pseducode of the sniffing movement is described as follows:

Algorithm 12 Military Dog based Algorithm (MDBO)

for $(i = 1\ to\ m)$ **do**
IF$(K < P_m)$
$FSV_i^j(t+1) = FSV_{loudest}^j$
$step(i) = \alpha \times K(0,1) \times (FSV_i^j - FSV_{loudest}^j)$;
$FSV_i^j(t+1) = FSV_i^j + R(0,1) \times step(i)$
end for

Table 4.2: Parameter values of algorithm of proposed and other algorithms

S. No.	Parameter	PSO	PBIL	GA	ES	MVO	GSA	GWO	MDBO
1.	Population Size (N)	50	50	50	50	50	50	50	50
2.	Number of Iterations (itr)	500	500	500	500	500	500	500	500
3.	Number of Dimensions (dim)	30	30	30	30	30	30		
4.	Elite Size ($keep$)	2	2	2	2	2	2		
5.	Inertial Constant (w)	$--$	0.3	$--$	$--$	$--$	$--$		
6.	Congnitive Constant (c_1)	$--$	1	$--$	$--$	$--$	$--$		
7.	Social Constant (c_2)	$--$	1	$--$	$--$	$--$	$--$		
8.	Mutation Probability (P_{mutate})	$--$	$--$	$--$	0.1	$--$	$--$		

4. Movement due to barking of other dogs(exploration step): It is the general nature of the military dogs that they bark loudly where they smell the suspected object. This creates a global movement of the military dogs. After a certain threshold of barking military dogs try to explore the search region with respect to the most loudly barking military dog. Each military dog takes a random move by considering the loudest barking military dog as global best and any randomly chosen barking military dog. The updated position is defined as per definition 5. The psedo-code of the barking movement is described as follows:

for $(i = 1\ to\ m)$ **do**
$K = rand(0,1)$
K=IF$(K < P_m)$
$B_m = rand * (FSV_{loudest} - FSV_q)$;
$FSV_i^{(i+1)} = FSV_i(t) + B_m \times K$
end for

5. Go to step three for the next iteration. This loop continues till the predefined number

of iterations, or the desired solution has been found . This is the implementation of the T operator described in definition 6.

Fig. 4.1a and Fig. 4.1b demonstrate the sniffing and barking movement of the MDs. It can be depicted from Fig. 4.1a that the sniffing movement represents the exploitation step of the MDBO. However, the barking movement corresponds to the exploration step, as its movement is influenced any randomly chosen MD.

4.3.2 MDBO based Clustering

In the MDBO based clustering, the FSV of each military dog represents a set of cluster centroids, $C = \{C_1, C_2, \cdots, C_k\}$ for K clusters. The $MDSI$ value of each military dog corresponds to the sum of squired Euclidean distance as defined the previous chapter.

4.4 Performance Analysis

The performance of the proposed algorithm is evaluated in two folds, first the MDBO is validated on benchmark functions and results are are detailed in section 4.4.1. Second, the clustering effectiveness of the MDBO is vindicated on 7 benchmark datasets and results are presented in section 4.4.2.

4.4.1 Performance Analysis on Benchmark Function

In this section, the performance and uniqueness of the proposed MDBO is analyzed and compared with five recent population based algorithms. Seventeen standard benchmark functions given in Table 4.3 are used for comparison of algorithm based on mean and standard deviation. Convergence behavior of MDBO is analyzed and compared with other

Table 4.3: Benchmark Functions

Sr. No.	Function Name	Equation	Range	Optimal value	Optimal position values	Category
1	Ackley	$F_1(X) = -20e^{-0.02\sqrt{d^{-1}\sum_{i=1}^{d} x_i^2}} - e^{d^{-1}\sum_{i=1}^{d} cos(2\pi x_i)} + 20 + e$	-32,+32	0	$(0, \cdots, 0)$	Multi-Model
2	Alpine	$F_2(X) = \sum_{i=1}^{d} \lvert x_i \sin(x_i) + 0.1x_i \rvert$	-100,+100	0	$(0, \cdots, 0)$	Multi-Model
3	Dixon and Price	$F_3(X) = (x_1 - 1)^2 + \sum_{i=2}^{d} i(2x_i^2 - x_{i-1})^2$	-100,+100	0	$(0, \cdots, 0)$	Unimodal
4	Griewank	$F_4(X) = 1 + \sum_{i=1}^{d} \frac{x_i^2}{4000} - \prod_{i=1}^{d} \cos(\frac{x_i}{\sqrt{i}})$	-20,+20	0	$(0, \cdots, 0)$	Multi-model
5	Levy	$F_5(X) = \sin^2(\pi\omega_1) + \sum_{i=1}^{d-1} (\omega_i - 1)^2 \left[1 + 10\sin^2(\pi\omega_i + 1)\right] + (\omega_d - 1)^2 \left[1 + \sin^2(2\pi\omega_d) +\right], \omega_i = 1 + \frac{x_i - 1}{4}, for all i = 1, \cdots, d$	-50,+50	0	$(0, \cdots, 0)$	Multi-Model
6	Pathological	$F_6(X) = \sum_{i=1}^{d-1} \left(0.5 + \frac{\sin^2\sqrt{100x_i^2 + x_{i+1}^2} - 0.5}{1 + 0.001(x_i^2 - 2x_i x_{i+1} + x_{i+1}^2)^2}\right)$	-100,+100	0	$(0, \cdots, 0)$	Multi-model
7	Perm	$F_7(X) = \sum_{i=1}^{d} \left(\sum_{j=1}^{d} (j + \beta) \left(x_i^j - \frac{1}{j_i}\right)\right)^2$	-100,+100	0	$(1, 1/2, \cdots, 1/d)$	Multi-Model
8	Powell	$F_8(X) = \sum_{i=1}^{d/4} [(x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-3} + 2x_{4i-2})^4 + 10(x_{4i-3} + x_{4i})^4]$	-10,10	0	$(0, \cdots, 0)$	Uni-model
9	PowellSum	$F_9(X) = \sum_{i=1}^{d} \lvert x_i \rvert^{i+1}$	-100,+100	0	$(0, \cdots, 0)$	Uni-Model
10	Rastrigin	$F_{10}(X) = 10d + \sum_{i=1}^{d} (x_i^2 - 10cos(2\pi x_i))$	-5.12,+5.12	0	$(0, \cdots, 0)$	Uni-Model
11	Rosenbrock's	$F_{11}(X) = \sum_{i=1}^{d-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	-30,+30	0	$(0, \cdots, 0)$	Multi-Model
12	Rotated Hyper-Ellipsoid	$F_{12}(X) = \sum_{i=1}^{d} \sum_{j=1}^{i} x_j^2$	-65.536,+65.536	0	$(0, \cdots, 0)$	Uni-Model
13	Schumer Steiglitz	$F_{13}(X) = \sum_{i=1}^{d} x_i^4$	-100,+100	0	$(0, \cdots, 0)$	Uni-model
14	Schwefel	$F_{14} = -\sum_{i=1}^{d} x_i \sin\sqrt{\lvert x_i \rvert}$	-500,+500	0	$(0, \cdots, 0)$	Multi-Model
15	Sphere	$F_{15}(X) = \sum_{i=1}^{d} x_i^2$	-100,+100	0	$(0, \cdots, 0)$	Uni-Model
16	Step	$F_{16}(X) = \sum_{i=1}^{d} (\lfloor \lvert x_i \rvert \rfloor)$	-100,+100	0	$(0, \cdots, 0)$	Uni-model
17	Trigonometric	$F_{17}(X) = \sum_{i=1}^{d} [d - \sum_{j=1}^{d} \cos x_j + i(1 - cos(x_i) - sin(x_i))]^2$	0,3.14	0	$(0, \cdots, 0)$	Uni-Model

algorithms by plotting the convergence graph for each benchmark function. Box plots are employed to visualize and establish the consistency of the proposed MDBO algorithm. Box plots are non parametric methods to display variations in results of proposed MDBO algorithm and compared the same with five algorithms on sixteen benchmark functions without making any assumptions of the underlying statistical distribution. Wilcoxon rank test shows dissimilarity of MDBO with other algorithms.

4.4.1.1 Accuracy

The proposed MDBO was tested on the minimization functions and the results were compared with five other algorithms namely MVO, ES, Pbil, GA as well as PSO. Table 4.2 contains the values of population size, number of dimensions, number of iterations, social constant, cognitive constant and mutation probability used in simulation. Table 4.3 contains the details of the seventeen benchmark functions including range values, optimal position values and categories upon which the proposed algorithm has been tested and compared. Each function either belongs to uni-model or multi-model class. Nine uni-model functions are used to test the convergence rate and eight multi-model functions are used to test the local optima avoidance capability of the algorithm. Further each algorithm was run fifteen times on each benchmark function to get the mean and standard deviation. Table 4.4 shows the values of mean and the standard deviation of fitness values computed in fifteen rounds by each algorithm. From the comparison of mean and standard deviation of seventeen benchmark functions for six algorithms as given in Table 4.4 it is observed that proposed MDBO outperformed all five algorithms under comparison on sixteen benchmark functions in terms of mean fitness values. However ES performed better than MDBO for only one benchmark functions i.e F6 with mean value 4.48 as compared to 4.68 mean value of

proposed MDBO. Further standard deviation of the proposed MDBO is minimum for sixteen benchmark functions while ES has given minimum value of the standard deviation for one function i.e., F1. It can be observed that in all the nine uni-model functions proposed MDBO algorithm has beat all other algorithms showing stronger local search ability. However proposed algorithm outperformed other algorithms in seven multi-model functions out of eight which confirms stronger exploration capability of the proposed algorithm.

4.4.1.2 Wilcoxon Test

The uniqueness of the proposed algorithms have been statistically validated using Wilcoxon rank sum test. NULL hypothesis assumes that the two algorithms are similar at the five percent significance level α for benchmark functions. p values has been computed for all the benchmark functions using the fitness values of compared and proposed algorithms. If the value of p$<$0.05 then null hypothesis is rejected and symbolized by '+' or '-', otherwise it is rejected and represented by symbol '='. However '+' indicates better result and '-' represents poor results of the proposed MDBO algorithm. Table 4.6 shows the results of Wilcoxon rank sum test for the NULL hypothesis over seventeen benchmark functions explained Table 4.3. The proposed MDBO algorithm is compared with ES, PSO, MVO, Pbil, and GA on the basis of $p-$ values value. The p value is computed by running fifteen iterations of each algorithm on all functions. A pair wise comparison of MDBO with other algorithms shows significant levels on the basis of p value, mean and standard deviation. Significant level is positive if p value is less than 0.05 and the value of mean and standard deviation are less than the compared algorithm. It is observed from the Table 4.6, that MDBO has outperformed ES on all the benchmark functions except F6 where ES has given competitive result. Further, MDBO has surpassed PSO for all the benchmark functions.

Table 4.4: Comparison of mean fitness and standard deviation values for 15 runs on benchmark functions for existing and proposed algorithms

Fun	PBIL		PSO		GA		MVO		ES		MDBO	
	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD
F_1	4.51E+00	0.19E+00	3.55E+00	0.30E+00	2.18E+00	4.32E-01	1.63E+00	4.50E-01	6.31E+00	**1.88E-01**	**9.71E-01**	7.39E-01
F_2	4.89E+02	3.54E+01	1.02E+00	5.01E+00	5.29E+01	1.15E+01	1.27E+02	3.42E+01	4.91E+02	3.67E+01	**2.34E+00**	**1.91E+00**
F_3	8.56E+09	1.84E+09	3.06E+08	3.03E+08	1.05E+06	1.06E+06	5.71E+03	5.32E+03	8.26E+09	2.54E+09	**1.90E+02**	**7.28E+02**
F_4	1.46E+00	0.06E+00	1.11E+00	0.04E+00	1.07E+00	2.57E-02	1.50E-02	1.25E-02	1.54E+00	4.98E-02	**2.79E-03**	**4.26E-03**
F_5	3.69E+03	5.89E+02	5.21E+02	2.13E+02	5.32E+01	1.86E+01	4.83E+02	3.14E+02	3.70E+03	3.60E+02	**2.22E+01**	**2.35E+01**
F_6	9.37E+00	0.66E+00	7.80E+00	0.52E+00	5.65E+00	4.94E-01	9.93E+00	3.62E-01	**4.48E+00**	**3.19E-01**	4.68E+00	8.03E-01
F_7	2.00E+117	3.60E+117	3.30E+98	1.20E+99	5.59E+58	2.15E+59	1.05E+17	4.05E+17	4.50E+117	6.50E+117	**1.00E+10**	**0.00E+00**
F_8	4.62E+04	93.54E+02	1.82E+04	7.40E+03	1.86E+02	1.22E+02	7.78E+00	5.49E+00	1.24E+05	2.91E+04	**9.65E-03**	**7.42E-03**
F_9	4.46E+41	4.99E+41	1.32E+54	4.80E+54	3.33E+16	1.20E+17	2.96E+11	4.10E+11	8.41E+43	1.70E+44	**1.00E+10**	**0.00E+00**
F_{10}	1.54E+02	1.17E+01	1.43E+00	3.94E+01	1.40E+01	6.49E+00	1.23E+02	3.56E+01	4.14E+02	1.96E+01	**2.07E+01**	**5.61E+00**
F_{11}	1.39E+08	3.50E+07	3.67E+06	3.10E+06	9.47E+03	1.03E+04	4.58E+02	5.30E+02	1.50E+08	2.87E+07	**1.02E+02**	**4.02E+01**
F_{12}	2.46E+01	1.99E+04	9.56E+04	7.77E+04	4.88E+03	2.50E+03	1.83E+01	1.31E+01	2.79E+05	2.70E+04	**8.89E-12**	**5.32E-12**
F_{13}	1.87E+08	3.56E+08	7.40E+06	6.77E+06	4.25E+04	3.49E+04	1.01E-01	5.52E-02	1.55E+08	4.13E+07	**3.55E-16**	**5.74E-16**
F_{14}	8.65E+00	3.32E+02	6.57E+03	1.07E+03	3.22E+03	6.44E+02	4.53E+03	7.70E+02	8.41E+03	3.79E+02	**5.48E+02**	**2.20E+02**
F_{15}	4.52E+00	4.14E+03	1.02E+04	2.70E+03	1.05E+03	4.98E+02	6.76E-01	1.91E-01	4.62E+04	4.21E+03	**2.07E-12**	**1.58E-12**
F_{16}	9.32E+02	6.70E+00	4.82E+02	7.82E+01	1.82E+02	5.65E+01	5.60E+00	4.73E+00	9.12E+02	5.87E+01	**2.67E-01**	**7.04E-01**
F_{17}	0.89E+00	2.36E+00	7.67E+03	6.86E+00	5.05E+00	1.96E+01	1.85E+00	9.18E-01	7.68E+03	2.36E+03	**2.74E-01**	**5.10E-01**

When MDBO is compared with MVO it has beaten on sixteen benchmark function out of seventeen. However for one function i.e., F6 GA performed well. Moreover, MDBO has given positive significance on all the benchmark functions when compared with PBIL and GA except for F6 function. Hence is can be concluded that the proposed algorithm is significantly different and outperforms five existing algorithms i.e., MVO, ES, PBIL, GA as well as PSO on each benchmark function.

