A NEW VARIANT OF BLACK HOLE ALGORITHM BASED ON MULTI POPULATION AND LEVY FLIGHT FOR CLUSTERING PROBLEM

HANEEN ABDUL WAHAB ABDUL RAHEEM

DOCTOR OF PHILOSOPHY

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name : HANEEN ABDUL WAHAB

Date of Birth : 24/2/1986

Title : A NEW VARIANT OF BLACK HOLE ALGORITHM BASED ON MULTI POPULATION AND LEVY FLIGHT FOR CLUSTERING PROBLEM

Academic Session : SEM 2 2019/2020

I declare that this thesis is classified as:

☐	CONFIDENTIAL	(Contains confidential information under the Official Secret Act 1997)*
☐	RESTRICTED	(Contains restricted information as specified by the organization where research was done)*
☑	OPEN ACCESS	I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Universiti Malaysia Pahang reserves the following rights:

1. The Thesis is the Property of Universiti Malaysia Pahang
2. The Library of Universiti Malaysia Pahang has the right to make copies of the thesis for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

DR. ABDULRAHMAN AHMED MOHAMMED AL-SEWARI
SENIOR LECTURER
FACULTY OF COMPUTER SYSTEMS & SOFTWARE ENGINEERING
UNIVERSITI MALAYSIA PAHANG
LEBUHRAYA TUN RAZAK, 26300 GAMBANG,
KUANTAN, PAHANG.
TEL : 09-549 2244 FAX: 09-549 2144

(Student's Signature) (Supervisor's Signature)

A10245978 Dr. AbdulRahman A. Alsewari

New IC/Passport Number Name of Supervisor
Date: 17JUNE 2020 Date: 17 JUNE 2020

NOTE : * If the thesis is CONFIDENTIAL or RESTRICTED, please attach a thesis declaration letter.

MAKLUMAT PANEL PEMERIKSA PEPERIKSAAN LISAN

Tesis ini telah diperiksa dan diakui oleh
This thesis has been checked and verified by

Nama dan Alamat Pemeriksa Dalam *Name and Address Internal Examiner*	:	Prof. Dr. Kamal Z Zamli Faculty of Computing, Universiti Malaysia Pahang
Nama dan Alamat Pemeriksa Luar *Name and Address External Examiner*	:	Prof. Dr. Azuraliza Abu Bakar University Kebangsaan Malaysia
Nama dan Alamat Pemeriksa Luar *Name and Address External Examiner*	:	Prof. Dr. Abu bakar Md Sultan Department of Computing, College of Computing and Informatics, UPM

Disahkan oleh Penolong Pendaftar di IPS
Verified by Assistant Registrar IPS

Tandatangan : Tarikh:
Signature *Date*

Nama :
Name

SUPERVISOR's DECLARATION

We hereby declare that we have checked this thesis and in our opinion, this thesis is adequate in terms of scope and quality for the award of the degree of Doctor of Philosophy.

(Supervisor's Signature)

Full Name : DR. ABDULRAHMAN A. ALSEWARI

Position : SENIOR LECTURER

Date : 17 JUNE 2020

(Co-supervisor's Signature)

Full Name : DR NORAZIAH AHMAD

Position : ASSOCIATE PROFESSOR

Date : 17 JUNE 2020

STUDENT'S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institutions.

_____

(Student's Signature)

Full Name : HANEEN ABDULWAHAB ABDULRAHEEM

ID Number : PCS15002

Date : 17 JUNE 2020

A NEW VARIANT OF BLACK HOLE ALGORITHM BASED ON MULTI POPULATION AND LEVY FLIGHT FOR CLUSTERING PROBLEM

HANEEN ABDUL WAHAB ABDUL RAHEEM

Thesis submitted in fulfillment of the requirements
for the award of the degree of
Doctor of Philosophy

Faculty of Computing
UNIVERSITI MALAYSIA PAHANG

JUNE 2020

DEDICATION

To my parents
Mr & Mrs
Abdul Wahab and Ikram
To my sister
Duaa
This humble work is a sign of my love to you!

ACKNOWLEDGEMENTS

Undertaking this PhD has been a truly life-changing experience for me and it would not have been possible to do without the support and guidance that I received from many people.

Foremost, I would like to express my sincere gratitude to my supervisor Associate Professor Ts. Dr. AbdulRahman A. Alsewari, and my beloved Co-supervisor Associate Professor Dr. Noraziah Ahmad for taking out time to ensure qualitative supervision of this research. Thanks a lot for your support and frequent feedback.

My parents, I am deeply grateful to my parents my father Abdul wahab and my mother Ikram who encouraged and helped me at every stage of my personal and academic life, and longed to see this achievement come true. Their love, sacrifice and blessings have always been a source of energy and courage to me. They are driving force in my life which keeps me continuing. I owe them a lot and wish I could show them just how much I love and appreciate them.

My sister, a very special word of thanks goes for Duaa. For her advice, her patience, and her faith, because she was always there for me.

Above all, I owe it all to Almighty God for granting me the wisdom, health and strength to undertake this research task and enabling me to its completion.

ABSTRAK

Penggugusan data adalah salah satu cabang yang paling popular dalam pembelajaran mesin dan analisis data. Algoritma penggugusan berasaskan pemetakan seperti pendekatan cara K terdedah kepada masalah penghasilan satu set gugusan yang jauh dari sempurna disebabkan sifat kebarangkalian. Proses penggugusan bermula dengan beberapa sekatan rawak yang mencuba untuk memperbaiki sekatan secara beransur-ansur. Sekatan awalan yang berbeza boleh menghasilkan gugusan akhiran yang berbeza. Mencuba semua calon gugusan untuk hasil yang sempurna terlalu memakan masa. Algoritma metaheuristik bertujuan mencari global optimum dalam masalah berdimensi tinggi. Algoritma metaheuristik berjaya dilaksanakan pada masalah penggugusan data yang mencari penyelesaian optimum yang terhampir dari segi kualiti gugusan yang dihasilkan. Baru-baru ini, algoritma yang diilhami semula jadi dicadangkan dan digunakan untuk menyelesaikan masalah pengoptimuman secara umum dan masalah penggugusan data khususnya. Algoritma pengoptimuman lohong hitam (BH) digariskan sebagai penyelesaian bagi masalah-masalah penggugusan data. BH adalah metaheuristik berasaskan populasi yang meniru fenomena BH di alam semesta. Dalam hal ini, setiap penyelesaian yang bergerak dalam ruang carian mewakili bintang individu. BH asli menunjukkan prestasi yang baik apabila diterapkan pada dataset tanda aras; walau bagaimanapun, ia tidak mempunyai keupayaan penerokaan. Selaras dengan batasan ini, kajian ini mencadangkan varian baru BH melalui dua modifikasi yang berbeza pada BH asli. Pengubahsuaian pertama ialah penyepaduan algoritma BH dan penerbangan Levy, yang menghasilkan kaedah penggugusan data, iaitu "lohong hitam penerbangan Levy (LBH)". Dalam LBH, pergerakan setiap bintang bergantung pada saiz langkah yang dihasilkan oleh pengagihan Levy. Oleh itu, bintang akan meneroka kawasan yang lebih jauh dari BH terkini apabila nilai saiz langkahnya besar, dan sebaliknya. Pengubahsuaian kedua adalah BH populasi berganda yang dicadangkan sebagai generalisasi kepada algoritma BH, di mana algoritmanya tidak bergantung kepada penyelesaian terbaik, tetapi pada satu set penyelesaian terbaik yang dihasilkan, yang dikenali sebagai "MBH". Hasilnya, varian baru BH untuk dataset dimensi tinggi yang dipanggil lohong hitam Levy populasi berganda (MLBH) dicadangkan untuk mengendalikan dataset dimensi biasa dan tinggi melalui penyepaduan LBH dan MBH. Hasil yang diperoleh dibandingkan dengan BH dan algoritma-algoritma pengoptimuman sebelumnya untuk kedua-dua fungsi ujian serta penggugusan data dari segi dataset dimensi biasa dan tinggi. Keseluruhan hasil eksperimen dan analisis hasil yang diperoleh menunjukkan bahawa algoritma yang dicadangkan memenuhi sebagian besar kriteria yang diperlukan. Tambahan pula, keputusan menunjukkan kadar penumpuan yang tinggi, di mana prestasi algoritma tertakluk kepada masalah penggugusan data dan disiasat menggunakan enam dataset sebenar. Data-data ini diambil dari makmal pembelajaran mesin UCI. Arah penyelidikan masa depan juga dibincangkan dalam kajian ini.