4.4.1.3 Convergence rate

The convergence behavior of the proposed MDBO algorithm is analyzed and compared with other five existing algorithms by plotting the convergence graph for each bench-mark function. Vertical axis of the graph represents the best of fitness value and the horizontal axis represents corresponding iteration number as depicted in Fig. 4.2. Fig. 4.2a, 4.2b...4.2q, shows convergence trends of six algorithms under study for benchmark functions F1 to F17 respectively. It can be visualized from the figures that the proposed MDBO is converging with a faster rate for fourteen benchmark functions out of seventeen benchmark functions as compared to MVO, ES, GA, PBIL and PSO. However for function F11, PSO and GA has

Table 4.5: Wilcoxon test for statistically significance level at α = 0.05 on benchmark functions

Function	MDBO-ES		MDBO-PSO		MDBO-MVO		MDBO-PBIL		MDBO-GA	
	p-value	SGFNT	p-value	SGFNT	p-value	SGFNT	p-value	SGFNT	p-value	SGFNT
F_1	3.27E-06	+	3.27E-06	+	3.27E-06	+	3.27E-06	+	3.27E-06	+
F_2	9.07E-06	+	3.39E-06	+	9.07E-06	+	9.07E-06	+	9.07E-06	+
F_3	3.39E-06	+	3.39E-06	+	3.39E-06	+	3.39E-06	+	3.39E-06	+
F_4	3.31E-06	+	3.31E-06	+	3.31E-06	+	3.31E-06	+	3.31E-06	+
F_5	3.39E-06	+	3.39E-06	+	3.39E-06	+	3.39E-06	+	3.39E-06	+
F_6	0.167962	=	3.39E-06	+	0.167962	=	0.167962	=	0.167962	=
F_7	6.87E-07	+	6.87E-07	+	6.87E-07	+	6.87E-07	+	6.87E-07	+
F_8	3.39E-06	+	3.39E-06	+	3.39E-06	+	3.39E-06	+	3.39E-06	+
F_9	6.87E-07	+	6.87E-07	+	6.87E-07	+	6.87E-07	+	6.87E-07	+
F_{10}	3.39E-06	+	3.39E-06	+	3.39E-06	+	3.39E-06	+	3.39E-06	+
F_{11}	3.39E-06	+	3.39E-06	+	3.39E-06	+	3.39E-06	+	3.39E-06	+
F_{12}	3.39E-06	+	3.39E-06	+	3.39E-06	+	3.39E-06	+	3.39E-06	+
F_{13}	3.39E-06	+	3.39E-06	+	3.39E-06	+	3.39E-06	+	3.39E-06	+
F_{14}	3.38E-06	+	3.38E-06	+	3.38E-06	+	3.38E-06	+	3.38E-06	+
F_{15}	3.39E-06	+	3.39E-06	+	3.39E-06	+	3.39E-06	+	3.39E-06	+
F_{16}	1.26E-06	+	1.26E-06	+	1.26E-06	+	1.26E-06	+	1.26E-06	+
F_{17}	1.90E-06	+	1.90E-06	+	1.90E-06	+	1.90E-06	+	1.90E-06	+

Table 4.6: Results of the wilcoxon test for statistically significance level at α = 0.05 on multiple problems

Algorithms	p-value	SGFNT
MDBO-PBIL	0.001269	+
MDBO-PSO	0.000652	+
MDBO-GA	0.003338	+
MDBO-MVO	0.046603	+
MDBO-ES	0.000369	+

outperformed MDBO and for function F13, F17 GA and ES are showing faster convergence rate respectively. It can also be concluded that the proposed algorithm is beating eighty eight percent of the uni-model functions and seventy eight percent of the multi-model functions. The results shows that the proposed algorithm not only gives better fitness value but also shows good convergence rate for both the uni-modal and multi-modal functions.

4.4.1.4 Consistency Analysis

Box plots are used to show the consistency in the final result values found for each compared and proposed algorithm over fifteen runs. Fig. 4.4 depicts groups of final best solution values through their quartiles. Further extending vertical lines from the boxes indicates variability outside the upper and lower quartiles of final best solution for all algorithms under stuty/comparison. Fig. 4.4 contains seventeen sub-figures a, b, c...q for seventeen benchmark functions F1, F2...F17 respectively as given in Table 4.3. Fig. 4.4d, 4.4e, 4.4j and 4.4p shows minimum spacing between the different parts of boxes as compared to ES, PSO, MVO, PBIL and GA that indicates the degree of dispersion and skewness in the final best solution. And hence the proposed MDBO has outperformed ES, PSO, MVO, PBIL and GA algorithms for functions F4, F5, F10 and F16. The proposed MDBO has beat ES, PSO and PBIL and tie with MVO and GA as observed from Fig. 4.4c, 4.4h, 4.4k, 4.4m. Fig. 4.4i shows a tie for all algorithms under study. Further 4.4g and 4.4n depicts that MDBO beat ES and PBIL and tie with PSO, MVO and GA. Fig. 4.4b depicts that MDBO beat ES, PSO, PBIL and GA and tie MVO. Fig. 4.4q depicts MDBO beat ES, PSO and tie with MVO, PBIL and GA. The proposed MDBO is defeated by all algorithms under study as depicted in Fig. 4.4a and 4.4f. Further MDBO beat PSO, MVO, PBIL and GA except ES as shown in Fig. 4.4e.

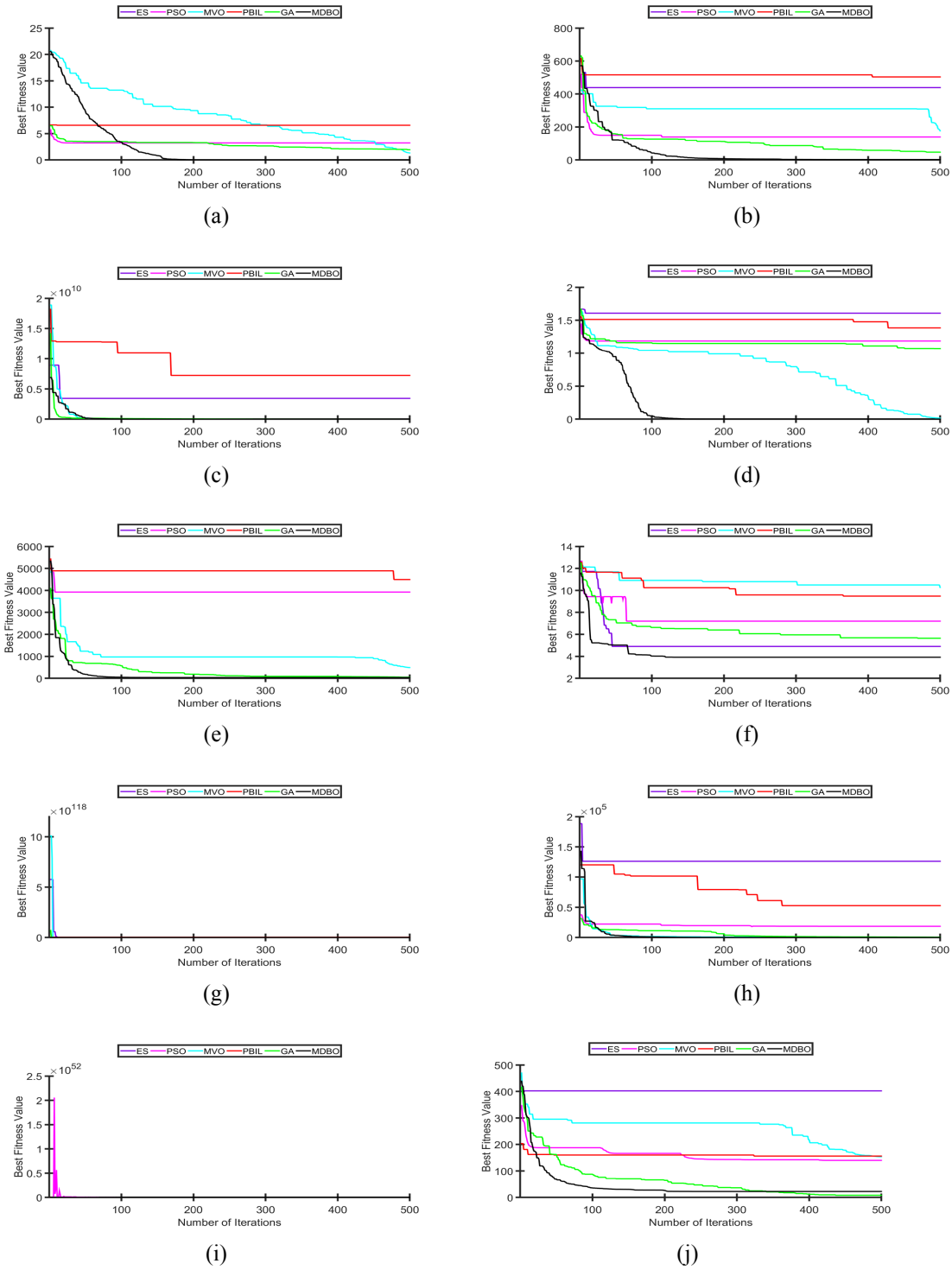


Figure 4.2: The convergence graphs for benchmark functions (a) Ackley, (b) Alpine, (c) Dixon and Price, (d) Griewank, (e) Levy, (f) Pathological, (g) Prem, (h) Powell, (i) PowellSum, and (j) Rastrigin

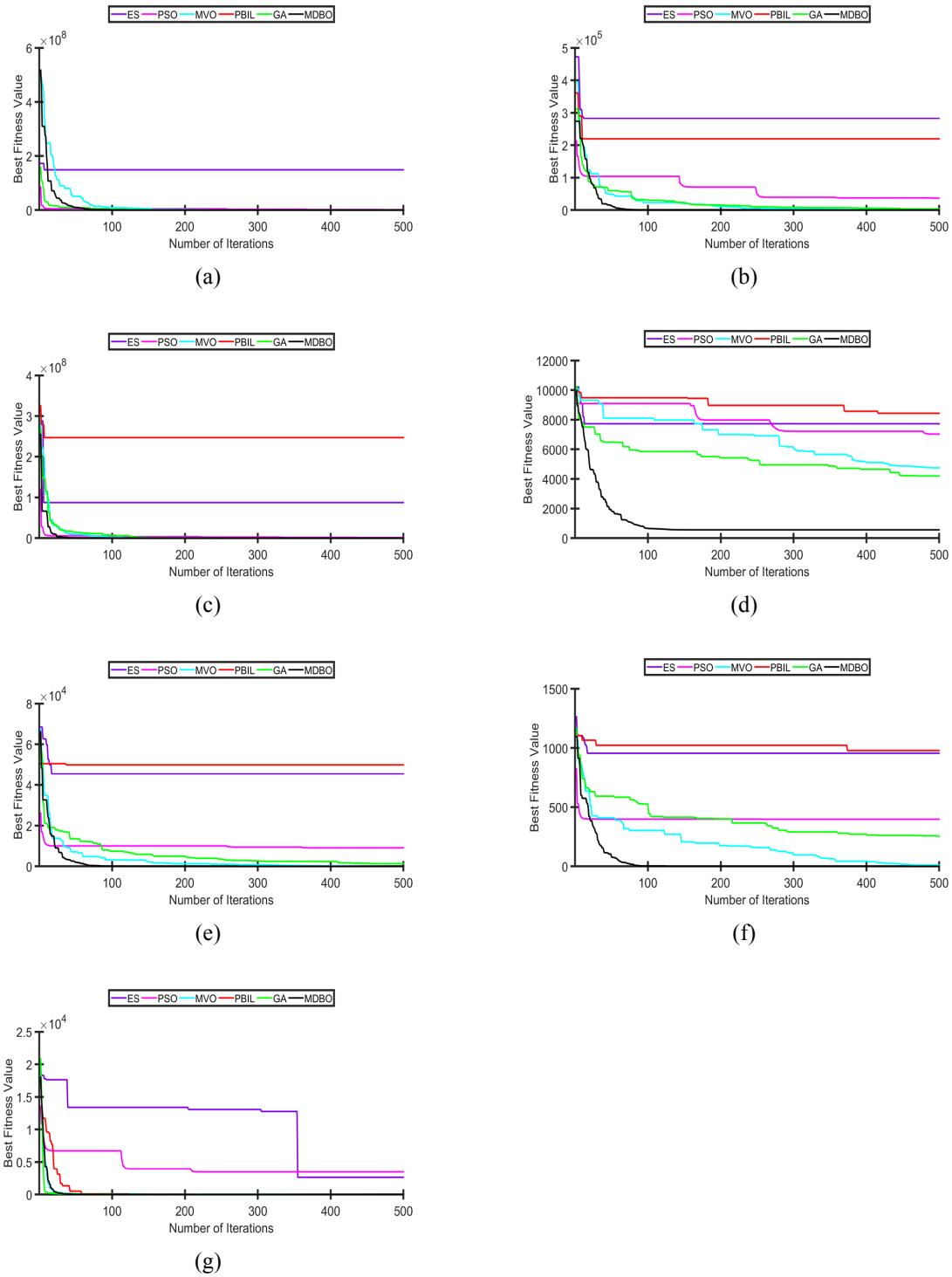


Figure 4.3: The convergence graphs for benchmark functions (a) Rosenbrock's, (b) Rotated Hyper-Ellipsoid, (c) Schumer Steiglitz, (d) Schwefel, (e) Sphere, (f) Step, and (g) Trigonometric

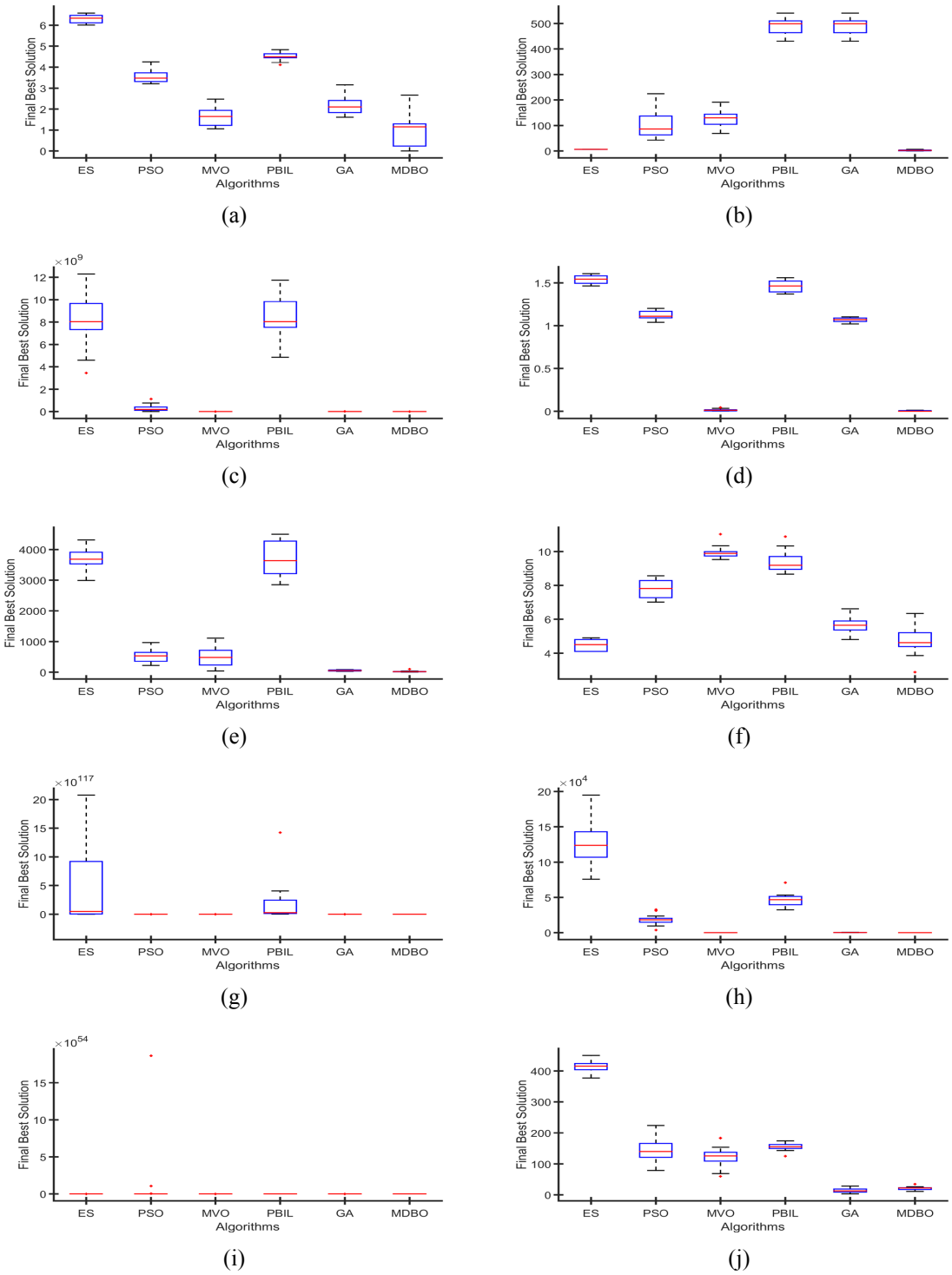


Figure 4.4: The box-plot graphs for benchmark functions(a) Ackley, (b) Alpine, (c) Dixon and Price, (d) Griewank, (e) Levy, (f) Pathological, (g) Prem, (h) Powell, (i) PowellSum, and (j) Rastrigin

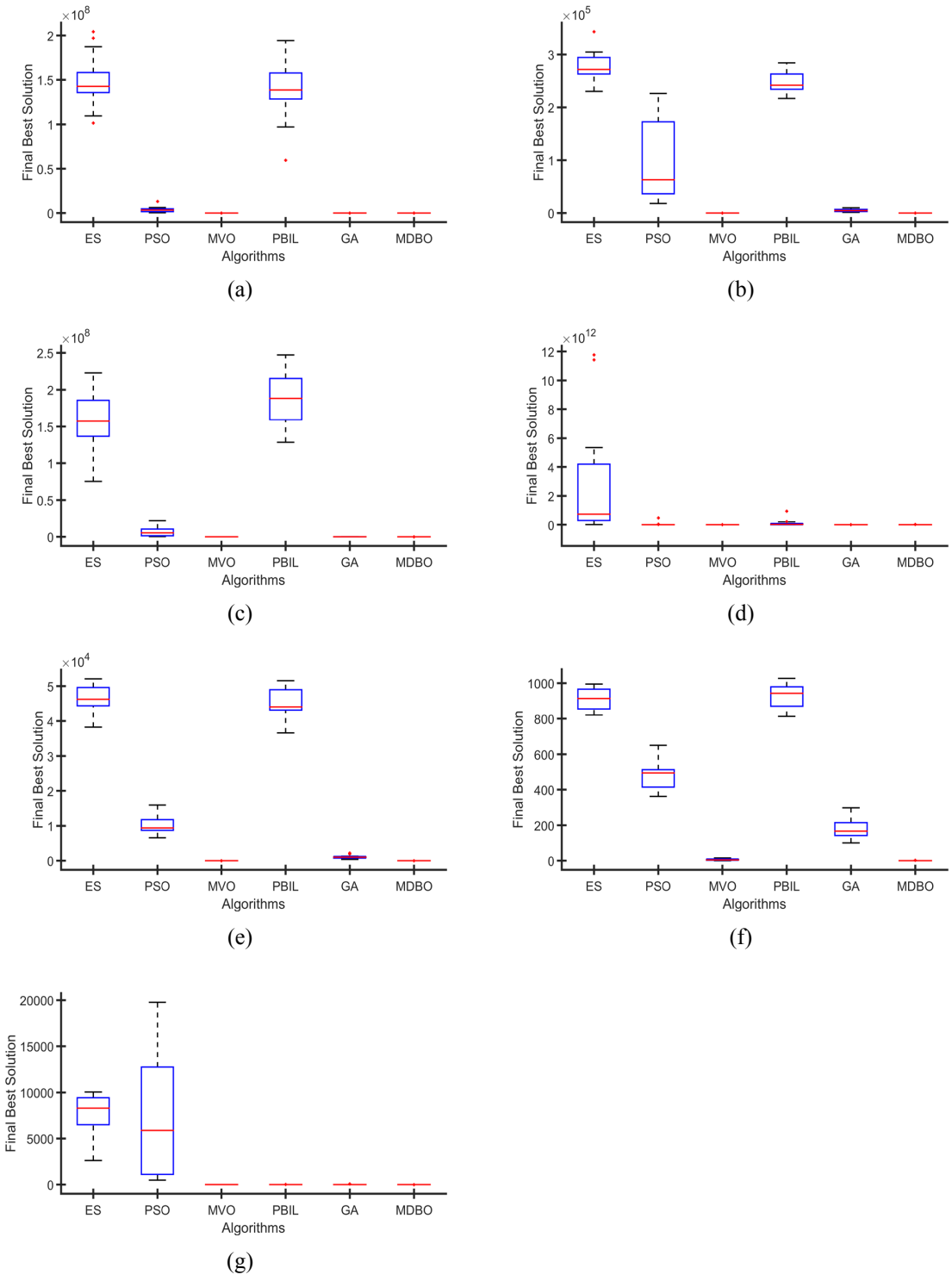


Figure 4.5: The box-plot graphs for benchmark functions (a) Rosenbrock's, (b) Rotated Hyper-Ellipsoid, (c) Schumer Steiglitz, (d) Schwefel, (e) Sphere, (f) Step, and (g) Trigonometric

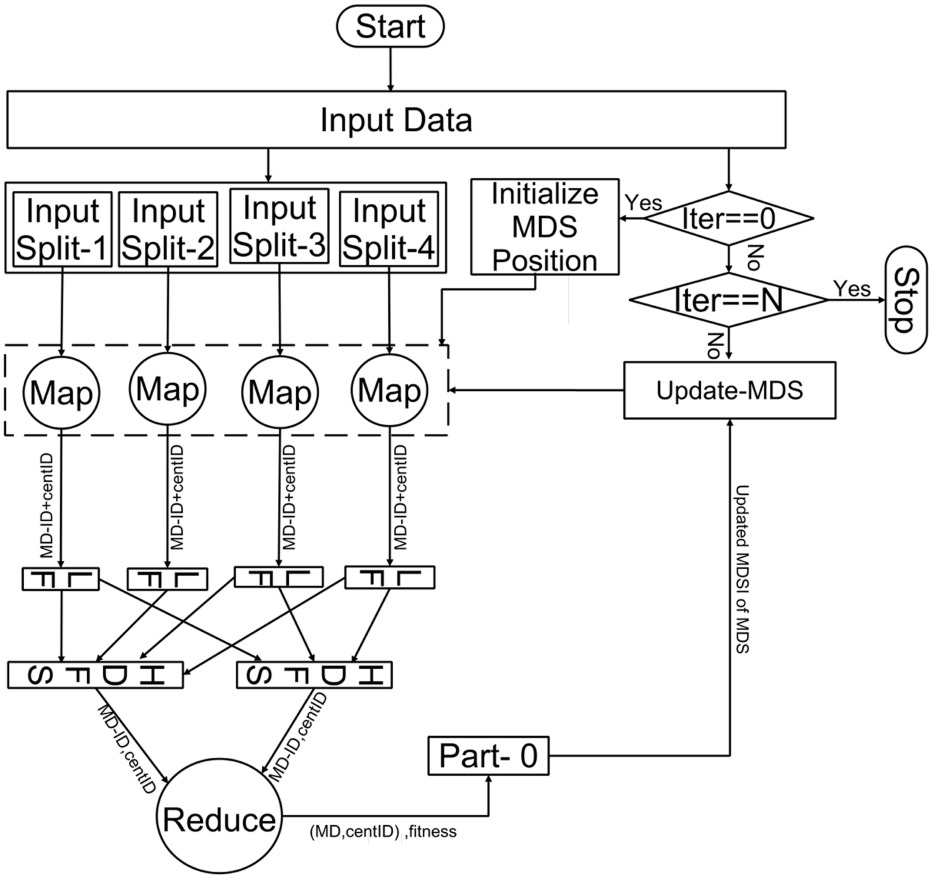


Figure 4.6: Model of parallel MDBO for data clustering

4.4.2 Performance Analysis on Data Clustering

The proposed MDBO algorithm is benchmarked on eight datasets taken from UCI repository and results are compared with K-means, PSO, GA, GSA, and GWO. Table 4.7 contains the details of eight benchmark datasets used for the study. The details of parameters, population size and maximum number of iteration have been provided in Table 3.8.

Table 4.8 presents the best and average of the intra-cluster distance of each algorithm over 30 independent runs. It can be seen from the Table 4.8 that the MDBO algorithm surpassed five compared algorithms for all the datasets in terms of best fitness value and

Table 4.7: Parameter values

Dataset	NOC	NOF	NOI
Iris	3	4	150
Glass	6	9	214
Seeds	3	7	210
Cancer	2	9	638
Yeast	8	10	1,448
Wine	3	13	178
Balance	3	4	625
Magic	2	10	19,020

NOC:Number of Clusters
NOF:Number of Features
NOI:Number of Instances

represented by bold fonts. However, for mean value MDBO outperformed all the algorithms on all the datasets except Iris and Balance for which GWO has given competitive results. In summary, it is pertinent from the results that the proposed MDBO algorithm is able to provide competitive results for the data clustering and hence it can be served as an alternative tool to solve the various real world clustering problems along with the optimization of the complex functions.

4.5 Parallel MDBO for Big Data Clustering

In this section, parallel model of MDBO algorithm using Hadoop MapReduce is explained. In MapReduce based MDBO, two main operations are performed, namely updating of the cluster centroid and the MDSI computation which represents the fitness of the clustering solution. In the proposed algorithm for data clustering the centroid of the clusters are updated according to MDBO with the aim of finding global optimum sum of the squared Euclidean distance between each data object and the respective cluster centroids. The complete MpReduce architecture of the MDBO for the clustering of large datasets is shown in Fig. 4.6. As shown in Fig., first the dataset is divided into smaller data blocks called input split. In the first iteration of the algorithm population of the MDBO is initialized and sup-

Table 4.8: Best and average fitness value over 30 runs

Dataset Name	**Criteria**	**K-Means**	**PSO**	**GSA**	**GA**	**GWO**	**MDBO**
Iris	Best	97.340	96.789	96.655	96.655	96.658	**96.655**
	Mean	106.334	97.136	96.675	99.530	**99.125**	96.655
Seeds	Best	587.319	312.683	311.798	311.798	311.882	**311.798**
	Mean	588.104	313.859	311.798	315.419	312.092	**311.803**
Glass	Best	292.757	238.511	286.118	205.703	265.814	**201.8051**
	Mean	325.547	257.065	316.710	264.104	302.041	**210.85**
Cancer	Best	19323.173	2969.239	2970.178	2964.387	2964.390	**2964.386**
	Mean	19323.176	2976.151	2994.779	3032.422	2964.394	**2965.799**
Balance	Best	3472.321	1423.967	1423.820	1424.043	1423.821	**1423.820**
	Mean	3493.800	1424.628	1424.515	1426.285	**1423.829**	1423.987
Yeast	Best	265.987	258.380	368.230	260.992	313.688	**245.268**
	Mean	333.333	227.832	378.028	284.128	338.613	**262.496**
Wine	Best	23706.687	16298.989	17038.592	16371.054	16307.092	**16292.184**
	Mean	24846.087	16305.117	17709.435	16865.723	16318.413	**16293.223**
Magic	Best	1,650,4.684	16230.271	18119.758	16234.079	16232.556	**16230.271**
	Mean	1,660,3.245	16230.332	19836.972	16239.088	16404.520	**16230.989**

plied to the each mapper running on the different nodes. In the clustering process using the MDBO algorithm the main computation intensive task is compute the sum of the squared Euclidean distance. The MapReduce model is thus employed to compute the fitness value. Each iteration of MDBO runs in two phases called MDBO-Map and MDBO-Reduce. In the proposed MapReduce based MDBO algorithm, the task of fitness computation is done in the MDBO-Map phase, in which parallelism is achieved since each machine have only some fraction of the whole dataset. As shown in in Algo 13, the MDBO-Map function starts with extracting the centroid of the cluster from the MDBO population which is stored in the HDFS. The MDBO-Map function then retrieves FSV vector of a military dog that represents location of the centroids. Further, the distance of centroid is calculated with each data object and minimum distance with its $centroid_ID$ is returned. The MDBO-Map function writes MD_ID, $centroid - ID$ having the minimum distance and the new key/value is computed from the minimum distance. After the completion of the MDO-Map phase, output of all the MDBO-Mappers are merged and grouped by keys. In the MDBO-reduce phase, the reduce function is called on each key,value pair generated by the MDBO-Map phase. The reducer function aggregates all the values with the identical key's to compute the fitness value that we aim to minimize. The main function of the MDBO-map is to decompose the task and that is merged by the MDBO-reduce phase. Finally, as shown in the Algo. algo:Reduce the reduce function computes the sum of the squared Euclidean distance between each data object and the respective cluster centroids. The newly computed fitness value is used to update the FSV of the military dog squad in the next iteration. The whole MapReduce cycle is repeated and this process continue until the stopping criterion or maximum iterations are not reached. Algo. 14 presents the Pseudo-code of the MDBO-reduce function.

Algorithm 13 :Map Function

Input: $Key - RecordID, Value - Record.$
Output: Key-MD-ID, Value-Minimum distance
Map (Key : dataId, Value : data)
MD-ID = read (file);
For each wolf;
MD-ID =read MD-ID;
Centroid =read centroids // FSV represents the location of centroids
Min-D= getMinD(record, Centroid); //getMinD returns centroid nearest to the record
centroid-ID = i //index of the centroid with minimum distance
new-key= MD-ID+centroid-ID;
end for
write (new-key, Min-D);

Algorithm 14 :Reduce Function

Input: $Key - (wolf - Id, centroid - ID), Value - (Min - Distance)$
Output: MD-ID, Value-MD-fitness // sum of intra-cluster distance
Reduce (Key:wolf-Id ,centroid-ID, Value-list: Min-Distance) // value contains list of distances of data point from their nearest centroids
Initialization
MD-fitness=0
For each distance in Min-Distance list
MD-fitness=sum-Dist+distance
end for
Update the positions of MD
emit(key, MD-fitness)

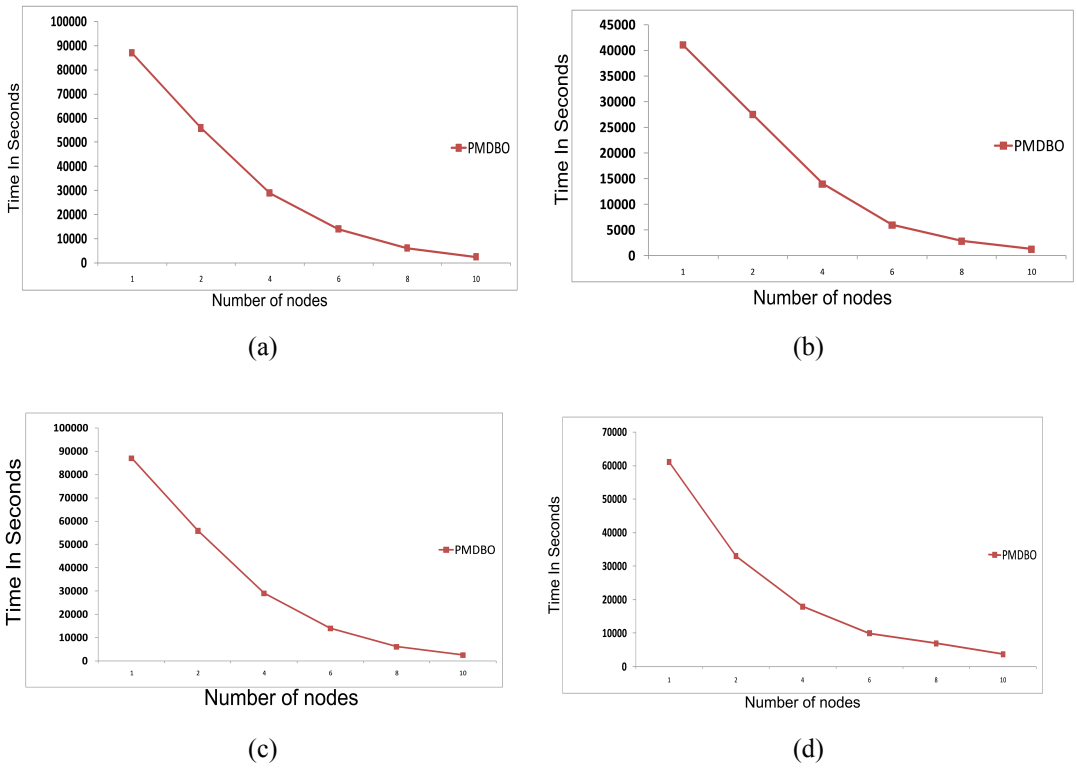


Figure 4.7: The speedup graph of (a) Pokerhand (b) Susy (c)Wine (d) CMC

Furthermore, the speedup performance of the MapReduce based MDBO is also studied. For the evaluation of speedup, four large size data sets have been used as described in Table 4.9. All the four datasets are taken from UCI repository in which Susy and Pokerhand are the original datasets while magic and wine are the synthetic datasets formed by duplicating each record of the original dataset 10^7 times. It is pertinent from the Fig. 4.7 that the running time of the MDBO decreases gradually with the increasing the number of nodes. It is also concluded from the speedup graph that the performance of parallel MDBO is close to ideal for the datasets large in size and having more number of features and clusters. Hence, it can be concluded that as the size and number of clusters, features of the dataset grew, the advantage of the proposed parallel MDBO are increased.

Table 4.9: Large Datasets

Name of data set	**Number of Clusters**	**Number of Features**	**Number of Data objects**
Pokerhand	10	10	1025010
Susy	2	18	5000000
Wine	3	7	10,000,050
CMC	3	9	10,000,197

4.6 Summery

This chapter has demonstrated a novel metaheuristic algorithm for the big data clustering. The proposed algorithm utilizes the searching capability of the suspected objects by the trained military dogs. MDBO uses the tendency of military dogs to communicate with each other and move towards the suspected object by their smell power. The proposed algorithm is validated on 17 benchmark functions. From the results, its is concluded that, the proposed algorithm outperforms the other algorithms in terms of mean fitness, standard deviations and convergence behavior. The convergence behavior and the consistency in the results of the MDBO has been validated using the Convergence graph and Box-plots curves respectively. Further, the quality of the results and the uniqueness of the MDBO has been also verified by the Wilcoxon rank sum test. It its observed that MDBO is different and better from the other considered algorithms in terms of consistency and convergence behavior.

Moreover, the speedup result of the parallel MDBO has been also presented to demonstrate the parallel performance effectiveness. Thus, it can be concluded that the proposed algorithm can be effectively used for solving the complex real world optimization problems.

Furthermore, the proposed algorithm opens up promising avenues of productive research in various big data and optimization based applications. The next chapter discusses the application of the proposed methods for solving the real world problems pertaining big data analysis. This chapter is a building block for mining the twitter sentiments and detecting the fake reviews with computational intelligence in big data environment.

Chapter 5

Real World Applications of Proposed Methods

This chapter presents the real world applications of the proposed methods pertaining big datasets. Two real world problems have been explored namely, twitter sentiment analysis and fake review detection. All the proposed methods are tested on both the problems and results are compared in terms of accuracy and computation time. Finally, the chapter is ended with conclusion and future application that can be further explored.