ABSTRACT

Data clustering is one of the most popular branches in machine learning and data analysis. Partitioning-based type of clustering algorithms, such as K-means, is prone to the problem of producing a set of clusters that is far from perfect due to its probabilistic nature. The clustering process starts with some random partitions at the beginning, and it tries to improve the partitions progressively. Different initial partitions can result in different final clusters. Trying through all the possible candidate clusters for the perfect result is too time consuming. Metaheuristic algorithm aims to search for global optimum in high dimensional problems. Meta-heuristic algorithm has been successfully implemented on data clustering problems seeking a near optimal solution in terms of quality of the resultant clusters. Recently, nature-inspired algorithms have been proposed and utilized for solving the optimization problems in general, and data clustering problem in particular. Black Hole (BH) optimization algorithm has been underlined as a solution for data clustering problems. The BH is a population-based metaheuristic that emulates the phenomenon of the BH in the universe. In this instance, every solution in motion within the search space represents an individual star. The original BH has shown a superior performance when applied on a benchmark dataset; however, it lacks exploration capabilities. In keeping with this limitation, this study proposes a new variant of BH through two different modifications on the original BH. The first modification is the integration of BH algorithm and levy flight, which result in data clustering method, namely "Levy Flight Black Hole (LBH)". In LBH, the movement of each star mainly depends on the step size generated by the Levy distribution. Therefore, the star explores a far area from the current BH when the value step size is big, and vice versa. The second modification is the multiple population BH that is proposed as a generalization to the BH algorithm, in which the algorithm was not reliant upon the best solution but rather on a set of best solutions generated, called "MBH". As a result, a new variant of BH for high dimensional datasets which is called multiple population levy black hole (MLBH) has been proposed for handling normal and high dimensional datasets through the integration of LBH and MBH. The obtained results were compared with the BH and previous optimization algorithms for both test functions as well as data clustering in terms of normal and high dimensional datasets. Overall, the experimental outcomes and analysis of the obtained results indicated that the proposed algorithms have satisfied most of the required criteria. Furthermore, the results revealed a high convergence rate, upon which the algorithm's performance was subjected to data clustering problems and investigated using six real datasets. The datasets were retrieved from the UCI machine-learning laboratory. The future research directions are also discussed in the study.

TABLE OF CONTENT

DECLARATION

TITLE PAGE

ACKNOWLEDGEMENTS **ii**

ABSTRAK **iii**

ABSTRACT **iv**

TABLE OF CONTENT **v**

LIST OF TABLES **ix**

LIST OF FIGURES **x**

LIST OF ABBREVIATIONS **xiii**

CHAPTER 1 INTRODUCTION **1**

1.1 Background 1

1.2 Problem Statement 3

1.3 Research Objectives 5

1.4 Research Scope and Limitations 5

1.5 Thesis Organization 6

CHAPTER 2 LITERATURE REVIEW **8**

2.1 Introduction 8

2.2 Clustering 9

2.2.1 Clustering Optimization Problem 13

2.2.2 Challenges of Clustering 14

2.2.3 Clustering Categories 15

2.2.4 Clustering High Dimensional Data 16

2.2.5 Evolutionary Algorithms in Data Clustering 17
2.2.6 State of the Art 18
2.3 Metaheuristic Algorithms 33
2.3.1 Exploration and Exploitation 34
2.3.2 Types of Metaheuristics 35
2.4 Metaheuristics based Levy Flight (LF) 42
2.5 Analysis of the Previous Work 49
2.6 Review on BH algorithm 52
2.7 Gap Analysis 56
2.8 Summary 56

CHAPTER 3 RESEARCH METHODOLOGY 58

3.1 Introduction 58
3.2 Research Methodology 58
3.2.1 Literature Review Phase 60
3.2.2 The Methodology 60
3.2.3 Result and Discussion 61
3.3 The Original Black Hole (BH) 62
3.3.1 Black Hole Phenomena 62
3.3.2 Black Hole Algorithm 63
3.4 Multiple Levy Flight Black Hole (MLBH) Algorithm 66
3.4.1 Levy Flight Black Hole (LBH) Algorithm 67
3.4.2 Multiple Black Hole (MBH) Algorithm 69
3.4.3 Multiple Levy Flight Black Hole 74
3.5 Clustering of High Dimensional Datasets 81
3.5.1 Datasets 82

3.5.2 Pre-processing 82
3.5.3 Mutual Information (MI) 82
3.5.4 Clustering Algorithm 83
3.5.5 Evaluation Step 84
3.6 Test Functions, Datasets, and Evaluation Metrics 84
3.6.1 Evaluation on Benchmark Test Functions 84
3.6.2 Clustering Datasets 85
3.6.3 Evaluation Measures for Normal Datasets 87
3.7 High Dimensional Datasets 88
3.7.1 Colon Tumour 88
3.7.2 Breast Cancer 88
3.7.3 CNS 89
3.8 Evaluation Measure for High Dimensional Clustering 89
3.8.1 Davies-Bouldin Index 89
3.8.2 Intra Cluster Distance 90
3.9 Summary 90

CHAPTER 4 RESULTS AND DISCUSSION 91

4.1 Introduction 91
4.2 Experimental Settings 91
4.3 Experimental Results 94
4.3.1 Test functions comparison 95
4.3.2 Statistical Analysis for the Experimental Results 98
4.3.3 Convergence Rate Analysis 101
4.4 Clustering Performance 115
4.4.1 Normal Datasets 115

4.4.2 High Dimensional Datasets 126

4.4.3 Validity Threats 133

4.5 Summary 134

CHAPTER 5 CONCLUSION AND FUTURE WORK 135

5.1 Introduction 135

5.2 Objectives Revisited 135

5.3 Contribution 137

5.4 Future Work 137

REFERENCES 139

APPENDIX A LIST OF PUBLICATIONS 156

LIST OF TABLES

Table 2.1	The analysis of existing clustering algorithms	28
Table 2.2	Summary of the Classification of Snapshot Metaheuristics Agorithms	50
Table 2.3	Summary of the Classification of Snapshot Metaheuristics Agorithms	51
Table 3.1	Benchmark Test Functions	85
Table 3.2	The main characteristics of the used datasets	87
Table 3.3	The main characteristics of high dimensional datasets	88
Table 4.1	Parameter settings	93
Table 4.2	Results of LBH, MBH and MLBH over benchmark test function from f1 to f9	96
Table 4.3	Results of LBH, MBH and MLBH over benchmark test function from $f1$ to $f9$	97
Table 4.4	Wilcoxon test	99
Table 4.5	The sum of intra-cluster distances and error rate obtained on Iris datasets.	116
Table 4.6	The sum of intra-cluster distances and error rate obtained on Wine datasets.	118
Table 4.7	The sum of intra-cluster distances and error rate obtained on CMC datasets.	120
Table 4.8	The sum of intra-cluster distances and error rate obtained on Cancer datasets.	122
Table 4.9	The sum of intra-cluster distances and error rate obtained on Glass datasets.	123
Table 4.10	The sum of intra-cluster distances and error rate obtained on Vowel datasets.	125
Table 4.11	Results of Friedman test based on the error rate	126
Table 4.12	Comparison result between BH, LBH, MBH and MLBH on Colon Tumor Datasets.	127
Table 4.13	Comparison result between BH, LBH, MBH and MLBH on CNS Datasets.	129
Table 4.14	Comparison result between BH, LBH, MBH and MLBH on Breast Cancer Datasets.	131
Table 4.15	The performance analysis of MLBH and other algorithms	133

LIST OF FIGURES

Figure 2.1	Main concepts covered in chapter two	8
Figure 2.2	Random point	12
Figure 2.3	Input data	13
Figure 2.4	The classification of clustering algorithms	15
Figure 3.1	The research process	59
Figure 3.2	Experiment evaluation process	62
Figure 3.3	The Black Hole Algorithm	64
Figure 3.4	Pseudocode of BH algorithm	66
Figure 3.5	Motion path in Levy flight and Brownian (random) walk (Haklı & Uğuz, 2014)	68
Figure 3.6	Pseudocode of LBH	68
Figure 3.7	Flowchart of LBH algorithm	69
Figure 3.8	Psuedocode of MBH	72
Figure 3.9	Flowchart of MBH algorithm.	73
Figure 3.10	Graphical illustration of a star movment in BH, LBH, MBH, and MLBH	75
Figure 3.11	The block diagram of MLBH	79
Figure 3.12	Flowchart of MLBH	80
Figure 3.13	The framework of MLBH for High dimensional datasets.	81
Figure 4.1	The results of all tests	98
Figure 4.2	The 3D plot of sumsqaure ($f1$)with the convergence analysis of LBH and BH algorithm.	101
Figure 4.3	The 3D plot of Rastrigin ($f2$)with the convergence analysis of LBH and BH algorithm.	102
Figure 4.4	The 3D plot of Quatric ($f3$)with the convergence analysis of LBH and BH	102
Figure 4.5	The 3D plot of Ackley ($f4$)with the convergence analysis of LBH and BH algorithm.	103
Figure 4.6	The 3D plot of Alpin N1 ($f5$)with the convergence analysis of LBH and BH algorithm.	103
Figure 4.7	The 3D plot of Griewauk ($f6$)with the convergence analysis of LBH and BH algorithm.	104
Figure 4.8	The 3D plot of sumsqaure ($f1$)with the convergence analysis of MBH and BH algorithms	104
Figure 4.9	The 3D plot of Rastrigin ($f2$)with the convergence analysis of MBH and BH algorithm.	105