5.1 Overview

Social media and e-commerce based data analysis has been a big trend in the today's world. It has bought a revolution in the field of data science due to its high impact on the social and commercial importance. Recently, a number of organizations and industries are working in this field to extract the important facts from the ocean of data generated from the social media and e-commerce. Also as discussed in the literature review that, so many government

agencies and private industries are investing this field keeping in mind the future scope and importance. In this chapter two problems from the social media and e-commerce analysis have been explored namely, twitter sentiment analysis and fake review detection. Both the problems have been already studied in the literature for the small datasets. However, no method studied the effect of big data for twitter sentiment analysis and fake review detection. As there are billions of tweets and reviews are being generated daily. The big data solution is an area of interest for the industry and academia. Also, the majority of the work has been done using the supervised methods which required annotated data. However, the annotated data pertaining these problems is not easily available. Thus, this chapter introduced unsupervised methods to remedy this problem. Further, a detailed analysis in the results using the comparative performance of all the methods is provided. The rest of the chapter is organized in three sections. Section 5.2 presents the applicability of the proposed methods for solving twitter sentiment analysis problem in big data environment. The fake review detection problem has been explored in section 5.3. Section 5.4 demonstrate the conclusion and the observations of the work done.

5.2 Sentiment Analysis of Massive Twitter Datasets

Sentiment analysis is an eminent part of data mining for the investigation of user perception. Twitter is one of the popular social platform for expressing thoughts in the form of tweets. Nowadays, tweets are widely used for analyzing the sentiments of the users, and utilized for decision making purposes. Though clustering and classification methods are used for the twitter sentiment analysis, meta-heuristic based clustering methods has witnessed better performance due to subjective nature of tweets. However, sequential meta-heuristic based

clustering methods are computation intensive for large scale datasets. From last one decade, enormous growth in the digital data has been observed [236]. The social sites such as Instagram, Face-book, Twitter etc., are the major source of the digital data. Such huge availability of data has attracted the user-based industries to analyze the sentiments of users for making business strategies. The twitter has also great impact on the social and economic values. Figure 5.1 depicts the impact of tweets on the rice price.

Thus, efficient data mining methods are required for the sentiment analysis of the social media. Twitter, one of the popular social platform, provides a prodigious platform for the sentiment analysis. Twitter database has approximately 200 millions of users and nearby 400 million tweets are posted everyday. Often, user share their personal experiences about products or companies. Since, the maximum length of the tweet is 140 characters. Therefore, some short symbols like emoji are available for expressing the sentiments. The study of the tweets can deliver profound viewpoints and emotions about any subject [237]. Sentiment analysis methods are mainly classified into three categories namely, machine learning, lexicon and hybrid [238]. Lexicon-based methods require prior knowledge of sentiment lexicon to predict the sentiment. However, for the short-hand and emoji based texts, lexicon-based methods fail to perform well [239]. Avinash et al. [2] proposed a hybrid cuckoo search based method for Twitter sentiment analysis and concluded that emoticons are good predictors of the sentiments for short texts. Further, Canuto et al. [240] used the meta-level features for prediction of sentiments. Bravo et al. [241] proposed a supervised approach to amalgamate the strengths of polar words and sentiments for better analysis. Furthermore, Mohammad et al. [242] employed the supervised classifier to analyze emotion stimulus, emotion state, and intent of tweets for the US election. An ontology-based method was introduced for sentiment analysis of tweets by Kontopoulos et al. [243] where

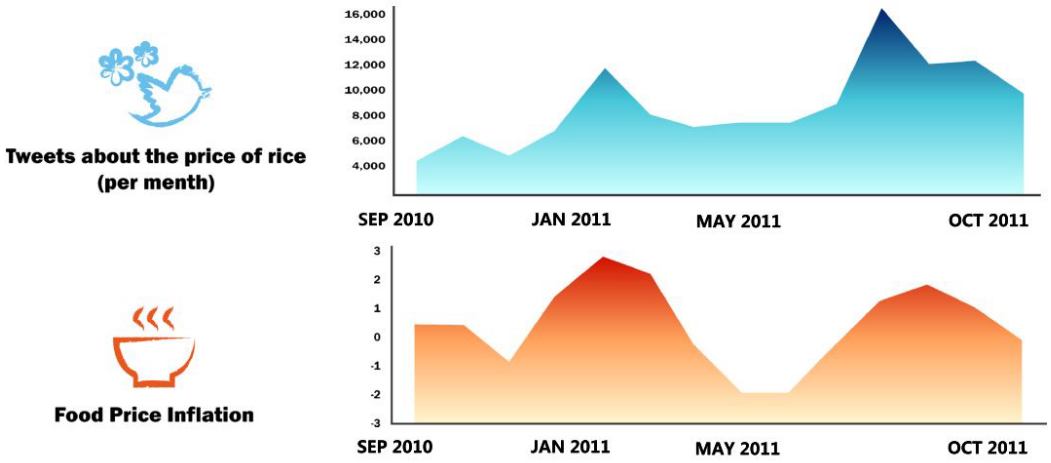


Figure 5.1: Twitter effect on rice price

a sentiment grade was allocated to each different perception in the tweet. Agarwal et al. [244] proposed a novel technique based on ConceptNet-based ontology for the sentiment analysis. Furthermore, SentiCircle method was introduced by Saif, et al. [245] in which context-specific polarity of words is determined. Qiu et al. [246] proposed a novel techniques which was based on semi-supervised classification. Further, Pandarachalil et al. [247] introduced a distributed unsupervised approach for extraction of the lexicons. Likewise, Fernndez et al. [248] proposed an unsupervised algorithm to predict the sentiment polarity of the informal texts using linguistic sentiment propagation model.

Recently, NLP is used to study the features and make classifiers more efficient [249]. Altnel et al. [250] introduced semantic smoothing kernels for extracting class specific semantics using vector space models (VSM). Muhammad et al. [251] proposed method using which used textual similarity metric for social media opinion mining. Further, Appel et al. [252] introduced hybridized approach based on NLP and fuzzy for the semantic polarity classification. Chen et al. [253] used the sequence modeling based neural network for the

text document sentiment prediction. Sulis et al. [254] studied the influence of figurative linguistic phenomena for distinguishing the tweets with the tags like irony, sarcasm using psycholinguistic and emotional features. Basari et al. [255] introduced hybrid PSO with SVM for classification of the movies. Likewise, Gupta et al. [256] performed aspect-based sentiment analysis using PSO-Asent method. However, the accuracy of PSO-Asent method on the unlabeled data is quite low. Further, Zhu et al. [257] developed a hybrid approach using GA and CRF for the prediction of sentiment. Meta-heuristic methods have shown good potential for clustering the sentiments of the twitter dataset [2]. However, on large scale datasets, these methods fail to perform in reasonable amount of time due to the sequential execution. For handling the computational complexities of large scale datasets, parallel and distributed computation is highly advantageous. Moreover, the majority of the methods present in the literature for the twitter sentiment analysis are proposed for the small datasets. This thesis has made a successful attempt to solve the sentiment analysis problem for the big data sets. All the proposed methods namely, DFBPKBA, MR-KBBO, MR-EGWO and MDBO have been utilized to unfold the twitter sentiment analysis problem. Moreover, the performance of proposed methods have been compared with three other state-of-the-art MapReduce based parallel methods namely, parallel K-means, parallel K-PSO, and MR-ABC in terms of mean of the accuracy and computation time.

5.2.1 Sentiment Analysis of Twitter via Proposed Methods

The sentiment analysis of the tweets is performed in three phases. First, the Tweets are preprocessed. Second, the feature are extracted. Third, the clustering is performed using the proposed methods.

5.2.1.1 Preprocessing

Before extracting the features, the raw tweets are prepossessed to remove useless and noisy words such as stop-words, URLs, fuzzy words etc. The proposed method used the following steps for the prepossessing:

- Remove URLS from the tweets by using regular expression. A regular expression denotes the pattern for strings or text. Generally, tt is used to search any specific textual or string pattern such as email address, URLs etc.
- ”@Username” is replaced with ”usr”.
- Multiple spaces are replaced with single space.
- Forward slash (/), backward slash (\) and all parenthesis are removed.
- # is removed from the hash-tags(#), for example #ashish is replaced with ashish and ”@” is removed from the user-name i.e @Username is replaced with usr.
- All the stop words such as the, a, is, as etc are removed. For this comparing them with stop word dictionary is used.
- All the words are converted to lower case.
- Sequence of repeated characters is removed for example ”hiiiiiiiiii” is replaced with ”hi”.
- The words not starting with alphabet are removed.

5.2.1.2 Feature extraction

In this phase, the features are extracted from the preprocessed twitter dataset. The tweets are transformed into feature vector matrix by extracting the features as given below.

1. **Word Length:** It is defined as the total word count of the tweets.
2. **Number of Positive Emoji:** It represents the count of symbols, such as :), ;), : D, etc., which are used to express positive emotions. For this, positive emoticon dictionary is used.
3. **Number of Negative Emoji:** It represents the count of symbols such as >: (, : (, :′ (, etc. These symbols are used by the users for expressing the negative emotions.
4. **Number of Neutral Emoji:** Neutral emoji is the straight-faced emoji which do not deliver any emotion. This feature is computed using neutral emoticon dictionary.
5. **Positive Exclamation:** The words, such as wow! , hurrah etc., are known as exclamatory words, these words are used to carry a very strong opinion/feeling about the message. Positive exclamation dictionary is used for computing this feature.
6. **Negative Exclamation:** Negative exclamations are the words, which express negative emotions of the users. This feature is counted by using negative exclamation dictionary.
7. **Negation:** Negation words such as no, not are used to express negative opinion.
8. **Positive word count:** The positive words such as confidence, achieve are counted using positive word dictionary.

9. **Negative word count:** This feature counts the sum of negative words such as poor, bad, worst etc., and it is found using dictionary.

10. **Neutral word count:** The words such as rarely, ok are treated as the neutral word and they does not provide any information about the emotion or opinion.

11. **Intense word count:** Intense words are used to make tweet more effective/intense, such as very, like or much. Intense word count is determined by matching the word with intense word dictionary.

5.2.1.3 Clustering

The complete procedure of the sentiment analysis is depicted in Figure 5.2. The process starts with the pre-processing of the tweets as discussed in the previous section.Thereafter, the clustering is performed using all the considered methods. Further, in the clustering process each tweet represents a vector of F where, F denotes the number of features. The value of each feature is scaled in the range [0, 1] to standardize the range feature variables. Table 5.1 describes the mean and standard deviation the considered dataset.

In the clustering using the proposed methods, the position each candidate solution represents a set of cluster centroids (C1, C2, C3... CK), where K represents the number of clusters. The objective function of the clustering is same as discussed in the previous chapter.

5.2.2 Performance Analysis

In this section, the accuracy and parallel performance of the proposed methods has been discussed. Four baseline datasets of the tweeter are used for testing the accuracy of the

Table 5.1: Mean and standard deviation of the extracted features

S.No	Dataset Name	Testdata.manual		Twitter-s1		Twitter-s2		Twitter avsh	
		Mean	STD	Mean	STD	Mean	STD	Mean	STD
1	$TotalCharacters$	0.452	0.220	0.560	0.208	0.540	0.201	0.353	0.171
2	$PositiveEmoji$	0.030	0.127	0.025	0.1564	0.011	0.077	0.012	0.094
3	$NegativeEmoji$	0.026	0.159	0.018	0.1359	0.007	0.083	0.012	0.084
4	$NeutralEmoji$	0.042	0.201	0.0009	0.0910	0.004	0.156	0.013	0.074
5	$PositiveWords$	0.006	0.077	0.014	0.1201	0.004	0.063	0.020	0.012
6	$NegativeWords$	0.008	0.089	0.022	0.1499	0.017	0.130	0.022	0.039
7	$NeutralWords$	0.065	0.24	0.055	0.1698	0.047	0.156	0.043	0.151
8	$IntenseWords$	0.115	0.246	0.050	0.1357	0.040	0.120	0.054	0.079
9	$PositiveExclamation$	0.024	0.087	0.043	0.1556	0.025	0.119	0.027	0.082
10	$NegativeExclamation$	0.110	0.210	0.012	0.1113	0.005	0.119	0.011	0.028
10	$NegativeExclamation$	0.022	0.093	0.062	0.2426	0.029	0.122	0.039	0.079

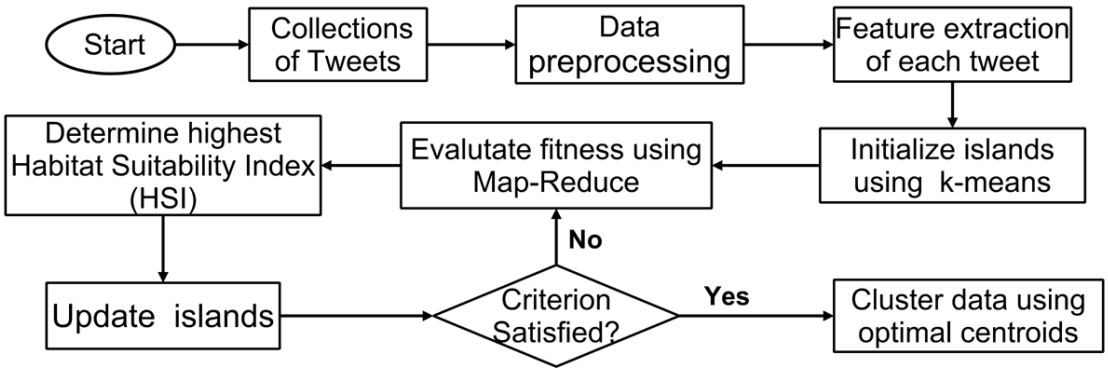


Figure 5.2: Complete flow of sentiment analysis process

proposed methods which are described as follows.

Testdata.manual.2009.06.14: This dataset is publicly available on Stanford Twitter corpus. The data set has 498 tweets having 182, 177, 139 positive, negative and neutral tweets respectively. The dataset is generated by collecting tweets based on topics like Obama, Chine, Iran, north Korea, Google, Nike, San Francisco, Kindle and San Francisco, in the duration from May 11, 2009 to Jul 14, 2009. The dataset is labeled as 0 for negative, 2 for neutral and 4 for positive sentiment.

Twitter-sanders-apple: This dataset have been collected from Oct 15, 2011 to Oct 20, 2011 by Sander Analytics on topics Apple, Google, Microsoft, and Twitter. The tweets were manually labeled by Niek Sanders in positive, negative or neutral polarity. Any tweet which contains positive indicator or topic is considered as positive. In this dataset, positive tweets are labeled by pos, negative tweets by neg, and neutral tweets were represented by neut. The complete dataset was divided into two parts as given below. **Twitter-sanders-apple2:** This dataset consists of 479 tweets out of which, 163 tweets are positive, 509 tweets are neutral and 316 tweets are negative.

Twitter-sanders-apple3: This dataset have 988 total tweets out of which 163 are positive, 509 are neutral and 316 are the negative tweets.

Twitter avsh: This dataset is collected and labeled by Pandey et al. . The dataset has total 2000 tweets posted from No. 17, 2014 to Dec 10, 2014. This dataset has 1000 positive tweets and 1000 negative tweets. The positive tweets were labeled by 0, while negative tweets were labeled by 1.

To evaluate the performance of the proposed methods on the large scale datasets, four

synthetic datasets sets are formed by duplicating each data instance of the baseline dataset by 10,000 times. Table 5.2 summarizes the features of the synthetic dataset used for the experiments. The parameter settings and running environment for all the considered methods are same as presented in chapter 3.

Further, the performance of the proposed methods is also compared with three other MapReduce based state-of-the-art methods namely, parallel K-Means, parallel K-PSO, and parallel MR-ABC, in terms of mean accuracy and mean computation time. Table 5.3 shows the mean of accuracy (MA) and computation time (MCT) obtained by running each method 30 times. The mean computation time is determined by running each method on cluster with 5 nodes.

It can be depicted from the table, that for the mean accuracy, MDBO has outperformed all the considered methods on three datasets namely, Twitter sander 2, Twitter avsh, and Testdata.manual, while for the one dataset i.e Twitter sander 3, MR-EGWO has given competitive results. However, the mean computation of the K-means is less as compared to meta-heuristic based methods, while the accuracy of the K-means is minimum among all the methods. Further, the MDBO is the second highest performer, in terms of computation time among all the considered methods and best among all the meta-heuristic based methods. Moreover, the MR-EGWO is the runner method in terms of mean accuracy. Thus, it can be concluded that MDBO can be used as an efficacious alternative for the twitter sentiment analysis of the large scale datasets.

Table 5.2: Parameter values

S.No	Dataset	NOC	NOI	Positive	Negative	Neutral
1	Twitter-avsh	2	20,000,000	1,000,000	1,000,000	–
2	Twitter-apple2	2	4,790,000	163,000	316,000	–
3	Twitter-apple3	3	9,880,000	163,000	316,000	509,000
4	Testdata-manual	6	4,980,000	182,000	182,000	139,000

NOC:Number of Clusters
NOF:Number of Features
NOI:Number of Instances

Table 5.3: Mean and standard deviation in the fitness value over 30 runs

S.No	Dataset	Criteria	parallel K-means	parallel K-PSO	MR-ABC	DFBPKBA	MR-KBBO	MR-EGWO	MDBO
1	Twitter avsh	MA	55.16	59.16	63.45	65.85	68.21	69.81	**71.89**
		MCT	**2.05E+04**	2.27E+04	2.25E+04	2.26E+04	2.25E+04	2.24E+04	2.25E+04
2	Twitter sander-2	MA	59.31	69.683	74.79	78.79	90.50	90.10	**91.20**
		MCT	**1.14E+E04**	1.54E+E04	1.53E+E04	1.52E+E04	1.52E+E04	1.41E+E04	1.41E+E04
3	Twitter sander-3	MCT	67.75	78.45	82.75	80.70	88.44	**91.88**	89.44
		MCT	**1.80E+04**	2.14E+04	2.13E+04	2.14E+04	2.13E+04	2.12E+04	2.12E+04
4	Testdata.manual	MA	62.21	72.55	72.45	74.75	88.80	91.80	**91.80**
		MCT	**1.04E+04**	1.28E+04	1.26E+04	1.25E+04	1.24E+04	1.21E+04	1.20E+04

5.3 Fake Review Detection of Online Reviews

Online reviews are increasingly used by the customers while purchasing a product or service. Over the time, there has been an increase in the number of fake reviews about the products or services due to the increasing competition among the e-commerce sites. Thus, fake review detection is an open and challenging problem. However, majority of research in this field is focused on sequential algorithms which provide inferior results when scaled on the big datasets. To mitigate this problem, this thesis introduces meta-heuristic based based approach to unfold the fake review detection problem in the big data environment. For the experimental analysis, the proposed methods have been studied on two standard fake review datasets and compared with three other state-of-the-art parallel methods in terms of accuracy and computation time. The results demonstrates that the proposed methods can

be effectively leveraged to solve the fake review detection problem.

With the increasing popularity and growth in the e-commerce, on-line reviews has become crucial asset for the customers in purchasing decision of the customer [258] [259]. Product or service review, given by the customer have great influence in the policy making of retailers and manufacturer. On-line reviews are helpful for the service providers in improving their services and products. Also customers finds some useful information from the opinion of the other customers in the form of reviews. Generally, customers are more interested in writing a review when they have exceptionally good or very bad experience. However, sometimes business owners pay someone to write good reviews about their products or write bad reviews about their competitors product. Therefore, sometime blind trust may be harmful for the customer and seller. In the last one decade, there has been a significant growth in the fake reviews. Fake reviews have have been written by the paid members that seems to be original [260][261] . For example, Samsung hired some students to defame the products of HTC. Similarity as reported as New York times fake reviews on i-Tunes was written on Amazon. Also fake reviews on Tripadvisor was reported by USA today in 2009. To, elucidate the above concern, a number of machine learning based models has been proposed in the literature. The machine learning based models have been seen to be more powerful as compared to humans for the detection of the fake reviews. However, the majority of the contemporary work has concentrated on supervised machine learning methods only, which need labeled data set for the training [262][263]. Ott et al. [264] have produced the first labeled dataset of gold slandered reviews by employing the crowd through Amazon Mechanical Turk(AMT). Again Ot et al. [265] studied n-gram based features using support vector machine for the classification of the fake reviews. Long et al. [266] used ontological features and stylometric features for the classification of the fake

reviews using the dataset produced by the Ott. In 2013, Mukhargee et al. [267] questioned the research based on the supervised machine learning techniques trained using AMT generated fake reviews. The author tested machine learning model of the Ott et al. [264] on Yelp data set and experienced much less accuracy as per the expectation. It has been observed that the behavioral features are the good predictors of the fake reviews as compared to the n-gram. In contrast to the supervised learning based methods, only few studies has been made on the unsupervised methods such as clustering [263]. Mukhargee et al. [267] modeled the spamicity of the reviewer by creating a margin, which separate the population distribution of the fake and real reviews cluster. Further, Rout et al. [268] have studied both supervised and unsupervised based methods and concluded that less work has been done using the unsupervised machine learning based methods.

Moreover, in recent years, nature inspired algorithm have caught the attention of researchers for handling the clustering problems due to their ability to find global best solution [269]. Further, a systematic review of the work done on fake review detection was performed by Crowford et al. [263]. The author studied that so far no Big Data analytic based technique has been witnessed in the literature for unfolding the fake review detection problem. For the big data sets, the sequential algorithms are not suitable as they can not handle the computational complexity of the massive data sets within the required time domain. Hence, parallel and distributed computation is required to study the big data sets. To abridge this gap, this thesis presents a novel fake review detection approach for the big datasets using the metaheuristic algorithm. All the methods proposed in the previous chapter have been tested on two benchmark datasets and compared with 3 other state-of the-art MapReduce based unsupervised methods introduced in the literature. The detailed procedure of the fake review detection is described in the following sections.

5.3.1 Fake Review Detection from Massive Datasets via Proposed Methods

The proposed method for fake review detection clusters the reviews in three phases; (i) Review are preprocessed, (ii) features are extracted, and (iii) clustering is performed. For the experimental analysis, two baseline datasets of the fake reviews have been used for testing. Total 12 features are extracted from each review based on the previous studies [268][262], which are illustrated in Table 5.4. The details of the dataset used in the study is described as follows.

Ott Dataset: This dataset was prepared by Ott et al. [265] and publicly available on his website. The dataset consists of 1600 reviews, out of which 800 reviews are the truth (400 with positive polarity and 400 with negative polarity) and extracted from Trip advisor, Expedite, Yelp and Hotels.com. The remaining 800 reviews are fake and collected from Amazon Mechanical Turk out of which 400 are fake with positive polarity and 400 reviews are fake with negative polarity.

Yelp Dataset: This is a real-life dataset extracted by from Yelp.com. Yelp has its own filtering algorithm to identify fake reviews and separates them into recommended and non recommended section. Total 12,8000 reviews from the yelp have been extracted out of which 60,000 are from non recommended (fake) section and rest are from the recommended section (true). The beautiful soup library of python for web scraping was used for crawling the reviews.

Table 5.4: Features taken for the clustering using parallel BBO

Feature Name	Description	Feature Name	Description
Word count	Total number of words in the review	Branding Frequency	Number of times brand name used
Verb Count	Number of verbs	Adjective Count	Number of adjectives used in the review
Spam Hit Score	Number of words similar to the Spam review	TruthHitScore	Number of words similar to the fake review
Diversity	Count of unique words	Adverb	Number of adverbs used
Noun	Number of nouns in the review	Proposition	Number of prepositions used
sentiment score	variation between sentiment polarity and rating of reviews	Average content similarity	It is defined as the average similarity in the reviews given by a single reviewer

5.3.2 Performance Analysis

In this section, the performance of the proposed methods for fake review detection is presented. Two datasets have been used to for performing the experiments which are detailed in section 5.3.1. The parameter values and the running environment of all the methods is same as presented in the previous chapter.

Further, the performance of the proposed methods for fake review detection is compared with four other MapReduce based state-of-the-art methods namely parallel K-Means, parallel Black hole, parallel K-PSO and parallel MR-ABC in terms of mean accuracy (MA) and mean computation time (MCT). The mean of accuracy and mean computation time is obtained by running each method 30 times is presented in table 5.5. Moreover, the mean computation time of each method is computed by running on cluster of 5 computers.

From the table, it can be observed that the MDBO has surpassed all the other considered methods in terms of accuracy in both the datasets. Moreover, MR-EGWO is the second best performer in terms of the mean accuracy and the MR-KBBO has performed better than the DFPKBA, BH and ABC. However, the mean computation time of MDBO is least on Ott dataset while MR-EGWO has minimum computation time on Yelp dataset.

Table 5.5: Best and average fitness value over 30 runs

Dataset	Criteria	BH	K-PSO	ABC	DFPKBA	MR-KBBO	MR-EGWO	MDBO
Ott dataset	Best	58.46	54.68	59.50	63.92	71.52	71.67	**72.14**
	MCT	805.60	816.40	884.60	876.50	776.50	775.50	**768.50**
Yelp review	Best	58.46	54.68	59.50	65.92	73.52	73.52	**74.52**
	MCT	4141.50	3980.90	4005.98	4051.20	3940.47	3850.47	**3875.47**

5.4 Summery

This chapter presented the two prominent applications of the proposed methods for solving big data problems. First, the problem of mining twitter sentiments from the large scale data has been unfolded. The accuracy in the results has been tested on four big datasets and compared against seven MapReduce based state-of-the-art methods namely, K-means, K-PSO, MR-ABC, DFBPKBA, MR-KBBO, MR-EGWO and MDBO. It is concluded from the results that MDBO has outperformed all the methods on the majority of the data sets while MR-EGWO is the runner method. Secondly, the proposed methods have been applied for identifying the fake reviews. Two authentic and publicly available data sets have been used for testing the validity of the proposed methods. The accuracy in the results has been compared with MapReduce based state-of-the-art methods namely, K-PSO, ABC, BH, DFPKBA, MR-KBBO, MR-EGWO and MDBO. The experimental results demonstrate that MDBO has outperformed all the other considered methods for identifying the fake reviews. Thus, it is concluded that the proposed methods are the powerful alternatives for solving the various real world application pertaining big datasets.

Chapter 6

Conclusion and Future Scope

In this chapter the major conclusions of the thesis works are highlighted. The future possibilities and scopes of the big data analysis is also explored using the developed model and methods.

6.1 Conclusion

This work presents the efficient data analysis methods by leveraging the strengths meta-heuristic algorithms. The meta-heuristic algorithms have been found to be better than the conventional methods like K-means and FCM for clustering analysis. However, these algorithms are computational complex and fails for the large scale data sets. This thesis made a successful attempt to adopt the meta-heuristic based methods for the big data analysis by distributed and parallel processing. Four novel methods have been developed and applied for solving real world problems. All the methods are independent but related to each other. Further, the proposed methods has been applied to solve two real world problems.

Some of the key finding are listed as follows.

1. Nature inspired meta-heuristic algorithms have been widely used for the partitional clustering, since it is a NP hard problem. However, these algorithms have not been used for the big data analysis due their high computational cost. Chapter 1 has formulated the problem of developing novel methods using nature inspired algorithms which will work efficiently for the big data analysis.

2. A parallel and distributed meta-heuristic computation model has been presented for handling the computation intensive problems. The presented model can be easily adopted for the parallelization of meta-heuristic algorithm to handle the large scale datasets. The proposed model is based on the MapReduce architecture, which works on an open source platform and uses commodity hardware. Thus, it is affordable and easy to use for the industry and academia.

3. A MapReduce based bat algorithm has been developed for clustering the large scale datasets. The proposed algorithm performed well for handling the massive datasets. Bat algorithm is simple, quick and easy to implement. However, there trade off between the exploration and exploitation is not balanced and convergence becomes slow for the computation intensive problems.

4. To overcome the above mentioned limitations a novel hybrid method has been developed based on the K-means and Bat algorithm. The population of proposed method was initialized using ten iterations of the K-means to make convergence fast. Further, the frequency of the bats were changed dynamically to make better exploration at the initial stage followed extensive exploitation at the later stage. The proposed hybrid model was tested on 5 benchmark datasets and compared with K-means, PSO and BA. The hybrid model has outperformed K-means and BA on all the datasets.

However, for one dataset PSO has given best results. In addition, the parallel performance of the proposed method has been tested on 4 datasets in which 2 datasets were small and two datasets were large in size. The speedup performance for the large scale datasets was almost linear. However, for small datasets the speedup performance was not as expected due to hidden IO cost. Thus, it has been concluded from the experimental results that the proposed parallel model is well suited for the large scale datasets.

5. It has been found that enhanced version of the nature inspired techniques have outperformed the parent algorithms in the majority of application. An enhanced grey wolf optimizer based method has been developed and compared with the five existing algorithms on seven datasets. The proposed method has outperformed all the considered methods on all the datasets in terms best fitness value. For mean fitness value, EGWO has surpassed results for wine, seeds, glass and cancer. However, GWO has competitive results on Iris and Balance datasets while PSO performed well on Haberman dataset.

6. It has been also established that, the EGWO has better convergence as compared to the considered methods. The convergence behavior has been validated by plotting the convergence graph. Further, the Boxplots are analyzed to study the consistency in the results of the proposed method. It was observed that the proposed method is able to provide the consistent results in different runs. Additionally, the uniqueness of the EGWO was validated using a non parametric test named wilcoxon rank sum test. It was affirmed from the test that the proposed method is different from the other methods.

7. The proposed EGWO has been adopted on MapReduce architecture and named MR-EGWO. The proposed MR-EGWO outperformed four state-of-the-arts MapReduce based clustering methods in terms of F-measure, when tested on four large scale datasets. Furthermore, the speedup efficiency of the MR-EGWO is also analyzed on two large scale datasets by running them in hadoop cluster with different nodes. It was concluded that, MR-EGWO is well suited for analyzing large datasets with significant speedup performance and better clustering quality. Thus, it is concluded that MR-EGWO can perform efficiently while analyzing the large scale datasets.

8. It has been also found that MR-EGWO has performed better than DFBPKBA and MR-KBBO in terms of F-measure and computation time. However the K-means algorithm has least computation time among all the population based methods, but at the same time it was also given poor performance as compared to the population based algorithms. Thus, it is vindicated from the experimental results that performance of swarm based methods is better as compared to evolutionary algorithms.

9. A novel swarm based optimization algorithm, which mimics the searching behavior of the Military dogs has been introduced. The proposed algorithm has been tested on 17 benchmark functions and results are presented in terms mean fitness, standard deviations and convergence behavior. The Box-plots are drawn for each function to validate the consistency in the results. The uniqueness in the results of the MDBO has been also validated using the Wilcoxon rank sum test. The results show that the proposed algorithm can be successfully applied for solving various real world optimization problem.

10. The MDBO has been adopted on the MapReduce model to handle big datasets. Fur-

ther, the performance of the MDBO has been analyzed and compared with DFBP-KBA, MR-KBBO, and MR-EGWO in terms of F-measure and speedup. It is observed that MDBO has outperformed all the other considered methods in terms of F-measure and computation time.

11. The proposed methods are leveraged to solve, twitter sentiment analysis problem on large scale dataset. Four large scale datasets are chosen to perform the experiments. The results show that, MDBO has outperformed other methods on 75% of the datasets with the highest accuracy of 91.80% on the Testdata.manual. However, for one dataset i.e. Twitter sander-3, MR-EGWO has given competitive results.

12. Further, the problem of fake review detection in large scale dataets has been unfolded using the proposed methods. Two large scale datasets have been used for performing the experiments. The results indicate that MDBO produces better results among the considered methods. In addition, MDBO has also passed the wilcoxon rank some test on all the the datasets. Also, the mean computation time of the MDBO is minimum among all the considered methods while for one dataset MDBO is runner method in terms of MCT.

13. Thus, it is pertinent from the experiments that proposed methods can efficiently used for solving the complex real world optimization problems where the conventional algorithm fails. Thus, the goal of developing parallel and distributed meta-heuristic based methods for the big data analysis has been successfully attempted in this thesis.

6.2 Future Scope

The development of the parallel meta-heuristic based methods and their applications for solving the real world problems of big data analysis present in this thesis are viewed as beginning mark in this area. Thus, the opportunities of further work in this area is vast. Following are some of the important area for the further extension of the work.

1. A parallel and distributed meta-heuristic model for the data clustering has been presented using the hadoop and MapReduce. Further models for the parallelization of these algorithms can be explored and other tools such as spark may be tested to improve the computation time.

2. The methods proposed and applied in this work can be further extended and validated with more real-world datasets for the decision making.

3. The hybrid method developed in this work can also be applied for the other big data analysis problem such as block chain, recommender system and health care for the better decision making.

4. The proposed extended model, EGWO may be further tested on some real world problems and its strengths can be utilized for solving the contemporary big data analysis problems.

5. Some more algorithms may be explored for the big data analysis. The hybridization of the proposed MDBO may be explored with other algorithms such as K-meas, EGWO and other contemporary techniques to improve the clustering process. Furthermore, some other tools may be explored for the parallelization to reduce the computation cost of the existing algorithm.

Bibliography

[1] A. C. Pandey, R. Pal, and A. Kulhari, "Unsupervised data classification using improved biogeography based optimization," *International Journal of System Assurance Engineering and Management*, vol. 9, pp. 1–9, 2018.

[2] A. C. Pandey, D. S. Rajpoot, and M. Saraswat, "Twitter sentiment analysis using hybrid cuckoo search method," *Information Processing & Management*, vol. 53, pp. 764–779, 2017.

[3] K. LIN, L. XU, and J. WU, "A fast fuzzy c-means clustering for color image segmentation," *Journal of Image and Graphics*, vol. 2, pp. 05–10, 2004.

[4] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Transactions on neural networks*, vol. 16, pp. 645–678, 2005.

[5] M. J. Meena, K. Chandran, A. Karthik, and A. V. Samuel, "An enhanced aco algorithm to select features for text categorization and its parallelization," *Expert Systems with Applications*, vol. 39, pp. 5861–5871, 2012.