Figure 4.10 The 3D plot of Quatric ($f3$)with the convergence analysis of MBH and BH algorithm. 105
Figure 4.11 The 3D plot of Ackley ($f4$)with the convergence analysis of MBH and BH algorithm. 106
Figure 4.12 The 3D plot of Alpin N1 ($f5$)with the convergence analysis of MBH and BH algorithm. 106
Figure 4.13 The 3D plot of Griewauk ($f6$)with the convergence analysis of MBH and BH algorithm. 107
Figure 4.14 The 3D plot of sumsqaure ($f1$)with the convergence analysis of MLBH and BH algorithm. 107
Figure 4.15 The 3D plot of Rastrigin ($f2$)with the convergence analysis of MLBH and BH algorithm. 108
Figure 4.16 The 3D plot of Quatric ($f3$)with the convergence analysis of MLBH and BH algorithm. 108
Figure 4.17 The 3D plot of Ackley ($f4$)with the convergence analysis of MLBH and BH algorithm. 109
Figure 4.18 The 3D plot of Alpin N1 ($f5$)with the convergence analysis of MLBH and BH algorithm. 109
Figure 4.19 The 3D plot of Griewauk ($f6$)with the convergence analysis of MLBH and BH algorithm. 110
Figure 4.20 Convergence analysis of LBH with other algorithms 111
Figure 4.21. Convergence analysis of MBH with other algorithms 112
Figure 4.22. Convergence analysis of MLBH with other algorithms 113
Figure 4.23 The convergence analysis of ($f7$) for MLBH, LBH, MBH and BH. 114
Figure 4.24 The convergence analysis of ($f8$) for MLBH, LBH, MBH and BH. 114
Figure 4.25 The convergence analysis of ($f9$) for MLBH, LBH, MBH and BH. 115
Figure 4.26 Best results of Iris datasets 117
Figure 4.27 Best results of wine datasets 119
Figure 4.28 Best results obtained of CMC datasets 121
Figure 4.29 Best results obtained of cancer datasets 122
Figure 4.30 Best results obtained of glass datasets 124
Figure 4.31 Best results obtained of Vowel datasets. 126

LIST OF SYMBOLS

f_i	Objective function
X_i	The variables of the problem
BH	Star
t	Iteration
f_{BH}	Fitness values of BH
S	A monumental process
τ	The standard gamma function
μ	Position or shift parameter
γ	Scale parameter
N	Number of stars
i^{th}	Initialized population
f_i	Objective function
X_i	The variables of the problem

LIST OF ABBREVIATIONS

SBA	Swam-Based Algorithms
PBA	Physics-Based Algorithms
EA	Evolutionary Algorithms
GA	Genetic Algorithms
BH	Black Hole
GP	Genetic Programming
ES	Evolutionary Strategies
PSO	Particle Swarm Optimization
ABC	Artificial Bees Colony
FFA	Firefly Algorithm
BA	Bat Algorithm
GWO	Grey Wolf Optimizer
BFO	Bacterial Foraging Optimization
CSA	Cuckoo Search Algorithm
FSO	Fish Swarm Optimization

CHAPTER 1

INTRODUCTION

1.1 Background

We are living in an information explosion era where data is produced every second in numerous formats including text, characteristic, number, voice, video, etc. Recently with the concept of big data and its "4V" features –volume, velocity, variety and value, data scientists are confronted with a new realm of computational challenges. Exploring this data and extracting meaningful patterns has become a hot topic in Knowledge Discovery in Database and Data Analysis that attracts unprecedented research attention from academia and industry. New models, methods and techniques are proposed to solve these problems, i.e. deep learning and distributed operating system.

Data clustering is one of the most significant branches of machine learning and data analysis. It has been widely applied in many research areas including pattern recognition, image segmentation. Data clustering aims to find the structure of a given dataset by grouping together data vectors into a number of clusters according to their similarity. Owing to most data clustering is used for unsupervised learning, it is more challengeable than supervised regression or classification (Sarstedt & Mooi, 2019).

Data clustering is widely used in many areas including data mining, statistical data analysis, machine learning, pattern recognition, image analysis, information retrieval, and more. This is due to clustering methods that can be categorized into various methods, such as partitional, hierarchical, density-based, grid-based, and model-based methods, accordingly (Arora & Chana, 2014). Per the above methods, partitional

clustering methods are the type that is commonly used, in which the K-means algorithm is an example of partitional and center-based clustering algorithms.

Since 1960s, most of the clustering algorithms were developed from the classical data clustering principles, including partition-based method i.e. K-means algorithm, hierarchical algorithm which is also known as single-link algorithm, density based method e.g. DBSCAN, and model-based algorithm e.g. Gaussian Mixture Model Expectation Maximization Algorithm (GMM-EM) algorithm. Some algorithms are developed from famous classification algorithms, e.g. support vector clustering (SVC). These methods differ in the choices of the definition of objective function, probabilistic generative models, and heuristics methods (Diaz et al., 2018).

Clustering could be seen as an optimization problem that tries to label all the data instances into a certain set of classes, where the distances between the data in different classes are maximized and the distances among the data within the same class are minimized. Classical algorithms have their own shortcomings that many parameters need to be manually predefined and the choices have great effect on clustering results. For instance, the pre-set parameter k in K-means algorithm refers to how many clusters the algorithm aims to find (Janardhanan et al., 2019). The radius and eps in DBSCAN, refer to the maximum distance of two data that can be clustered into one class and the minimum number of data a cluster should have, respectively. Although a number of clustering methods have been proposed, they are confronted with difficulties in meeting the requirements of automation, high quality, and high efficiency at the same time (Gupta & Jha, 2018). Classical clustering method, such as K-means algorithm, has the advantage of being simple and fast. However, its iterative and probabilistic nature may lead the clustering centers stuck into local optimum easily. That means better arrangements of clusters could be available ahead, after some clusters were formed prematurely (Aggarwal & Reddy, 2013).

Meta-heuristic algorithms are some branches of optimization algorithms which are designed to find global or near global optimal solutions at reasonable computational cost. These algorithms have unique designs usually equipped with powerful searching ability (Kumar et al., 2018). They have been successfully implemented in many application areas including pattern recognition, data mining, parameter tuning and so on. The well-known metaheuristic algorithms that have ever been applied to data

clustering successfully include but not limit to Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Cuckoo Search Algorithm (CS), Firefly Algorithm (FF), and Wolf Search Algorithm (WSA) (Yang, 2010c).

In case of its implementation to solve a data clustering issue, the black hole remains relevant despite performance evaluation showing that BH is superior compared to other similar processes. Similarly, further enhancement for the approach will allow the discovery of powerful phenomenon in the solution space, while also making space for effectual clustering processing. In this perspective, the original black hole algorithm suffers from weaknesses in exploration (Kumar et al., 2015; Piotrowski et al., 2014). Therefore, it requires too many reiterations to attain an optimum resolution. In addition, the formula for moving the stars to explore the solution space also causes them to over-scatter and leads to slow convergence.

1.2 Problem Statement

Data clustering is one of the common data mining techniques that is used for retrieving useful information from a particular dataset (Agarwal & Mehta, 2019). Data clustering involves selecting the k-cluster centers randomly and grouping that data around those centers. Clustering techniques have been used in many areas such as image processing, document clustering, geophysics, prediction, marketing and customers' analysis, agriculture, security and crime detection, medicine, anomaly detection and biology (Mahdavi & Abolhassani, 2009; Škrjanc et al., 2018).

With the advancement in complementary data and knowledge base, gene expression data analysis is gradually shifting from the application of pure data-oriented methods to those that aim to include additional knowledge in data analysis, otherwise known as intelligent data analysis (Bellazzi & Zupan, 2007). Clustering techniques help in the understanding of the functions of genes, gene regulation, cellular processes and cell subtypes; genes with similar expression patterns and cellular functions can be clustered together. This approach can also help in understanding the unexpressed functions of many genes (Eisen et al., 1998). Co-expressed genes in the same cluster may be involved in the same cellular processes; a strong correlation of the expression patterns between such genes can indicate co-regulation (Clough & Barrett, 2016; Fehrmann et al., 2015; McDowell et al., 2018).

The nature inspired clustering techniques have been introduced recently as soft computing techniques based on the natural behaviour of swarms. These clustering algorithms generally make predictions on gene expression datasets by exploiting the similarity of gene expression patterns to make good clusters. However, such clustering suffers from incorrect grouping of genes. Although the soft clustering approach performs better than the traditional clustering, it still lags in adapting intelligence to discover inherent structure of clusters (Banu & Andrews, 2015).

Clustering problems can also be considered as optimization problems which can be addressed using either single or multi-objective metaheuristics (Jaiprakash & Nanda, 2019; Kowalski et al., 2019). A meta-heuristic optimization approach called Black Hole (BH) was invented recently. The BH optimization was inspired by nature or physics of BH and its interaction with the surrounding stars. It mimics the behaviour of the black hole in pulling the surrounding stars to itself (Hatamlou, 2013). This algorithm has been used to solve data clustering problems and it showed a superior performance compared to meta-heuristics (Zuwairie et al., 2018). The BH algorithm consists of two main searching components- the global search ability (exploration) and local search ability (exploitation). In the BH, the stars should explore the search space while moving towards the best solution (i.e., Black Hole) in a uniform distribution (randomly) generated step size (Piotrowski et al., 2014). This leads to the generation of almost the same step sizes for every star in the population to ensure the stars do not explore far areas from the best solution. In other words, the BH algorithm does not perform exploration in most iterations (Kumar et al., 2015); thus, the exploitation ability will be much higher than the exploration. Consequently, the algorithm will be easily trap in local optima (Mirjalili et al., 2016; Wang et al., 2015).