[6] P. Shelokar, V. K. Jayaraman, and B. D. Kulkarni, "An ant colony approach for clustering," *Analytica Chimica Acta*, vol. 509, pp. 187–195, 2004.

[7] D. Karaboga and C. Ozturk, "A novel clustering approach: Artificial bee colony (abc) algorithm," *Applied soft computing*, vol. 11, pp. 652–657, 2011.

[8] A. Hatamlou, S. Abdullah, and H. Nezamabadi-Pour, "A combined approach for clustering based on k-means and gravitational search algorithms," *Swarm and Evolutionary Computation*, vol. 6, pp. 47–52, 2012.

[9] U. Maulik and S. Bandyopadhyay, "Genetic algorithm-based clustering technique," *Pattern recognition*, vol. 33, pp. 1455–1465, 2000.

[10] V. Kumar, J. K. Chhabra, and D. Kumar, "Grey wolf algorithm-based clustering technique," *Journal of Intelligent Systems*, vol. 26, pp. 153–168, 2017.

[11] S. Zhang and Y. Zhou, "Grey wolf optimizer based on powell local optimization method for clustering analysis," *Discrete Dynamics in Nature and Society*, 2015.

[12] S. Alam, G. Dobbie, and P. Riddle, "Particle swarm optimization based clustering of web usage data," in *Proc. of International Conference on Web Intelligence and Intelligent Agent Technology*, 2008, pp. 451–454.

[13] Y.-J. Gong, W.-N. Chen, Z.-H. Zhan, J. Zhang, Y. Li, Q. Zhang, and J.-J. Li, "Distributed evolutionary algorithms and their models: A survey of the state-of-the-art," *Applied Soft Computing*, vol. 34, pp. 286–300, 2015.

[14] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, "The hadoop distributed file system," in *Proc. of International Conference on mass storage systems and technologies*, 2010, pp. 1–10.

[15] "Frontpage - hadoop wiki," https://wiki.apache.org/hadoop/, (Accessed on 02/27/2018).

[16] S. N. Khezr and N. J. Navimipour, "Mapreduce and its application in optimization algorithms: A comprehensive study," *Majlesi Journal of Multimedia Processing*, vol. 4, pp. 35–39, 2015.

[17] J. Wang, D. Yuan, and M. Jiang, "Parallel k-pso based on mapreduce," in *Proc. of International Conference on Communication Technology*, 2012.

[18] C.-Y. Lin, Y.-M. Pai, K.-H. Tsai, C. H.-P. Wen, and L.-C. Wang, "Parallelizing modified cuckoo search on mapreduce architecture," *Journal of Electronic Science and Technology*, vol. 11, pp. 115–123, 2013.

[19] M. A. Beyer and D. Laney, "The importance of □big data□: a definition," *Stamford, CT: Gartner*, vol. 8, pp. 2014–2018, 2012.

[20] K. Armstrong, "Big data: a revolution that will transform how we live, work, and think," *Information, Communication & Society*, vol. 17, pp. 1300–13 002, 2014.

[21] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. H. Byers, "Big data: The next frontier for innovation, competition, and productivity," 2011.

[22] M. Chen, S. Mao, and Y. Liu, "Big data: A survey," *Mobile networks and applications*, vol. 19, pp. 171–209, 2014.

[23] D. Laney, "3d data management: Controlling data volume, velocity and variety," *META group research note*, vol. 6, pp. 1–5, 2001.

[24] P. Zikopoulos, C. Eaton *et al.*, *Understanding big data: Analytics for enterprise class hadoop and streaming data*. McGraw-Hill Osborne Media, 2011.

[25] E. Meijer, “The world according to linq,” *Communications of the ACM*, vol. 54, pp. 45–51, 2011.

[26] L. Wang, J. Zhan, C. Luo, Y. Zhu, Q. Yang, Y. He, W. Gao, Z. Jia, Y. Shi, S. Zhang *et al.*, “Bigdatabench: A big data benchmark suite from internet services,” in *Proc. of Internation Conference on High Performance Computer Architecture*, 2014, pp. 488–499.

[27] J. Gantz and D. Reinsel, “Extracting value from chaos,” *IDC iview*, vol. 1142, pp. 1–12, 2011.

[28] D. DeWitt and J. Gray, “Parallel database systems: the future of high performance database systems,” *Communications of the ACM*, vol. 35, pp. 85–98, 1992.

[29] T. Walter, “Teradata past, present, and future,” *UCI ISG lecture series on scalable data management*, vol. 1, pp. 44–48, 2009.

[30] S. Ghemawat, H. Gobioff, and S.-T. Leung, *The Google file system*. ACM, 2003, vol. 37, no. 5.

[31] T. Hey, S. Tansley, K. M. Tolle *et al.*, *The fourth paradigm: data-intensive scientific discovery*. Microsoft research Redmond, WA, 2009.

[32] “Opinion | a.i. and big data could power a new war on poverty - the new york times,” https://www.nytimes.com/2018/01/01/opinion/ai-and-big-data-could-power-a-new-war-on-poverty.html, (Accessed on 07/03/2018).

[33] “Drowning in numbers - big data,” https://www.economist.com/graphic-detail/2011/11/18/drowning-in-numbers, (Accessed on 07/03/2018).

[34] "Specials : Nature," https://www.nature.com/collections/wwymlhxvfs, (Accessed on 07/03/2018).

[35] R. Cattell, "Scalable sql and nosql data stores," *Acm Sigmod Record*, vol. 39, pp. 12–27, 2011.

[36] J. H. Howard, M. L. Kazar, S. G. Menees, D. A. Nichols, M. Satyanarayanan, R. N. Sidebotham, and M. J. West, "Scale and performance in a distributed file system," *ACM Transactions on Computer Systems (TOCS)*, vol. 6, pp. 51–81, 1988.

[37] C. D. Manning, C. D. Manning, and H. Schütze, *Foundations of statistical natural language processing*. MIT press, 1999.

[38] S. K. Pal, V. Talwar, and P. Mitra, "Web mining in soft computing framework: relevance, state of the art and future directions," *IEEE Transactions on Neural Networks*, vol. 13, pp. 1163–1177, 2002.

[39] C. C. Aggarwal and H. Wang, "Text mining in social networks," in *Proc. of International Conference on Social network data analytics*, 2011, pp. 353–378.

[40] W. Chen, Y. Wang, and S. Yang, "Efficient influence maximization in social networks," in *Proc. of 15th International Conference on Knowledge discovery and data mining*, 2009, pp. 199–208.

[41] "Google play store: number of apps 2018 | statistic," https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/, (Accessed on 11/01/2018).

[42] J. Hagerty, R. L. Sallam, and J. Richardson, "Magic quadrant for business intelligence platforms," *Gartner for Business Leaders (February 6, 2012)*, vol. 10, pp. 13–17, 2012.

[43] A. Prugel-Bennett, "When a genetic algorithm outperforms hill-climbing," *Theoratical Computer Science*, vol. 320, pp. 135–153, 2004.

[44] S. Bandyopadhyay, S. Saha, U. Maulik, and K. Deb, "A simulated annealing-based multiobjective optimization algorithm: Amosa," *IEEE transactions on evolutionary computation*, vol. 12, pp. 269–283, 2008.

[45] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. of International Conference on Neural Networks*, 1995.

[46] A. H. Gandomi and A. H. Alavi, "Krill herd: a new bio-inspired optimization algorithm," *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, pp. 4831–4845, 2012.

[47] G.-G. Wang, A. H. Gandomi, A. H. Alavi, and G.-S. Hao, "Hybrid krill herd algorithm with differential evolution for global numerical optimization," *Neural Computing and Applications*, vol. 25, pp. 297–308, 2014.

[48] M. Dorigo and M. Birattari, "Ant colony optimization," in *Encyclopedia of machine learning*. Springer, 2011, pp. 36–39.

[49] J. C. Bansal, H. Sharma, S. S. Jadon, and M. Clerc, "Spider monkey optimization algorithm for numerical optimization," *Memetic computing*, vol. 6, no. 1, pp. 31–47, 2014.

[50] P.-W. Tsai, J. Zhang, S. Zhang, V. Istanda, L.-C. Liao, and J.-S. Pan, "Improving swarm intelligence accuracy with cosine functions for evolved bat algorithm."

[51] K. Sharma, V. Chhamunya, P. Gupta, H. Sharma, and J. C. Bansal, "Fitness based particle swarm optimization," *International Journal of System Assurance Engineering and Management*, vol. 6, no. 3, pp. 319–329, 2015.

[52] S. S. Jadon, J. C. Bansal, R. Tiwari, and H. Sharma, "Artificial bee colony algorithm with global and local neighborhoods," *International Journal of System Assurance Engineering and Management*, pp. 1–13, 2014.

[53] D. Dasgupta and Z. Michalewicz, *Evolutionary algorithms in engineering applications*. Springer Science & Business Media, 2013.

[54] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization*, vol. 11, pp. 341–359, 1997.

[55] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "Gsa: a gravitational search algorithm," *Information sciences*, vol. 179, pp. 2232–2248, 2009.

[56] H. Shah-Hosseini, "The intelligent water drops algorithm: a nature-inspired swarm-based optimization algorithm," *International Journal of Bio-Inspired Computation*, vol. 1, pp. 71–79, 2009.

[57] Ş. İ. Birbil and S.-C. Fang, "An electromagnetism-like mechanism for global optimization," *Journal of global optimization*, vol. 25, pp. 263–282, 2003.

[58] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-verse optimizer: a nature-inspired algorithm for global optimization," *Neural Computing and Applications*, vol. 27, pp. 495–513, 2016.

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

[60] A. Hatamlou, "Black hole: A new heuristic optimization approach for data clustering," *Information sciences*, vol. 222, pp. 175–184, 2013.

[61] H. Du, X. Wu, and J. Zhuang, "Small-world optimization algorithm for function optimization," in *Proc. of International Conference on Natural Computation*, 2006.

[62] A. Kaveh and M. Khayatazad, "A new meta-heuristic method: ray optimization," *Computers & structures*, vol. 112, pp. 283–294, 2012.

[63] F. F. Moghaddam, R. F. Moghaddam, and M. Cheriet, "Curved space optimization: a random search based on general relativity theory," *arXiv preprint arXiv:1208.2214*, 2012.

[64] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

[65] E. Emary, H. M. Zawbaa, and A. E. Hassanien, "Binary grey wolf optimization approaches for feature selection," *Neurocomputing*, vol. 172, pp. 371–381, 2016.

[66] G. Komaki and V. Kayvanfar, "Grey wolf optimizer algorithm for the two-stage assembly flow shop scheduling problem with release time," *Journal of Computational Science*, vol. 8, pp. 109–120, 2015.

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

[68] S. A. Medjahed, T. A. Saadi, A. Benyettou, and M. Ouali, "Gray wolf optimizer for hyperspectral band selection," *Applied Soft Computing*, vol. 40, pp. 178–186, 2016.

[69] Y.-C. Ho and D. L. Pepyne, "Simple explanation of the no-free-lunch theorem and its implications," *Journal of optimization theory and applications*, vol. 115, pp. 549–570, 2002.

[70] M. Shakarami and I. F. Davoudkhani, "Wide-area power system stabilizer design based on grey wolf optimization algorithm considering the time delay," *Electric Power Systems Research*, vol. 133, pp. 149–159, 2016.

[71] T. Jayabarathi, T. Raghunathan, B. Adarsh, and P. N. Suganthan, "Economic dispatch using hybrid grey wolf optimizer," *Energy*, vol. 111, pp. 630–641, 2016.

[72] D. Guha, P. K. Roy, and S. Banerjee, "Load frequency control of interconnected power system using grey wolf optimization," *Swarm and Evolutionary Computation*, vol. 27, pp. 97–115, 2016.

[73] X. Song, L. Tang, S. Zhao, X. Zhang, L. Li, J. Huang, and W. Cai, "Grey wolf optimizer for parameter estimation in surface waves," *Soil Dynamics and Earthquake Engineering*, vol. 75, pp. 147–157, 2015.

[74] S. Mirjalili, "How effective is the grey wolf optimizer in training multi-layer perceptrons," *Applied Intelligence*, vol. 43, pp. 150–161, 2015.

[75] S. Amirsadri, S. J. Mousavirad, and H. Ebrahimpour-Komleh, "A levy flight-based grey wolf optimizer combined with back-propagation algorithm for neural network training," *Neural Computing and Applications*, vol. 30, pp. 1–14, 2018.

[76] D. Simon, "Biogeography-based optimization," *IEEE transactions on evolutionary computation*, vol. 12, pp. 702–713, 2008.

[77] R. Rarick, D. Simon, F. E. Villaseca, and B. Vyakaranam, "Biogeography-based optimization and the solution of the power flow problem," in *Proc. of International Conference on Man and Cybernetics*, 2009, pp. 1003–1008.

[78] V. Panchal, P. Singh, N. Kaur, and H. Kundra, "Biogeography based satellite image classification," *arXiv preprint arXiv:0912.1009*, 2009.

[79] S. H. A. Rahmati and M. Zandieh, "A new biogeography-based optimization (bbo) algorithm for the flexible job shop scheduling problem," *The International Journal of Advanced Manufacturing Technology*, vol. 58, pp. 1115–1129, 2012.

[80] R. Pal and M. Saraswat, "Data clustering using enhanced biogeography-based optimization," in *Proc. of International Conference on Tenth International Conference on*, 2017, pp. 1–6.

[81] W. Gong, Z. Cai, C. X. Ling, and H. Li, "A real-coded biogeography-based optimization with mutation," *Applied Mathematics and Computation*, vol. 216, pp. 2749–2758, 2010.

[82] H. Ma and D. Simon, "Blended biogeography-based optimization for constrained optimization," *Engineering Applications of Artificial Intelligence*, vol. 24, pp. 517–525, 2011.

[83] X. Li, J. Wang, J. Zhou, and M. Yin, "A perturb biogeography based optimization with mutation for global numerical optimization," *Applied Mathematics and Computation*, vol. 218, pp. 598–609, 2011.

[84] X. Li and M. Yin, "Multi-operator based biogeography based optimization with mutation for global numerical optimization," *Computers & Mathematics with Applications*, vol. 64, no. 9, pp. 2833–2844, 2012.

[85] M. Lohokare, S. S. Pattnaik, B. K. Panigrahi, and S. Das, "Accelerated biogeography-based optimization with neighborhood search for optimization," *Applied Soft Computing*, vol. 13, pp. 2318–2342, 2013.

[86] Q. Feng, S. Liu, G. Tang, L. Yong, and J. Zhang, "Biogeography-based optimization with orthogonal crossover," *Mathematical Problems in Engineering*, vol. 2013, 2013.

[87] G. Xiong, D. Shi, and X. Duan, "Enhancing the performance of biogeography-based optimization using polyphyletic migration operator and orthogonal learning," *Computers & Operations Research*, vol. 41, pp. 125–139, 2014.

[88] Q. Feng, S. Liu, Q. Wu, G. Tang, H. Zhang, and H. Chen, "Modified biogeography-based optimization with local search mechanism," *Journal of Applied Mathematics*, vol. 2013, pp. 100–105, 2013.

[89] W. Gong, Z. Cai, and C. X. Ling, "De/bbo: a hybrid differential evolution with biogeography-based optimization for global numerical optimization," *Soft Computing*, vol. 15, pp. 645–665, 2010.

[90] I. Boussaïd, A. Chatterjee, P. Siarry, and M. Ahmed-Nacer, "Two-stage update biogeography-based optimization using differential evolution algorithm (dbbo)," *Computers & Operations Research*, vol. 38, pp. 1188–1198, 2011.

[91] T. Niknam, S. Sharifinia, and R. Azizipanah-Abarghooee, "A new enhanced bat-inspired algorithm for finding linear supply function equilibrium of gencos in the competitive electricity market," *Energy Conversion and Management*, vol. 76, pp. 1015–1028, 2013.

[92] D. Sambariya and R. Prasad, "Robust tuning of power system stabilizer for small signal stability enhancement using metaheuristic bat algorithm," *International Journal of Electrical Power & Energy Systems*, vol. 61, pp. 229–238, 2014.

[93] J. W. Zhang and G. G. Wang, "Image matching using a bat algorithm with mutation," in *Proc. of International Conference on Applied Mechanics and Materials*, 2012, pp. 88–93.

[94] S. Mishra, K. Shaw, and D. Mishra, "A new meta-heuristic bat inspired classification approach for microarray data," *Procedia Technology*, vol. 4, pp. 802–806, 2012.

[95] P. Musikapun and P. Pongcharoen, "Solving multi-stage multi-machine multi-product scheduling problem using bat algorithm," in *Proc. of International Conference on management and artificial intelligence*, 2012, pp. 98–102.

[96] A. H. Gandomi and X.-S. Yang, "Chaotic bat algorithm," *Journal of Computational Science*, vol. 5, pp. 224–232, 2014.

[97] B. Bahmani-Firouzi and R. Azizipanah-Abarghooee, "Optimal sizing of battery energy storage for micro-grid operation management using a new improved bat algo-

rithm," *International Journal of Electrical Power & Energy Systems*, vol. 56, pp. 42–54, 2014.