The BH was hybridized with a HS algorithm to solve the problem of BH. In this framework, the BH is used to produce an initial clustering solution to a problem while the HS algorithm is applied to improve the solution's quality (Chandrasekar & Krishnamoorthi, 2014; Eskandarzadehalamdary et al., 2014).

To overcome the issue of the BH being easily trapped in local optima, this study proposes the combination of Levy Flight with the uniform distribution movement equation in order to generate long and small step sizes (Chawla & Duhan, 2018; Emary et al., 2019), which will enhance the exploration ability of BH and keep it from local

optima entrapment. The proposed algorithm is called Levy flight black hole (LBH) algorithm. Even after the use of the Levy Flight to enhance the exploration of the BH, the exploration and exploitation capabilities of the resulting LBH are still not balanced. This is because of the failure of the best solution to explore different areas of the solution space when all the stars move towards the best solution (Hussain et al., 2018; Niu et al., 2005). To address this imbalance, the Multiple Population Black Hole (MBH), a new variant of the BH, was proposed for the enhancement of the trade-off between the exploration and exploitation capabilities of original BH. The overall problem of the BH (easy entrapment in local optima and exploration-exploitation imbalance) was solved in this study by combining LBH and MBH to produce a new variant of the BH algorithm called "Multiple Levy Flight Black Hole (MLBH)" algorithm. This algorithm ensures the minimization of the intra-cluster distance in the original BH and a trade-off between the exploration and exploitation capabilities of the BH. The MLBH was proposed to handle these issues in both normal and high dimensional datasets.

1.3 Research Objectives

The main goal of this research is to develop a new clustering algorithm for normal and high dimensional data based on Black Hole algorithm. In order to achieve this goal, the following objectives are formulated:

i. To design a new variant of black hole (BH) algorithm with levy flight (called LBH)

ii. To improve the BH and LBH by introducing the multi-population support (called MBH) and its ensemble algorithm (called MLBH).

iii. To evaluate LBH, MBH and MLBH with existing meta-heuristic algorithms use standard functions and datasets.

1.4 Research Scope and Limitations

As stated previously, data clustering is a very important process that is used to enhance the performance of data mining tasks. MLBH is single objective algorithm consists of two modifications, first the Levy Flight Black Hole (LBH), while the second

is the Multiple Black Hole (MBH). The proposed algorithm with their modifications is tested and validated over nine continuous benchmark test functions (unimodal and multimodal). The results of MLBH, LBH, and MBH are compared with of Big Bang–Big Crunch (BB-BC), Artificial Bees Colony (ABC), Particle Swarm Optimization (PSO), and Levy Firefly Algorithm (LFFA), Grey Wolf Optimizer (GWO), Gravitational search algorithm (GSA), Bat algorithm (BA), cat swarm algorithm (CSA), and Black hole (BH) respectively. Additionally, the proposed algorithm with their modifications are tested and validated on data clustering problem, over 6 normal well-known University of California Irvine (UCI) data sets, which have been used by many researchers in the literature. These datasets are Iris, Wine, Glass, Cancer, Contraceptive Method Choice (CMC) and Vowel. Finally, MLBH, LBH and MLBH used to solve the clustering problem in the high dimensional datasets.

1.5 Thesis Organization

The remainder of this thesis is organized into five chapters. The current chapter gives an overview of data clustering the background followed by problem statement, then the research objectives and scope.

Chapter 2 is basically divided into three main sections. The first section is about data clustering. The second section covers extensive review on the metaheuristic algorithm in data clustering and also various approaches were used to solve data clustering problem. The third section is about Black hole algorithm, a meta-heuristic algorithm which is inspired by the behaviour of physics phenomena, followed by review of Black Hole algorithm in different fields and the pros and cons of the algorithm. The chapter is concluded with highlights of the research gap in applying Black Hole algorithm for data clustering.

Chapter 3 discusses and justifies the detailed of research methodology that applied to achieve the research objectives. It describes all the new modification applies for the new variant and evaluation measures. The components of the LBH, MBH and MLBH are explained in detail. Moreover, the chapter describes the normal datasets and the high dimensional datasets.

Chapter 4 presents the experimental evaluation including: experimental setting, searching performance, clustering performance as well as the statistical analysis of this

study. Moreover, the chapter describes the experimental evaluation results of the proposed MLBH, LBH and MBH on high dimensional datasets. In the last chapter is concluded with the analysis and findings.

Finally, the conclusion of this work is given in chapter 5, where the achievements and contributions are summarised.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In this chapter, first the thesis briefly described data science, then, the main steps in data clustering before providing an overview of clustering based on metaheuristic algorithms. Additionally, the thesis also described the concept of black hole. An illustration of the review process carried out in this chapter is depicted in Figure 2.1.

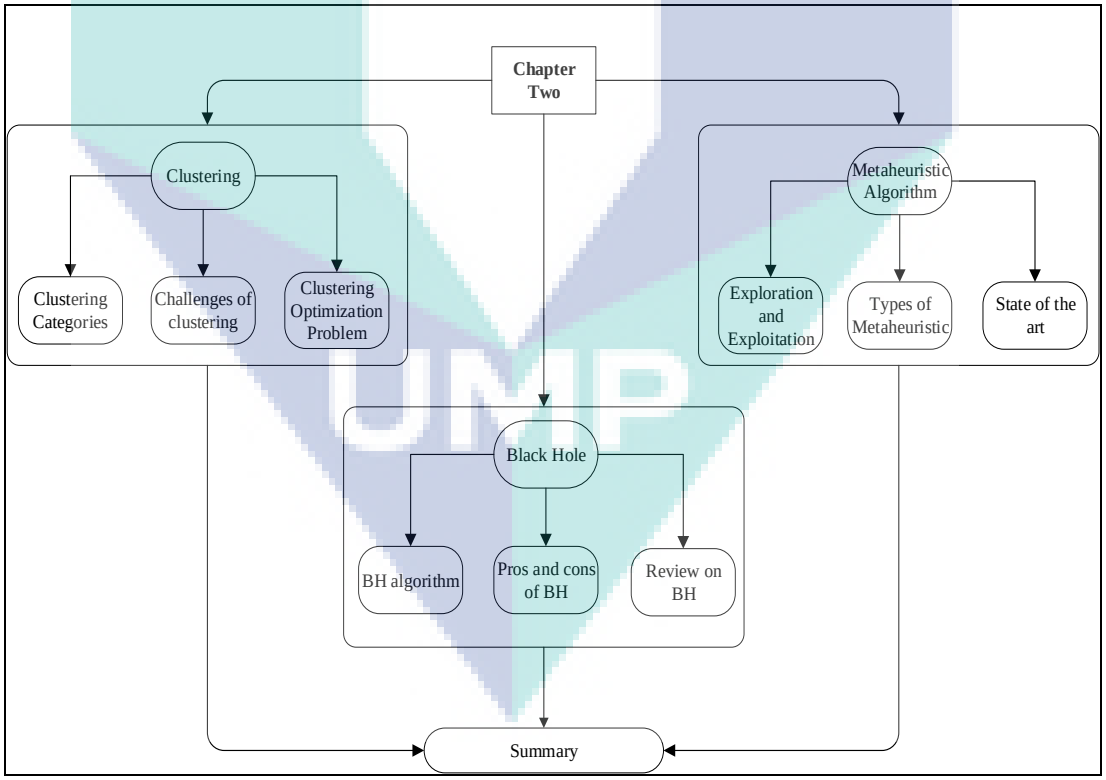


Figure 2.1 Main concepts covered in chapter two

2.2 Clustering

Clustering is a data mining technique (unsupervised) which can be applied effectively in such circumstance. In this technique, the data is partitioned into clusters and each cluster has elements of the same attributes but different from the elements of the other clusters. Clustering is applicable in data exploration to find the shape of the dataset; it can also be applied in the detection of anomalies. Data clustering before the analysis minimizes the computational cost significantly (Aggarwal & Reddy, 2013).

The term data clustering came into print for the first time in the year 1954 when an article dealing with anthropological data had this term in its title (Cohen et al., 1954). Cluster analysis has its origin in domains, such as machine learning, artificial intelligence, data mining, biology, statistics, and so on. Different fields use distinctive names for cluster analysis; some of them are as follows: Q Analysis, data visualization, typology, numerical taxonomy, clumping, and so on.

Clustering can be categorized into hard clustering or fuzzy clustering; hard clustering refers to the process of assigning each object to just one of the clusters with a certain level of membership (equal to 1) and well-defined boundaries with the other clusters. In Fuzzy clustering, each object can be assigned to more than one cluster with different degrees of membership (between 0 and 1), and fuzzy boundaries with the other clusters. The aim of hard clustering is to divide the dataset Z into c clusters (assuming that c is known based on the previous knowledge). A hard partition of Z can be defined using classical sets as a family of the subsets $\{A_i | 1 \leq i \leq c\} \subset P(Z)$.

- Union of all the $A_i s$ is equal to the data set Z itself.
- All these subsets are disjoint.
- There is no empty set, but no one contains all the data in Z.

Considering the membership functions (MF), a cluster can be conveniently represented by the partition matrix $U = [\mu_{ik}]_{c*N}$. In this matrix, the i^{th} contains values of the MF μ_i of the i^{th} subset $A_i \; of \; Z \; \mu_i$ and can take the value 0 and 1, and a classic one is described as follows:

(i) In the cluster, all the instances closely be alike while those in different clusters must significantly differ.

(ii) There must be a clear measure of similarity and differences, and they must have a practical meaning.

Similarity and dissimilarity measures are an important notion in clustering which deals with the similarities and differences between the objects to be clustered. There are several similarity measures, as discussed below:

- Euclidean Distance (ED): The ED between two points (p and q) is a measure of the length of the line that connects them. In Cartesian coordinates, if $p = p_1, p_2, p_3 \ldots . p_n)$, and $q = (q_1, q_2, \ldots \ldots, q_n)$, the ED from p to q or vice versa is given by:

$$\mathrm{D(p,q)} = \mathrm{d(q,p)} = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2} + \cdots (q_n - p_n)^2 = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2} \quad 2.1$$

- Manhattan Distance: Also known as rectilinear distance, $L1$ distance or 1norm, city block distance, Manhattan distance or Manhattan length, with corresponding variations in the name of the geometry.

$$\mathrm{d(p,q)} = \|p - q\| = \sqrt{\sum_{i=1}^{n}|q_i - p_i\|} \quad 2.2$$

- Mahalabonis Distance: This measure is based on the relationship between variables through which different patterns can be identified and analysed. This measure determines the similarity between a known and an unknown sample set. It is different from the ED by considering the relationship between a dataset and its scale-invariant. The Mahalanobis distance of a multivariate vector x from a group of values with mean and covariance matrix S is defined as:

$$x = (x_1, x_2, \ldots ., x_n)^T \; x = (x_1, x_2, \ldots ., x_n)^T \quad 2.3$$

$$x\mu = (\mu_1, \mu_2, \ldots ., \mu_n)^T \quad 2.4$$

$$D_M(x) = \sqrt{(x-\mu)^T S^{-1}}(x-\mu) \quad 2.5$$

If the covariance matrix is the identity matrix, the Mahalanobis distance reduces to the Euclidean distance (Nerurkar et al., 2019). If the covariance matrix is diagonal, then the resulting distance measure is called the normalized Euclidean distance.

- Hamming Distance: The Hamming distance between two strings of equal length is the number of positions at which the corresponding symbols are different. It measures the minimum number of substitutions required to change one string into the other, or the number of errors that transformed one string into the other. For binary strings a and b the Hamming distance is equal to the number of ones in $a\ XOR\ b$.
- Minkowski distance: The Minkowski distance is generalization of both the Euclidean distance and the Manhattan distance. The Minkowski distance of order p between two points $p = (x_1, x_2, x_3 \ldots\ldots x_n)$ and $Q = y_1, y_2, y_3 \ldots\ldots y_n) \in R^n$ is:

$$(\sum_{i=1}^{n} x_1\ |y_1)^{\frac{1}{p}} \quad 2.6$$

- Jaccard Distance: Jaccard index (Jaccard, 1901) is one of the external metrics that has been used in various studies as external index (Chaovalit, 2009; Papapetrou & Chen, 2011; Kremer et al., 2011). The Jaccard score is defined as:

$$\text{Jaccard}(\text{C}, \text{G}) = \sqrt{\frac{|TP|}{|\text{TP}| + |\text{FN}| + |\text{FP}|}} \quad 2.7$$

The standard clustering process can be classified into the following steps:

1- Feature extraction/selection: During feature selection, the specific features that portrays the differences between different patterns that belongs to different clusters (features such as immune to noise, easy to extract and interpret) are selected. However, some clustering frameworks perform poorly when applied to large highly dimensional datasets, especially the model-based algorithms which have been shown to be good on small-sized datasets but poor when applied to large-scale datasets.

2- Design/selection of a clustering algorithm: Usually, this step is combined with the corresponding proximity measure selection and the criterion function construction. Virtually, all clustering frameworks are explicitly or implicitly associated with some proximity measures definition. Upon the selection of a proximity measure, the clustering criterion function construction makes the clusters partitioning an optimization problem which is mathematically well defined but with several solutions in the literature. Several clustering algorithms have been developed to address different problems in several fields; however, no universal algorithm which can solve all problems currently exists.

3- Cluster validation: To provide users with a certain level of confidence for clustering results derived from certain algorithms, there is a need to have effective evaluation criteria and standards.

4- Results interpretation**:** Clustering processes mainly aims at the provision of meaningful insights from the original data which can be used to effectively solve the problems at hand. However, there may be a ned for further analyses to ensure the reliability of the extracted information from the dataset.

Clustering is often applied in image processing, medical imaging analysis, data statistical analysis, and other scientific/engineering fields. Additionally, it is a common statistical data analysis technique used in different fields such as machine learning, image analysis, pattern recognition, information retrieval, and bioinformatics. The differences in clusters based on their shape, size, and density. Figure 2.2 and Figure 2.3 demonstrates that clusters may differ in terms of their shape, size, and Density.

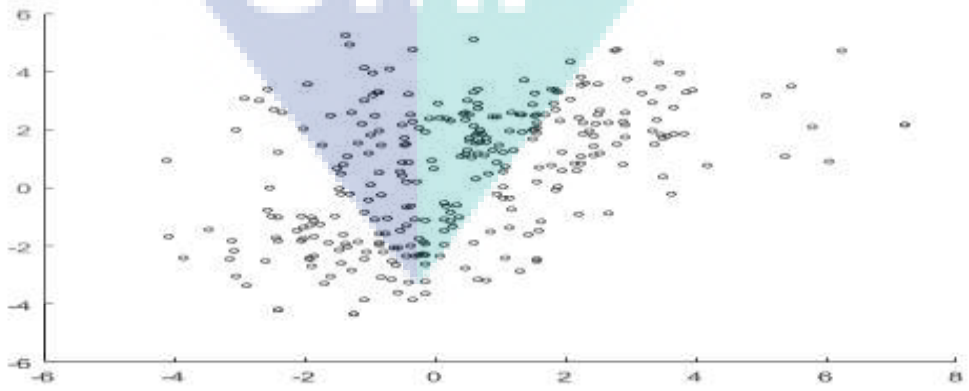


Figure 2.2 Random point

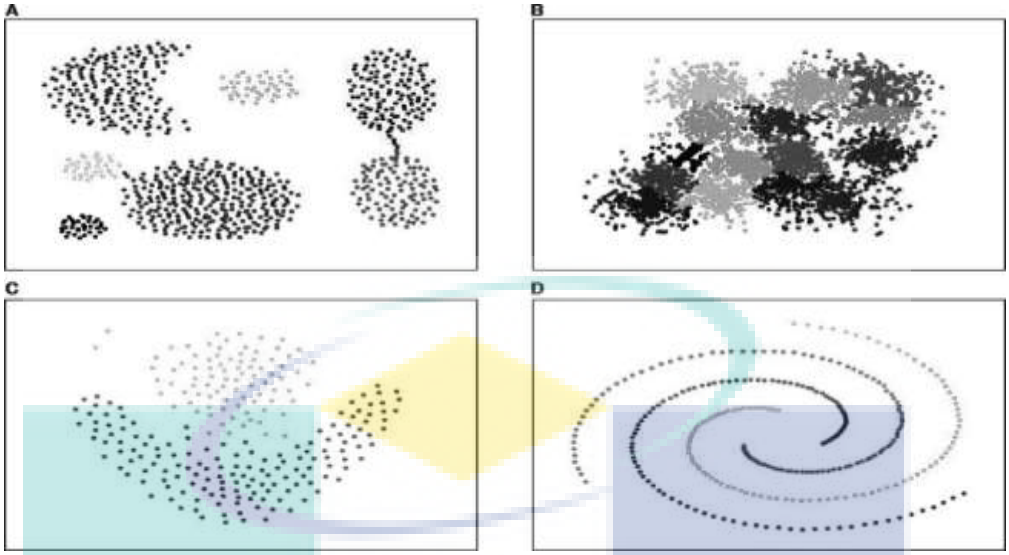


Figure 2.3 Input data

2.2.1 Clustering Optimization Problem

Clustering problems can be considered as optimization problems that could be addressed using either single or multi-objective metaheuristics (Nayak et al., 2019; Ramadas & Abraham, 2019). $Let\ \Omega = \{C^1, C^2, \ldots . C^{B(n)}\}$ represent a set of all the feasible clustering whose elements are the clustering solutions of a given dataset X, and let f represent a single criterion function. Then, the major aim of a single-objective clustering problem (Ω, f) is the determination of the clustering C^* for which $f(C^*) = min\{f(C)|C \in \Omega\}$; note the minimization of f $(\cdot)$ without a loss of generality. Contrarily, a multi-objective clustering problem $(\Omega, f_1, f_2, \ldots, f_m)$ aims at the determination of the clustering C^* for which $f(C^*) = min\{f(C)|C \in \Omega\}, t = 1,2, \ldots m$, where $f\_t, t = 1,2, \ldots m$ represents a set of m criterion functions. Normally, there are multiple optimal solution to multi-objective problems the pareto dominance principle (Nayak et al., 2019) which are often identified using. Considering two clustering solutions $C\_1, C\_2 \in \Omega, C\_1$ is said to have dominated C_2 (denoted as $C_1 \prec C_2$) if the following two criteria are met equation 2.8, 2.9 and 2.10 (Prakash & Singh, 2019):

$$f(C_1) \leq f_t(C_2) \forall t \in 1,2, \ldots . m \quad 2.8$$

$$f(C_1) < f_t(C_2) \exists t \in 1,2, \ldots . m \quad 2.9$$

$$F(O.Z) = \sum_{i=1}^{N} \sum_{j=1}^{K} \left\| O_i - Z_j \right\|^2 \quad 2.10$$

It is worth to mention that the proposed algorithms in this thesis solves the clustering problem as a single objective problem, where the main goal is to minimize the distance between the intra-clusters.