[98] N. S. Jaddi, S. Abdullah, and A. R. Hamdan, "Multi-population cooperative bat algorithm-based optimization of artificial neural network model," *Information Sciences*, vol. 294, pp. 628–644, 2015.

[99] K. Khan and A. Sahai, "A comparison of ba, ga, pso, bp and lm for training feed forward neural networks in e-learning context," *International Journal of Intelligent Systems and Applications*, vol. 4, pp. 23–27, 2012.

[100] G. Wang and L. Guo, "A novel hybrid bat algorithm with harmony search for global numerical optimization," *Journal of Applied Mathematics*, vol. 2013, 2013.

[101] X.-s. He, W.-J. Ding, and X.-S. Yang, "Bat algorithm based on simulated annealing and gaussian perturbations," *Neural Computing and Applications*, vol. 25, pp. 459–468, 2014.

[102] J. Sadeghi, S. M. Mousavi, S. T. A. Niaki, and S. Sadeghi, "Optimizing a bi-objective inventory model of a three-echelon supply chain using a tuned hybrid bat algorithm," *Transportation Research Part E: Logistics and Transportation Review*, vol. 70, pp. 274–292, 2014.

[103] S. Yılmaz and E. U. Küçüksille, "A new modification approach on bat algorithm for solving optimization problems," *Applied Soft Computing*, vol. 28, pp. 259–275, 2015.

[104] G. Wang and L. Guo, "A novel hybrid bat algorithm with harmony search for global numerical optimization," *Journal of Applied Mathematics*, vol. 2013, 2013.

[105] U. M. Fayyad, A. Wierse, and G. G. Grinstein, *Information visualization in data mining and knowledge discovery*. Morgan Kaufmann, 2002.

[106] M. Friedman, M. Last, Y. Makover, and A. Kandel, "Anomaly detection in web documents using crisp and fuzzy-based cosine clustering methodology," *Information sciences*, vol. 177, pp. 467–475, 2007.

[107] L. Liao, T. Lin, and B. Li, "Mri brain image segmentation and bias field correction based on fast spatially constrained kernel clustering approach," *Pattern Recognition Letters*, vol. 29, pp. 1580–1588, 2008.

[108] H. Frigui and R. Krishnapuram, "A robust competitive clustering algorithm with applications in computer vision," *Ieee transactions on pattern analysis and machine intelligence*, vol. 21, pp. 450–465, 1999.

[109] Y. Leung, J.-S. Zhang, and Z.-B. Xu, "Clustering by scale-space filtering," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, pp. 1396–1410, 2000.

[110] S. C. Johnson, "Hierarchical clustering schemes," *Psychometrika*, vol. 32, pp. 241–254, 1967.

[111] F. Murtagh, "A survey of recent advances in hierarchical clustering algorithms," *The Computer Journal*, vol. 26, pp. 354–359, 1983.

[112] A. Baraldi and P. Blonda, "A survey of fuzzy clustering algorithms for pattern recognition. i," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 29, pp. 778–785, 1999.

[113] F. T. Evers, F. Höppner, F. Klawonn, R. Kruse, and T. Runkler, *Fuzzy cluster analysis: methods for classification, data analysis and image recognition*. John Wiley & Sons, 1999.

[114] F. Hoeppner, "Fuzzy shell clustering algorithms in image processing: fuzzy c-rectangular and 2-rectangular shells," *IEEE Transactions on Fuzzy Systems*, vol. 5, pp. 599–613, 1997.

[115] P.-E. Danielsson, "Euclidean distance mapping," *Computer Graphics and image processing*, vol. 14, pp. 227–248, 1980.

[116] R. Wilkinson and P. Hingston, "Using the cosine measure in a neural network for document retrieval," in *Proc. of International Conference on Research and development in information retrieval*, 1991, pp. 202–210.

[117] S. Das, A. Abraham, and A. Konar, "Automatic clustering using an improved differential evolution algorithm," *IEEE Transactions on systems, man, and cybernetics-Part A: Systems and Humans*, vol. 38, pp. 218–237, 2008.

[118] S. Nanda and G. Panda, "Automatic clustering using moclonal for classifying actions of 3d human models," in *Proc. of International Conference on Humanities, Science and Engineering Research*, 2012, pp. 945–950.

[119] M. Younis, P. Munshi, G. Gupta, and S. M. Elsharkawy, "On efficient clustering of wireless sensor networks," in *Proc. of International Conference on Dependability and Security in Sensor Networks and Systems*, 2006, pp. 10–14.

[120] S. K. Halgamuge and L. Wang, *Classification and clustering for knowledge discovery*. Springer Science & Business Media, 2005, vol. 4.

[121] R. MacGregor, *Small Business Clustering Technologies: Applications in Marketing, Management, IT and Economics: Applications in Marketing, Management, IT and Economics*. IGI Global, 2006.

[122] M. N. Murty and A. K. Jain, "Knowledge-based clustering scheme for collection management and retrieval of library books," *Pattern recognition*, vol. 28, pp. 949–963, 1995.

[123] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proc. of International Conference on mathematical statistics and probability*, 1967, pp. 281–297.

[124] M. Steinbach, G. Karypis, V. Kumar *et al.*, "A comparison of document clustering techniques," in *Proc. of International Conference on Text Mining*, 2000, pp. 525–526.

[125] S. J. Phillips, "Acceleration of k-means and related clustering algorithms," in *Proc. of International Conference on Algorithm Engineering and Experimentation*, 2002, pp. 166–177.

[126] B. Zhang, M. Hsu, and U. Dayal, "K-harmonic means-a spatial clustering algorithm with boosting," in *Proc. of Temporal, spatial, and spatio-temporal data mining*. Springer, 2001, pp. 31–45.

[127] A. Chaturvedi, P. E. Green, and J. D. Caroll, "K-modes clustering," *Journal of Classification*, vol. 18, pp. 35–55, 2001.

[128] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural computation*, vol. 10, pp. 1299–1319, 1998.

[129] S. Z. Selim and K. Alsultan, "A simulated annealing algorithm for the clustering problem," *Pattern recognition*, vol. 24, pp. 1003–1008, 1991.

[130] J. C. Bezdek, S. Boggavarapu, L. O. Hall, and A. Bensaid, "Genetic algorithm guided clustering," in *Proc. of International Conf. on Evolutionary Computational Intelligence.*, 1994, pp. 34–39.

[131] M. Sarkar, B. Yegnanarayana, and D. Khemani, "A clustering algorithm using an evolutionary programming-based approach," *Pattern Recognition Letters*, vol. 18, pp. 975–986, 1997.

[132] A. E. Langham and P. Grant, "Using competing ant colonies to solve k-way partitioning problems with foraging and raiding strategies," in *Proc. of International Conference on European Conference on Artificial Life*, 1999, pp. 621–625.

[133] K. Krishna and M. N. Murty, "Genetic k-means algorithm," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 29, pp. 433–439, 1999.

[134] Y. Lu, S. Lu, F. Fotouhi, Y. Deng, and S. J. Brown, "Incremental genetic k-means algorithm and its application in gene expression data analysis," *BMC bioinformatics*, vol. 5, p. 172, 2004.

[135] W. Sheng and X. Liu, "A hybrid algorithm for k-medoid clustering of large data sets," in *Proc. of International Conference on Evolutionary Computation*, 2004, pp. 77–82.

[136] R. Kuo and L. Lin, "Application of a hybrid of genetic algorithm and particle swarm optimization algorithm for order clustering," *Decision Support Systems*, vol. 49, pp. 451–462, 2010.

[137] F.-j. Sun and Y. Tian, "Transmission line image segmentation based ga and pso hybrid algorithm," in *Proc. of International Conference on Computational and Information Sciences*, 2010, pp. 677–680.

[138] Y. Hong and S. Kwong, "To combine steady-state genetic algorithm and ensemble learning for data clustering," *Pattern Recognition Letters*, vol. 29, pp. 1416–1423, 2008.

[139] A. A. Chaves and L. A. N. Lorena, "Hybrid evolutionary algorithm for the capacitated centered clustering problem," *Expert Systems with Applications*, vol. 38, pp. 5013–5018, 2011.

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

[141] E. Falkenauer, *Genetic algorithms and grouping problems*. Wiley New York, 1998.

[142] L. Agustı, S. Salcedo-Sanz, S. Jiménez-Fernández, L. Carro-Calvo, J. Del Ser, J. A. Portilla-Figueras *et al.*, "A new grouping genetic algorithm for clustering problems," *Expert Systems with Applications*, vol. 39, pp. 9695–9703, 2012.

[143] C. Tan, T. Chuah, S. Tan, and M. Sim, "Efficient clustering scheme for ofdma-based multicast wireless systems using grouping genetic algorithm," *Electronics letters*, vol. 48, pp. 184–186, 2012.

[144] G. P. Babu and M. N. Murty, "Clustering with evolution strategies," *Pattern recognition*, vol. 27, pp. 321–329, 1994.

[145] H.-G. Beyer and H.-P. Schwefel, "Evolution strategies–a comprehensive introduction," *Natural computing*, vol. 1, pp. 3–52, 2002.

[146] K. Lee, J. H. Kim, T. S. Chung, B.-S. Moon, H. Lee, and I. S. Kohane, "Evolution strategy applied to global optimization of clusters in gene expression data of dna microarrays," in *Proc. of International Conference on Evolutionary Computation*, 2001, pp. 845–850.

[147] D. Van der Merwe and A. P. Engelbrecht, "Data clustering using particle swarm optimization," in *Proc. of International Conference on Evolutionary Computation*, 2003, pp. 215–220.

[148] S. C. Cohen and L. N. de Castro, "Data clustering with particle swarms," in *Proc. of International Conference on Evolutionary Computation*. IEEE, 2006, pp. 1792–1798.

[149] B. Jarboui, M. Cheikh, P. Siarry, and A. Rebai, "Combinatorial particle swarm optimization (cpso) for partitional clustering problem," *Applied Mathematics and Computation*, vol. 192, pp. 337–345, 2007.

[150] L.-Y. Chuang, C.-J. Hsiao, and C.-H. Yang, "Chaotic particle swarm optimization for data clustering," *Expert systems with Applications*, vol. 38, pp. 14 555–14 563, 2011.

[151] C.-Y. Tsai and I.-W. Kao, "Particle swarm optimization with selective particle regeneration for data clustering," *Expert Systems with Applications*, vol. 38, pp. 6565–6576, 2011.

[152] J. Sun, W. Chen, W. Fang, X. Wun, and W. Xu, "Gene expression data analysis with the clustering method based on an improved quantum-behaved particle swarm

optimization," *Engineering Applications of Artificial Intelligence*, vol. 25, pp. 376–391, 2012.

[153] M. Eusuff, K. Lansey, and F. Pasha, "Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization," *Engineering optimization*, vol. 38, pp. 129–154, 2006.

[154] A. Bhaduri and A. Bhaduri, "Color image segmentation using clonal selection-based shuffled frog leaping algorithm," in *Proc. of International Conference on Advances in Recent Technologies in Communication and Computing*, 2009, pp. 517–520.

[155] 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, pp. 9847–9852, 2009.

[156] K. Y. Huang, "A hybrid particle swarm optimization approach for clustering and classification of datasets," *Knowledge-Based Systems*, vol. 24, pp. 420–426, 2011.

[157] Z. Du, Y. Wang, and Z. Ji, "Pk-means: A new algorithm for gene clustering," *Computational Biology and Chemistry*, vol. 32, pp. 243–247, 2008.

[158] Y. Zhang, D. Huang, M. Ji, and F. Xie, "Image segmentation using pso and pcm with mahalanobis distance," *Expert Systems with Applications*, vol. 38, pp. 9036–9040, 2011.

[159] R. Xu, J. Xu, and D. C. Wunsch, "Clustering with differential evolution particle swarm optimization," in *Proc. of International Conference on Evolutionary Computation*, 2010, pp. 1–8.

[160] R. Kuo, Y. Syu, Z.-Y. Chen, and F.-C. Tien, "Integration of particle swarm optimization and genetic algorithm for dynamic clustering," *Information Sciences*, vol. 195, pp. 124–140, 2012.

[161] K. Thangavel, J. Bagyamani, and R. Rathipriya, "Novel hybrid pso-sa model for biclustering of expression data," *Procedia Engineering*, vol. 30, pp. 1048–1055, 2012.

[162] M. F. Lima, L. D. Sampaio, B. B. Zarpelao, J. J. Rodrigues, T. Abrao, and M. L. Proença Jr, "Networking anomaly detection using dsns and particle swarm optimization with re-clustering," in *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*. IEEE, 2010, pp. 1–6.

[163] M. Omran, A. P. Engelbrecht, and A. Salman, "Particle swarm optimization method for image clustering," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 19, pp. 297–321, 2005.

[164] C.-Y. Lee, J.-J. Leou, and H.-H. Hsiao, "Saliency-directed color image segmentation using modified particle swarm optimization," *Signal Processing*, vol. 92, pp. 1–18, 2012.

[165] H. Yu and W. Xiaohui, "Pso-based energy-balanced double cluster-heads clustering routing for wireless sensor networks," *Procedia Engineering*, vol. 15, pp. 3073–3077, 2011.

[166] S. Nanda, B. Mahanty, and M. Tiwari, "Clustering indian stock market data for portfolio management," *Expert Systems with Applications*, vol. 37, pp. 8793–8798, 2010.

[167] D. Mishra, "Discovery of overlapping pattern biclusters from gene expression data using hash based pso," *Procedia Technology*, vol. 4, pp. 390–394, 2012.

[168] O. Durán, N. Rodriguez, and L. A. Consalter, "A pso-based clustering algorithm for manufacturing cell design," in *Proc. of International Conference on Forensic applications and techniques in telecommunications, information, and multimedia*, 2008, pp. 53–53.

[169] X. Cui, T. E. Potok, and P. Palathingal, "Document clustering using particle swarm optimization," in *Proc. of International Conference on Swarm Intelligence*, 2005, pp. 185–191.

[170] Y. Cheng, M. Jiang, and D. Yuan, "Novel clustering algorithms based on improved artificial fish swarm algorithm," in *Proc. of International Conference on Fuzzy Systems and Knowledge Discovery*, 2009, pp. 141–145.

[171] W. Zou, Y. Zhu, H. Chen, and X. Sui, "A clustering approach using cooperative artificial bee colony algorithm," *Discrete dynamics in nature and society*, vol. 2010, 2010.

[172] S.-C. Chu, J. F. Roddick, C.-J. Su, and J.-S. Pan, "Constrained ant colony optimization for data clustering," in *Proc. of International Conference on Artificial Intelligence*, 2004, pp. 534–543.

[173] X. Xu, L. Chen, and Y. Chen, "A/sup 4/c: an adaptive artificial ants clustering algorithm," in *Proc. of International Conference Computational Intelligence in Bioinformatics and Computational Biology*, 2004, pp. 268–275.

[174] D. A. Ingaramo, M. G. Leguizamón, and M. L. Errecalde, "Adaptive clustering with artificial ants," *Journal of Computer Science & Technology*, vol. 5, 2005.

[175] A. Ghosh, A. Halder, M. Kothari, and S. Ghosh, "Aggregation pheromone density based data clustering," *Information Sciences*, vol. 178, pp. 2816–2831, 2008.

[176] S. Ghosh, M. Kothari, A. Halder, and A. Ghosh, "Use of aggregation pheromone density for image segmentation," *Pattern Recognition Letters*, vol. 30, no. 10, pp. 939–949, 2009.

[177] Y. Yang, M. Kamel, and F. Jin, "Topic discovery from document using ant-based clustering combination," in *Proc. of International Conference on Asia-Pacific Web Conference*, 2005, pp. 100–108.

[178] Y. Yang and M. S. Kamel, "An aggregated clustering approach using multi-ant colonies algorithms," *Pattern Recognition*, vol. 39, pp. 1278–1289, 2006.

[179] J. Handl and B. Meyer, "Improved ant-based clustering and sorting in a document retrieval interface," in *Proc. of International Conference on Parallel Problem Solving from Nature*, 2002, pp. 913–923.

[180] J. Handl, J. Knowles, and M. Dorigo, "Ant-based clustering and topographic mapping," *Artificial life*, vol. 12, pp. 35–62, 2006.

[181] R. Kuo, H. Wang, T.-L. Hu, and S. Chou, "Application of ant k-means on clustering analysis," *Computers & Mathematics with Applications*, vol. 50, pp. 1709–1724, 2005.

[182] S.-C. Chi and C. C. Yang, "Integration of ant colony som and k-means for clustering analysis," in *Proc. of International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, 2006, pp. 1–8.

[183] 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, pp. 8679–8684, 2010.

[184] A. H. Channa, N. M. Rajpoot, and K. M. Rajpoot, "Texture segmentation using ant tree clustering," in *Proc. of International Conference on Engineering of Intelligent Systems*, 2006, pp. 1–6.

[185] N. Labroche, N. Monmarché, and G. Venturini, "Antclust: ant clustering and web usage mining," in *Proc. of International Conference on Genetic and evolutionary computation*, 2003, pp. 25–36.