2.2.2 Challenges of Clustering

Clustering is an exploratory analysis technique that performs unsupervised learning. It has received several attentions because in practice, labelled data is usually available in a small proportion along with unlabelled data (Sivaraman et al., 2019). As no information is available regarding the number of clusters or with regard to specific assignments of objects, the clustering problem leaves space to a wide choice of objective functions and similarity functions, depending strongly on the domain under investigation. The choice is not straight forward. Thus, several challenges can be identified in the clustering analysis.

- An objective function must be formulated to quantify the degree of interest or naturalness in groupings.
- Although in clustering the data items are grouped based on similarity, the notion of similarity is seldom given in the problem statement. The metric employed has a great impact on the result of the clustering algorithm since under different metrics, the similarity space changes. If extra-information is available in the form of pair-wise constraints of data items that must reside in the same cluster, then, an optimal distance metric can be learned.
- The definition formulates clustering as an optimization problem. It is a hard optimization problem due to the huge search space. Even if the number of clusters is fixed, the number of possible partitions increases exponentially with the number of objects; the size of the search space in this case is given by the Stirling number of the second kind. When the number of clusters is not known, the number of ways to partition a set of n objects into non-empty subsets is given by the nth Bell number.

- A noisy data makes clusters detection more difficult; an ideal cluster is considered as a set of compact and isolated points. Although humans can seek clusters excellently in two or three dimensions, there is a need for automatic algorithms for highly dimensional datasets. In this challenge, the increased number of unknown number of clusters for a given data has resulted in the development of several clustering algorithms.

2.2.3 Clustering Categories

There is not direct or canonical way of classifying clustering algorithms; in fact, there is an overlap between different classes of clustering algorithms. The conventional clustering techniques are mainly classified into hierarchical, partitioning, grid-based, density-based, and model-based frameworks. Hierarchical clustering is further sub classified into divisive and agglomerative. Density is also subdivided into micro and grid based. The classification of clustering algorithms is depicted Figure 2.4.

Partitioning methods are among the most popular approaches to clustering due to the ease of its implementation and a favourable runtime behaviour. K-means is the best-known algorithm in this class. The criterion function used is that of minimum variance, i.e., the sum of squares of the differences between data items and their assigned cluster centers are minimized. Hierarchical algorithms build clusters gradually.

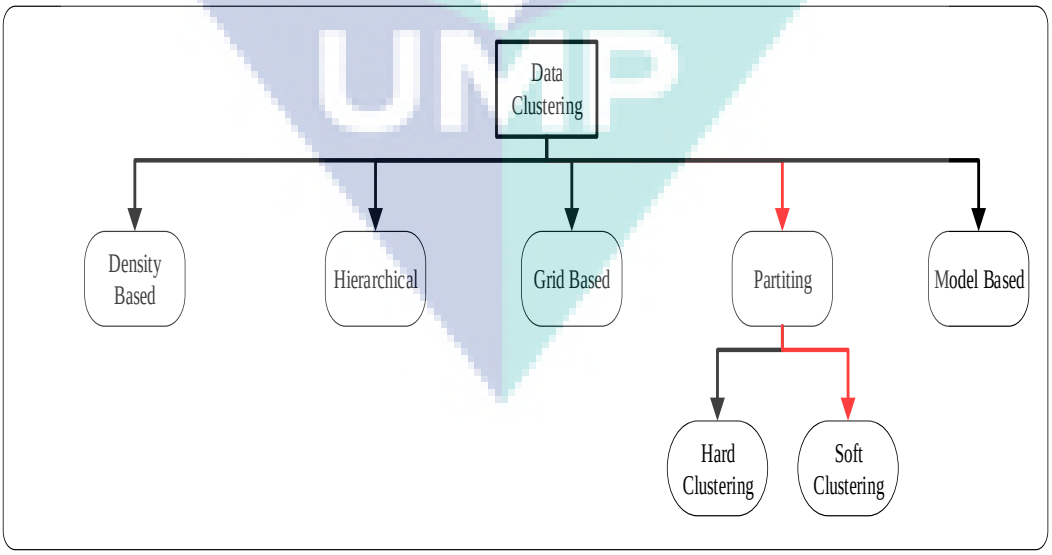


Figure 2.4 The classification of clustering algorithms

Furthermore, hierarchical clustering is subdivided into agglomerative and divisive approaches; an agglomerative approach is initiated with each pattern in a specific cluster before merging the successive clusters until a termination criterion is met. For the divisive method, it initialized with all the patterns in one single cluster, followed by their splitting until a termination criterion is met. Most of the hierarchical clustering algorithms are variants of the single and complete link algorithms.

In the density-based methods, the density, connectivity, and boundary concepts are applied towards the identification of clusters in the input data. The sensitivity of these algorithms is lower, and they can identify clusters of irregular shapes. They are usually suitable for low-dimensional data of numerical attributes, otherwise known as spatial data. Some of the algorithm that uses density-based connectivity are DBSCAN, OPTICS, and DBCLASD.

The grid-based methods perform space segmentation and then aggregate the appropriate segments. A partitioning of the input space in hyper rectangles is advantageous for application to large datasets. The algorithm, STING (Statistical information grid-based method), works with numerical attributes (spatial data) and is designed to facilitate "region-oriented" queries.

2.2.4 Clustering High Dimensional Data

The expression levels of thousands of genes can be simultaneously monitored today due to the emergence of microarray technology. However, the major problem is how to effectively manage this large volume of data. Microarray data are usually managed using either classification and clustering technique, but the clustering of microarray data analysis is the most significant aspect (Eisen et al., 1998). Microarray gene expression data clustering can provide information on the level of cellular processes, gene functions, and gene regulation (Jiang et al., 2004). Genes can be clustered based on tissues to detect a group of genes that undergo changes in their expression level or those that follow the same pattern. Genes that exhibit a similar pattern of expression under different conditions can participate in the same signal pathway; they can also be co-regulated. Clustering is proven to be an effective way of relating gene expression patterns with the ligand specificity and functional class of neurotransmitter receptors. Clustering has been used in cancer studies to identify both gene expression, cell type's signatures, and signatures for biological processes

(Alizadeh et al., 2000; Alon et al., 1999; Eisen et al., 1998; Golub et al., 1999; Spellman et al., 1998; Wen et al., 1998) used gene clustering techniques to analyze temporal gene expression data during the development of rat central nervous system.

2.2.5 Evolutionary Algorithms in Data Clustering

Darwin's theory of natural selection formed the basis for the evolutionary algorithms (EA) as they are based on the survival of fittest individual in any given environment. The EAs are initiated with a population that strives to survive in an environment. The offspring of any generation inherits the adaptability of their parents to a given via several evolutionary mechanisms such as crossover and mutation. This process is repeated over several generations until the most suitable solutions for the environment are found (Aljarah et al., 2019; Nayak et al., 2019).

2.2.5.1 Bio-inspired Data Clustering Algorithms

The biologically inspired (bio-inspired) frameworks comprise of the natural metaheuristics which are inspired by the living patterns and behaviours of biological organisms. These bio-inspired frameworks are distributed, self-organizing, decentralized, and adaptive in their nature. The major bio-inspired frameworks include Artificial Immune Systems (AIS), Dendritic Cell Algorithm, Bacterial Foraging Optimization (BFO), and Krill Herd algorithm. These algorithms are efficiently used in solving data clustering problems (Esmin et al., 2015; García et al., 2019; Lakshmi et al., 2018; Sarstedt & Mooi, 2019).

2.2.5.2 Physical Data Clustering Algorithms

These are algorithms that were developed based on inspirations from physical processes. Such algorithms include Simulated Annealing developed by (Kirkpatrick et al., 1983) based on the cooling and heating of materials; Discrete Cultural Information which was considered as in-between genetic and culture evolution (Moscato, 1989), Harmony Search by (Geem et al., 2001) based on the harmony of music played by musicians, and the Gravitational Search algorithm (GSA) based on the Big Bang-Big Crunch concept. These algorithms have successfully been applied to data clustering problems (Joshua et al., 2019).

For the last two decades, most of the nature-inspired metaheuristics have been applied for solving the problem of data clustering (Dey et al., 2019). Most of these metaheuristics were designed for global optimization problems in both type continuous and discrete types while some of them were mainly designed for the data clustering problem, such as Black Hole (BH) algorithm.

2.2.6 State of the Art

Researchers have always drawn inspiration from natural occurrences. Several algorithms have been proposed based on inspiration from natural process of evolution, laws, and social behaviour of species (Senthilnath et al., 2019). Nature-inspired algorithms are the most recent algorithms and they are efficient in handling optimization problems and other problems except the classical methods due to their inflexibility. Several researchers have shown that nature-inspired algorithms are efficient in handling complex computational problems. Numerous studies have been conducted on the use of metaheuristics for solving clustering problems. Thus, this is devoted to a brief literature overview of metaheuristic-based clustering algorithms with more focus on the most related techniques to the proposed algorithm in this study.

The Genetic algorithms (GAs) have been initially investigated for the improvement of the performance of classic clustering frameworks. For instance, a GA-based clustering technique known as called GA-clustering was proposed by Maulik and Bandyopadhyay (2000) and evaluated for superiority over the K-means algorithm. The investigation proved the superiority of the method over the k-means method via several dataset experiments. They performance of the algorithm was tested on both synthetic and real-life datasets.

A novel clustering algorithm for unsupervised learning which was inspired by the self-organizing behaviour observed in ants has been proposed by Xiao et al. (2003). The defined artificial ants similarly build a tree and each ant represents a data. The similarity between the data determines the way the ants move and build this tree. The obtained results from the proposed algorithm were compared to those from k-means algorithm and AntClass on numerical databases (either real, artificial). The result showed a significant improvement of the clustering process using the AntTree technique.

Van der Merwe and Engelbrecht (2003b) also proposed a standard PSO and a hybrid approach for clustering problem in the same year. In the proposed methods, the members of each swarm are seeded based on the result of the k-means algorithm. The performance of the two approaches was compared to that of K-means clustering and confirmed to be superior.

A year later, the Ant Colony optimization algorithm was proposed for solving clustering problems (Shelokar et al., 2004). The ACO software uses pheromone matrix (a kind of adaptive memory) to guide the other ants to the optimal clustering solution. The value of the objective function and the rate of pheromone evaporation determines its rate of deposition at location (i,j) (i.e. allocation of sample i to cluster j in a constructed solution. The rate of pheromone evaporation is a kind of a forgetting factor that helps to monitor the other clustering locations of object i. Thus, the optimal cluster representation for a clustering problem must be provided as iterations progress. The ACO algorithm can only be applied for data clustering when the number of clusters is previously known and are crispy in nature. The performance of the ACO algorithm was evaluated by comparison with other stochastic frameworks such as GA, SA, and Tabu search. The framework was implemented and evaluated on numerous real and simulated datasets and the preliminary computational experience was promising with respect to the quality of established solutions, the required processing time, and the average number of evaluation functions.

Handl et al. (2006) proposed an adaptive time-dependent transporter ant for clustering (ATTA-C) by suggesting for some modifications to the traditional ACO Ant-based clustering framework to penalize high dissimilarities, accelerate the clustering process, and improve the spatial separation between clusters. A neighbourhood function (NF) was used to calculate the fitness value of each clustering solution.

Chandramouli and Izquierdo (2006) proposed a PSO-based cluster analysis for image clustering. (Chu et al., 2004) suggested a constrained ACO (C-ACO) for the handling of arbitrarily shaped data clusters and outliers. Later, several researchers proposed the adaptive ACO for the improvement of the convergence rate and the determination of the optimal number of clusters (Dorigo et al., 2008).

A Tabu search-based clustering technique known as TS-Clustering was proposed by Liu et al. (2008) to handle the minimum sum of square clustering problem.

The author suggested three neighbourhood modes and five improvement operations in the algorithm. The suggested improvement operations were for the enhancement of the obtained clustering solution during the iteration process, while the neighbourhood mode is for the creation of the Tabu search neighbourhood. They used the generalized string property for the grouping of similar objects and setting up the initial solution, while the releasing procedure is for the separation of the packed elements from each other to enhance the effectiveness of the search process.

Two multiple pheromone concepts (ant-based clustering with ant nest algorithm and ant memory algorithm) were proposed by Ngenkaew et al. (2008). The ants were directed by the artificial trailing pheromone and foraging pheromone regarding the direction to follow or where to pick up or drop food items.

Chu et al. (2006) proposed a Cats Swarm Optimization algorithm by monitoring cats' natural hunting skills. Later, (Santosa & Ningrum, 2009) deployed the CSO-based clustering method to classify UCI benchmark datasets. The determination of the optimal solution by the algorithm is based on two operation modes of cats which are the seeking mode (representing the global search process which mimics the cats' resting position with slow movement) and the tracing mode (a local search technique that reflects the rapid chase of the target by the cat).

Cheng et al. (2009) artificial fish swarm algorithm, The AFSA several good application properties such as good optimization precision, strong robustness and flexibility in practice, rapidness to search the global optimum, searching adaptability, and tolerance of parameter setting. To reduce the complexity of the algorithm, a new fish behaviour called swallowing behaviour was proposed in the AFSA. The experimental results demonstrated a lower complexity of IAFSA compared to that of AFSA but with an almost the same performance with AFSA.

(Bhaduri & Bhaduri, 2009), the Shuffled Frog-Leaping Algorithm (SFLA) was proposed to solve clustering problems. The SFLA was developed as a metaheuristic to carry out an informed heuristic search based on a heuristic function to establish the solution to a combinatorial optimization problem. The SFLA was inspired by the evolution of memes carried by the interactive individuals and a global exchange of information among themselves. The algorithm is regarded as a typical swarm-based optimization approach. The formulation of the SFLA is based on two other search

techniques which are the local search of PSO and the competitiveness mixing of the Shuffled Complex Evolution technique. The application of the SFLA for data clustering can be done when the number of clusters is previously known, and the data are crisp in nature. The performance of the SFLA was evaluated by its comparison with other stochastic algorithms such as Ant colony, GA, SA, and Tabu search after its implementation on several real and simulations datasets.

An Artificial Bee Colony(ABC) was presented by Zhang et al. (2010) as a state-of-the-art clustering approach. To solve infeasible solutions, they authors adopted Deb's constrained handling method (Goldberg & Deb, 1991) usually used in the ABC algorithm instead of the greedy selection process. Upon testing the algorithm, the results were promising in terms of efficiency and effectiveness.

PSO-based clustering frameworks have been used effectively in several real-life applications such as node clustering (in wireless sensor network (WSN) for the enhancement of the sensors' lifetime and coverage area), energy-balanced cluster routing in WSN, cluster analysis of stock market data for portfolio management, clustering in ad hoc mobile networks for the determination of the cluster heads that will be responsible for topology information aggregation, grouping for security checks in power systems, colour image segmentation, gene expression data analysis, image clustering, clustering for manufacturing cell design, document clustering, network anomaly detection, and cluster analysis of web data usage (Rana et al., 2011).

A new nature-inspired algorithm called FA has been proposed for clustering and evaluation performance by Senthilnath et al. (2011) The performance of the proposed FA was compared with those of ABC, PSO, and other population-based nature-inspired optimization techniques. The performance of the technique was demonstrated using thirteen typical benchmark datasets sourced from the UCI machine learning repository and the results showed the FA to perform better than the benchmarking algorithms for clustering. The proposed ABC-based clustering algorithm was applied for solving sensor deployment and network routings problems in WSNs.

A new hybrid GSA-HS algorithm was proposed by Hatamlou et al. (2011c) based on GSA and heuristic search techniques. The initial population in the proposed algorithm was generated by the GSA after the application of the heuristic search technique for the exploration of the population. The evaluation of the GSA-HS

performance was based on two parameters (i.e. sum of intra-cluster distance and canter of the corresponding cluster, including four benchmark datasets) and compared with PSO and K-means, where the proposed GSA-HS was found to provide better results.

A Cuckoo Search Clustering Algorithm (CSCA) was proposed by Goel et al. (2011) and found to yield good results on the benchmark dataset. Based on the results, the proposed CSCA was validated for water body extraction on two real-time remote sensing satellite-image datasets, which on its own, is a complex problem. The CSCA depends on the Davies-Bouldin index (DBI) as a fitness function, while a method for the generation of new cuckoos was introduced in the algorithm. Conceptually, the resulting algorithm was simpler, required less parameter compared to the other nature-inspired frameworks, and yields good results after some parameter tuning,

A quantum-based PSO algorithm (QPSO) was proposed by Sun et al. (2012) for cluster analysis of gene expression database. The algorithm was based on an improved functional flow based approach through QPSO algorithm for automatically finding the optimum threshold when calculating the least similarity between modules. Bridging nodes were also considered to improve the clustering outcome. The performance of the algorithms was tested on the Munich Information Center for Protein Sequences (MIPS) PPI datasets and shown to have a better performance compared to the functional flow method in terms of the number of matched clusters and accuracy.

A novel PSO algorithm inspired by the flocking and schooling behaviours of birds and fishes was developed by Cura (2012) to solve clustering problems. Unlike any other approach, the PSO can be applied with both unknown and known number of clusters. The proposed PSO was confirmed to be computationally effective, robust, easy to tune, and tolerable compared to the other approaches.

The Firefly algorithm (FA) was proposed by Yang based o inspiration from the rhythmic flashes of light by fireflies (Yang, 2009) . The performance of the algorithm was evaluated for clustering purposes on the UCI datasets. The movement of the fireflies is determined by the intensity of light emitted by the adjacent fireflies; those with weaker light intensities are attracted to those with a higher light intensity. (Hassanzadeh & Meybodi, 2012) successfully applied the FA as a clustering algorithm for image segmentation.