[186] J. Chen, J. Sun, and Y. Chen, "A new ant-based clustering algorithm on high dimensional data space," in *Complex Systems Concurrent Engineering*. Springer, 2007, pp. 605–611.

[187] Y. He and S. C. Hui, "Exploring ant-based algorithms for gene expression data analysis," *Artificial Intelligence in Medicine*, vol. 47, pp. 105–119, 2009.

[188] M. Ebrahimi, E. ShafieiBavani, R. K. Wong, S. Fong, and J. Fiaidhi, "An adaptive meta-heuristic search for the internet of things," *Future Generation Computer Systems*, vol. 76, pp. 486–494, 2017.

[189] D. E. Brown and C. L. Huntley, "A practical application of simulated annealing to clustering," *Pattern recognition*, vol. 25, pp. 401–412, 1992.

[190] L.-X. Sun, F. Xu, Y.-Z. Liang, Y.-L. Xie, and R.-Q. Yu, "Cluster analysis by the k-means algorithm and simulated annealing," *Chemometrics and intelligent laboratory systems*, vol. 25, pp. 51–60, 1994.

[191] Z. Güngör and A. Ünler, "K-harmonic means data clustering with simulated annealing heuristic," *Applied mathematics and computation*, vol. 184, pp. 199–209, 2007.

[192] W. Jin, X. Li, and Z. Baoyu, "A genetic annealing hybrid algorithm based clustering strategy in mobile ad hoc network," in *Proc. of International Conference on Communications, Circuits and Systems*, 2005, pp. 100–104.

[193] Z. Lu, Y. Peng, and H. H. Ip, "Combining multiple clusterings using fast simulated annealing," *Pattern Recognition Letters*, vol. 32, pp. 1956–1961, 2011.

[194] P. Moscato *et al.*, "On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms," *Caltech concurrent computation program, C3P Report*, vol. 826, p. 1989, 1989.

[195] C. M. Fuller, D. P. Biros, and D. Delen, "An investigation of data and text mining methods for real world deception detection," *Expert Systems with Applications*, vol. 38, pp. 8392–8398, 2011.

[196] A.-A. Salehpour, A. Afzali-Kusha, and S. Mohammadi, "Efficient clustering of wireless sensor networks based on memetic algorithm," in *Proc. of International Conference on Innovations in Information Technology*, 2008, pp. 450–454.

[197] L. Jiao, M. Gong, S. Wang, B. Hou, Z. Zheng, and Q. Wu, "Natural and remote sensing image segmentation using memetic computing," *IEEE computational Intelligence magazine*, vol. 5, pp. 78–91, 2010.

[198] H. Mittal and M. Saraswat, "An optimum multi-level image thresholding segmentation using non-local means 2d histogram and exponential kbest gravitational search

algorithm," *Engineering Applications of Artificial Intelligence*, vol. 71, pp. 226–235, 2018.

[199] O. Nasraoui, F. Gonzalez, C. Cardona, C. Rojas, and D. Dasgupta, "A scalable artificial immune system model for dynamic unsupervised learning," in *Proc. of International Conference on Genetic and Evolutionary Computation*, 2003, pp. 219–230.

[200] T. Liu, Y. Zhou, Z. Hu, and Z. Wang, "A new clustering algorithm based on artificial immune system," in *Proc. of International Conference on Fuzzy Systems and Knowledge Discovery*, 2008, pp. 347–351.

[201] Z. Li and H.-Z. Tan, "Generic intelligent systems-machine learning and classical ai-a combinational clustering method based on artificial immune system and support vector machine," *Lecture Notes in Computer Science*, vol. 4251, pp. 153–162, 2006.

[202] S. J. Nanda, G. Panda, and B. Majhi, "Improved identification of hammerstein plants using new cpso and ipso algorithms," *Expert systems with applications*, vol. 37, pp. 6818–6831, 2010.

[203] M. Wan, L. Li, J. Xiao, C. Wang, and Y. Yang, "Data clustering using bacterial foraging optimization," *Journal of Intelligent Information Systems*, vol. 38, pp. 321–341, 2012.

[204] G. S. Gaba, K. Singh, and B. S. Dhaliwal, "Sensor node deployment using bacterial foraging optimization," in *Proc. of international conference on recent trends in information systems*, 2011, pp. 73–76.

[205] X.-S. Yang, "Firefly algorithm, stochastic test functions and design optimisation," *International Journal of Bio-Inspired Computation*, vol. 2, pp. 78–84, 2010.

[206] A. Chowdhury, S. Bose, and S. Das, "Automatic clustering based on invasive weed optimization algorithm," in *Proc. of International Conference on Swarm, Evolutionary, and Memetic Computing*, 2011, pp. 105–112.

[207] R. Liu, X. Wang, Y. Li, and X. Zhang, "Multi-objective invasive weed optimization algortihm for clustering," in *Proc. of International Conference on Evolutionary Computation*, 2012, pp. 1–8.

[208] S. del Río, V. López, J. M. Benítez, and F. Herrera, "On the use of mapreduce for imbalanced big data using random forest," *Information Sciences*, vol. 285, pp. 112–137, 2014.

[209] Z.-H. You, J.-Z. Yu, L. Zhu, S. Li, and Z.-K. Wen, "A mapreduce based parallel svm for large-scale predicting protein–protein interactions," *Neurocomputing*, vol. 145, pp. 37–43, 2014.

[210] J. Ma, W. Gao, P. Mitra, S. Kwon, B. J. Jansen, K.-F. Wong, and M. Cha, "Detecting rumors from microblogs with recurrent neural networks," *IJCAI*, pp. 3818–3824, 2016.

[211] W. Zhao, H. Ma, and Q. He, "Parallel k-means clustering based on mapreduce," in *Proc. of International Conference on Cloud Computing*, 2009, pp. 674–679.

[212] A. Verma, X. Llorà, D. E. Goldberg, and R. H. Campbell, "Scaling genetic algorithms using mapreduce," in *Proc. of International Conference on Intelligent Systems Design and Applications*, 2009, pp. 13–18.

[213] R. Bhavani, G. S. Sadasivam, and R. Kumaran, "A novel parallel hybrid k-means-de-aco clustering approach for genomic clustering using mapreduce," in *Proc. of*

International Conference on Information and Communication Technologies, 2011, pp. 132–137.

[214] I. Aljarah and S. A. Ludwig, "Towards a scalable intrusion detection system based on parallel pso clustering using mapreduce," in *Proc. of International Conference on companion of Genetic and evolutionary computation*, 2013, pp. 169–170.

[215] B. Wu, G. Wu, and M. Yang, "A mapreduce based ant colony optimization approach to combinatorial optimization problems," in *Proc. of International Conference on Natural Computation*, 2012, pp. 728–732.

[216] H. Mittal and M. Saraswat, "ckgsa based fuzzy clustering method for image segmentation of rgb-d images," in *Proc. of International Conference on Eleventh International Conference on Contemporary Computing*, 2018, pp. 1–6.

[217] S. Yang, R. Wu, M. Wang, and L. Jiao, "Evolutionary clustering based vector quantization and spiht coding for image compression," *Pattern Recognition Letters*, vol. 31, pp. 1773–1780, 2010.

[218] J. Kogan, M. Teboulle, and C. Nicholas, "Data driven similarity measures for k-means like clustering algorithms," *Information Retrieval*, vol. 8, pp. 331–349, 2005.

[219] A. Banharnsakun, "A mapreduce-based artificial bee colony for large-scale data clustering," *Pattern Recognition Letters*, vol. 93, pp. 78–84, 2017.

[220] J. Dean and S. Ghemawat, "Mapreduce: simplified data processing on large clusters," *Communications of the ACM*, vol. 51, pp. 107–113, 2008.

[221] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature inspired cooperative strategies for optimization (NICSO 2010)*. Springer, 2010, pp. 65–74.

[222] C. Blake and C. J. Merz, "Uci repository of machine learning databases," 1998.

[223] T. Ashish, S. Kapil, and B. Manju, "Parallel bat algorithm-based clustering using mapreduce," in *Proc. of International Conference on Networking Communication and Data Knowledge Engineering*. Springer, 2018, pp. 73–82.

[224] M. F. Shlesinger, G. M. Zaslavsky, and U. Frisch, "Lévy flights and related topics in physics," *Lecture notes in physics*, vol. 450, pp. 52–54, 1995.

[225] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," *Machine learning*, vol. 3, pp. 95–99, 1988.

[226] J. Kennedy, "Particle swarm optimization," in *Proc. of International Conference on Encyclopedia of machine learning*, 2011.

[227] R. A. Formato, "Central force optimization: a new metaheuristic with applications in applied electromagnetics," *Progress In Electromagnetics Research*, vol. 77, pp. 425–491, 2007.

[228] A. Kaveh, *Advances in metaheuristic algorithms for optimal design of structures*. Springer, 2014.

[229] A. Kaveh and M. Khayatazad, "Ray optimization for size and shape optimization of truss structures," *Computers & Structures*, vol. 117, pp. 82–94, 2013.

[230] R. Rajabioun, "Cuckoo optimization algorithm," *Applied soft computing*, vol. 11, pp. 5508–5518, 2011.

[231] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE transactions on evolutionary computation*, vol. 1, pp. 67–82, 1997.

[232] "Detection dog - wikipedia, the free encyclopedia," https://en.wikipedia.org/wiki/Detection_dog.

[233] "Understanding a dog's senses," http://www.dogbreedinfo.com/articles/dogsenses.htm.

[234] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowledge-Based Systems*, vol. 89, pp. 228–249, 2015.

[235] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm," *Journal of global optimization*, vol. 39, pp. 459–471, 2007.

[236] "Idc study: Digital universe in 2020," https://www.kdnuggets.com/2012/12/idc-digital-universe-2020.html, (Accessed on 05/11/2018).

[237] S. Asur and B. A. Huberman, "Predicting the future with social media," in *Proc. of International Conference on Web Intelligence and Intelligent Agent Technology*. IEEE Computer Society, 2010, pp. 492–499.

[238] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: A survey," *Ain Shams Engineering Journal*, vol. 5, pp. 1093–1113, 2014.

[239] A. Z. Khan, M. Atique, and V. Thakare, "Combining lexicon-based and learning-based methods for twitter sentiment analysis," *International Journal of Electronics, Communication and Soft Computing Science & Engineering (IJECSCSE)*, p. 89, 2015.

[240] S. Canuto, M. A. Gonçalves, and F. Benevenuto, "Exploiting new sentiment-based meta-level features for effective sentiment analysis," in *Proc. of the ninth ACM international conference on web search and data mining*, 2016, pp. 53–62.

[241] F. Bravo-Marquez, M. Mendoza, and B. Poblete, "Combining strengths, emotions and polarities for boosting twitter sentiment analysis," in *Proc. of International Conference on Issues of Sentiment Discovery and Opinion Mining*, 2013, pp. 2–5.

[242] S. M. Mohammad, X. Zhu, S. Kiritchenko, and J. Martin, "Sentiment, emotion, purpose, and style in electoral tweets," *Information Processing & Management*, vol. 51, pp. 480–499, 2015.

[243] E. Kontopoulos, C. Berberidis, T. Dergiades, and N. Bassiliades, "Ontology-based sentiment analysis of twitter posts," *Expert systems with applications*, vol. 40, pp. 4065–4074, 2013.

[244] B. Agarwal, N. Mittal, P. Bansal, and S. Garg, "Sentiment analysis using common-sense and context information," *Computational intelligence and neuroscience*, vol. 2015, p. 30, 2015.

[245] H. Saif, Y. He, M. Fernandez, and H. Alani, "Contextual semantics for sentiment analysis of twitter," *Information Processing & Management*, vol. 52, pp. 5–19, 2016.

[246] G. Qiu, B. Liu, J. Bu, and C. Chen, "Expanding domain sentiment lexicon through double propagation." in *Proc. of Joint International on Conference on Artificial Intelligence*, 2009, pp. 1199–1204.

[247] R. Pandarachalil, S. Sendhilkumar, and G. Mahalakshmi, "Twitter sentiment analysis for large-scale data: an unsupervised approach," *Cognitive computation*, vol. 7, pp. 254–262, 2015.

[248] M. Fernández-Gavilanes, T. Álvarez-López, J. Juncal-Martínez, E. Costa-Montenegro, and F. J. González-Castaño, "Unsupervised method for sentiment analysis in online texts," *Expert Systems with Applications*, vol. 58, pp. 57–75, 2016.

[249] M. Kanakaraj and R. M. R. Guddeti, "Nlp based sentiment analysis on twitter data using ensemble classifiers," in *Proc. of International Conference on Signal processing, communication and networking*, 2015, pp. 1–5.

[250] B. Altınel and M. C. Ganiz, "A new hybrid semi-supervised algorithm for text classification with class-based semantics," *Knowledge-Based Systems*, vol. 108, pp. 50–64, 2016.

[251] A. Muhammad, N. Wiratunga, and R. Lothian, "Contextual sentiment analysis for social media genres," *Knowledge-based systems*, vol. 108, pp. 92–101, 2016.

[252] O. Appel, F. Chiclana, J. Carter, and H. Fujita, "A hybrid approach to the sentiment analysis problem at the sentence level," *Knowledge-Based Systems*, vol. 108, pp. 110–124, 2016.

[253] T. Chen, R. Xu, Y. He, Y. Xia, and X. Wang, "Learning user and product distributed representations using a sequence model for sentiment analysis," *IEEE Computational Intelligence Magazine*, vol. 11, pp. 34–44, 2016.

[254] E. Sulis, D. I. H. Farías, P. Rosso, V. Patti, and G. Ruffo, "Figurative messages and affect in twitter: Differences between# irony,# sarcasm and# not," *Knowledge-Based Systems*, vol. 108, pp. 132–143, 2016.

[255] A. S. H. Basari, B. Hussin, I. G. P. Ananta, and J. Zeniarja, "Opinion mining of movie review using hybrid method of support vector machine and particle swarm optimization," *Procedia Engineering*, vol. 53, pp. 453–462, 2013.

[256] D. K. Gupta, K. S. Reddy, A. Ekbal *et al.*, "Pso-asent: Feature selection using particle swarm optimization for aspect based sentiment analysis," in *Proc. of applications of natural language to information systems*, 2015, pp. 220–233.

[257] J. Zhu, H. Wang, and J. Mao, "Sentiment classification using genetic algorithm and conditional random fields," in *Proc. of International Conference on Information Management and Engineering*, 2010, pp. 193–196.

[258] G. Cui, H.-K. Lui, and X. Guo, "The effect of online consumer reviews on new product sales," *International Journal of Electronic Commerce*, vol. 17, pp. 39–58, 2012.

[259] C. M. Fuller, D. P. Biros, J. Burgoon, and J. Nunamaker, "An examination and validation of linguistic constructs for studying high-stakes deception," *Group Decision and Negotiation*, vol. 22, pp. 117–134, 2013.

[260] K. M. Hunt, "Gaming the system: Fake online reviews v. consumer law," *Computer Law & Security Review*, vol. 31, pp. 3–25, 2015.

[261] M. Luca and G. Zervas, "Fake it till you make it: Reputation, competition, and yelp review fraud," *Management Science*, vol. 62, pp. 3412–3427, 2016.

[262] D. Zhang, L. Zhou, J. L. Kehoe, and I. Y. Kilic, "What online reviewer behaviors really matter? effects of verbal and nonverbal behaviors on detection of fake online reviews," *Journal of Management Information Systems*, vol. 33, pp. 456–481, 2016.

[263] M. Crawford, T. M. Khoshgoftaar, J. D. Prusa, A. N. Richter, and H. Al Najada, "Survey of review spam detection using machine learning techniques," *Journal of Big Data*, vol. 2, pp. 23–29, 2015.

[264] M. Ott, Y. Choi, C. Cardie, and J. T. Hancock, "Finding deceptive opinion spam by any stretch of the imagination," in *Proc. of Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011, pp. 309–319.

[265] M. Ott, C. Cardie, and J. T. Hancock, "Negative deceptive opinion spam," in *Proc. of International Conference on Computational Linguistics: human language technologies*, 2013, pp. 497–501.

[266] N. H. Long, P. H. T. Nghia, and N. M. Vuong, "Opinion spam recognition method for online reviews using ontological features," *Tap chi Khao hoc*, vol. 61, p. 44, 2014.

[267] A. Mukherjee, A. Kumar, B. Liu, J. Wang, M. Hsu, M. Castellanos, and R. Ghosh, "Spotting opinion spammers using behavioral footprints," in *Proc. of International Conference on Knowledge discovery and data mining*, 2013, pp. 632–640.

[268] J. K. Rout, S. Singh, S. K. Jena, and S. Bakshi, "Deceptive review detection using labeled and unlabeled data," *Multimedia Tools and Applications*, vol. 76, pp. 3187–3211, 2017.

[269] A. Kulhari, A. Pandey, R. Pal, and H. Mittal, “Unsupervised data classification using modified cuckoo search method,” in *Proc. of International Conference on Contemporary Computing*, 2016, pp. 1–5.