A novel algorithm which incorporated ACO with kernel Principal component analysis (KPCA) was proposed by Zhang and Cao (2011). In the proposed algorithm, efficient data features are computed by applying the KPCA on the dataset while the ACO-based clustering is performed in the feature space.

The BB-BC is one of the recently developed heuristics which can be used to solve search and optimization problems. Its applicability and potential in cluster analysis has been demonstrated via simulation studies which confirmed the BB-BC algorithm as a reliable and suitable data clustering technique. It provides quality clusters (based on the sum of intra-cluster distance) and has a simple structure (Hatamlou et al., 2011a).

The GSA-HS was proposed in 2011 as an efficient framework for cluster analysis based on a heuristic search algorithm and gravitational search. The GSA in the GSA-HS is used to find the near optimal solution for clustering problem while the HS algorithm is for the improvement of the initial solution by searching around it. The performance of the GSA-HS was evaluated on four benchmark datasets and later compared with two other known clustering algorithms which are K-means and PSO (Yin et al., 2011).

A clustering algorithm called Bacterial Foraging clustering (BF-C) was proposed in 2012 for data grouping based on the bacterial foraging behaviour. BF-C is a global optimization-based framework which provides a new perspective towards solving NP-hard problems. However, it is a recent application of the foraging behaviour of bacterial. In this algorithm, the clustering problem is transformed into that of finding the centre of each cluster via the optimization of the objective function. Based on numerical experiments, the BF-C achieved high-quality performance on multi-dimensional real datasets and can detect clusters with different densities and shapes, multi-clusters or isolated points (Wan et al., 2012).

Cura (2012) proposed a new version of the PSO called CPSO for solving clustering problems. This version is applicable when there is a known or unknown number of clusters. The CPSO follows the gbest neighbourhood topology and encodes the cluster centroids in particles. During an optimization procedure, it creates new partitions by removing or splitting clusters until the required number of clusters is achieved (Cura, 2012).

Yan et al. (2012) proposed a hybrid clustering algorithm called Hybrid Artificial Bee Colony (HABC) for data clustering. The HABC is based on the enhancement of the mechanism of information exchange between bees through the introduction of a crossover operator of GA to ABC. The performance of HABC was evaluated on 10 benchmark functions and proved to be significantly improved compared to the normal ABC and most of the benchmarking algorithms. Later, the HABC was used for data clustering on 6 real datasets selected from the UCI machine learning database where it performed compared to the other data clustering approaches (Yan et al., 2012).

Senthilnath et al. (2013) comparatively studied three nature-inspired algorithms, namely GA, PSO, and Cuckoo Search (CS) on clustering problem. During the analysis CS was used with levy flight and the heavy-tail property of levy flight was exploited. The performance of these algorithms was evaluated on three standard datasets and one real-time multi-spectral satellite dataset while the results were analysed using various analytical techniques. The authors concluded that based on the given set of parameters, CS works better for most of the dataset due to the important role played by levy flight.

A new clustering method based on the light flashing pattern of fireflies was proposed by Fister et al. (2013) for solving clustering problems. This proposal was a recast of the work previously done by Łukasik and Żak (2009) for continuous constrained optimisation problems to be applicable to data clustering. The study demonstrated the suitability of the standard FA to cluster arbitrary data and proposed the FA-based clustering algorithm called FClust as a centroid evolutionary-based framework. Thus, the performance of FClust was compared with two centroid evolutionary approaches which are PSO and DE. Each algorithm was evaluated for performance based on two statistical criteria which are TWR and VRC.

In (Hatamlou & Hatamlou, 2013), the PSO is one of the commonest heuristic optimization frameworks which has been successfully used to solve clustering problems. At the early stage of a search process, the PSO converges rapidly, but as it approaches the global optimum, the convergence speed slows down. The PSO may be trapped in local optimum if the local best and global best values are equal to the position of the particle over a given range of iterations. However, this problem has been addressed by the proposal of a two-stage clustering algorithm based on PSO and a heuristic search algorithm (2013). The PSO algorithm is used at the first stage to

produce the initial solution to the problem while a heuristic search algorithm is used and in the second stage to improve the initial solution by searching around it.

Singh and Sood (2013) proposed a hybrid approach to show the swarm behaviour of clusters. They used a Krill herd algorithm to simulate the herding behaviour of each krill. The clusters were discovered using a density-based approach; it was also used to show the regions with sufficiently high-density krill clusters. The minimum distance from each krill to the food source and from high-density of herds were considered as the objective function of the krill movement. The movement of each krill is determined by the random diffusion and foraging movement.

A global optimization algorithm for large-scale computational problems was proposed by Jadidoleslamy (2014). The proposed algorithm is a variant of the PSO but based on a parallel annealing clustering algorithm. It was proposed as a novel algorithm based on a group method and is effective for solving continuous variable problems. The proposed parallel PSO algorithm has less computation time and provides clusters with improved quality. The effectiveness of the algorithm was evaluated on large datasets.

An approach based on the combination of Levy flight with modified Bat algorithm to improve the clustering result has been proposed (Jensi & Jiji, 2015). The proposed approach was tested on ten datasets and the experimental results showed that the proposed algorithm clusters the data objects efficiently. It also illustrates that it escapes from local optima and explores the search space effectively.

Ji et al. (2015) suggested an ABC clustering approach for categorical data by first introducing a one-step k-modes procedure before integrating this procedure with the ABC heuristic to cluster categorical data.

An FA-based GA (FAG) was proposed by Kaushik and Arora (2015) in which the selection of the initial population is from a pool of population using an FA. The proposed FAG was applied to the UCI database and the results were satisfactory and better than that of the basic GA and FA.

A new quantum chaotic cuckoo search algorithm (QCCS) was proposed by Boushaki et al. (2018) for data clustering. The superiority of CS over the conventional metaheuristics for clustering problems has been confirmed by various studies. However, all the cuckoos have a similar search pattern, and this may result to the premature

convergence of the algorithm to local optima. Similarly, the convergence rate of the CS is sensitive to the randomly generated initial centroids seeds. Thus, the authors strived to extend the CS capabilities using nonhomogeneous update based on the quantum theory in a bid to tackle CS clustering problem in terms of the global search ability. They also replaced the randomness at the initialization step with a chaotic map to increase the efficiency of the search process and improve the convergence speed. An effective strategy was further developed for a proper management of the boundaries. The results of the experiments on six common real-life datasets show a significant superiority of the developed QCCS over eight recently developed algorithms, including, hybrid cuckoo search, genetic quantum cuckoo search, differential evolution, hybrid K-means, standard cuckoo search, improved cuckoo search, quantum particle swarm optimization, hybrid K-means chaotic PSO, differential evolution, and GA in terms of external and internal clustering quality.

Alswaitti et al. (2018) developed a new heuristic gravitational-based framework for data clustering with the aim of addressing the excessive centroid movement (due to the accumulation of the centroid velocity history in the gravitational clustering algorithm) to achieve a better trade-off between exploration and exploitation. The initialization step of the proposed algorithm uses the variance and median method to avoid random initialization effects. Then, the accumulated velocity history of a centroid is discarded; hence, during an iteration, only the force exerted by the data points in a cluster is affecting the position of the centroid associated with this centroid.

A combination of K-Harmonic Means with improved cuckoo search algorithm (ICS) and PSO has been proposed by Bouyer and Hatamlou (2018)for the enhancement of the search for solutions and achieving a quick convergence while avoiding local optima entrapment. The standard CSA has a lower convergence speed compared to most of the other evolutionary algorithms like SA and PSO. Hence, ICS was proposed for finding the optimum clusters with fast convergence. The convergence of ICS is enhanced by computing a better radius in each iteration. Furthermore, a good variant of the PSO called MPSO to help the ICS avoid local optimum and avoid fast convergence to local optima. It can, therefore, be argued that the proposed ICMPKHM combines the advantages of ICS and MPSO to achieve efficient data clustering. Another benefit of this combination is that it achieved a stable data-clustering algorithm compared to all the evolutionary-based methods. Generally, the major objective of most clustering

algorithms is to meet the required qualities in clusters such as standard deviation parameters, processing time, F-measure, k-Error, and Error. In this study, the selected test suits include two artificial data sets, UCI real datasets, and 31 standard benchmark functions. The results of the evaluations showed the proposed algorithm to perform better than the other algorithms tested.

A new version of Artificial Bee Colony (ABC) algorithm called History-driven Artificial Bee Colony (Hd-ABC) was proposed by Zabihi and Nasiri (2018) by applying a memory mechanism to improve the performance of ABC. The proposed Hd-ABC uses a binary space partitioning (BSP) tree to memorize useful information of evaluated solutions. By the application of this memory mechanism, the fitness landscape can be approximated before the actual fitness evaluation. Fitness evaluation is a time and cost inefficient process in clustering problem, but the use of a memory mechanism has significantly reduced the number of fitness evaluations and facilitated the optimization process via the estimation of the solutions' fitness value instead of estimating the actual fitness values. The proposed data clustering algorithm was applied on 9 UCI datasets and 2 artificial datasets and both the statistical and experimental outcomes showed the proposed algorithm to perform better than the original ABC, its variants, and the other recent clustering algorithms.

Elephant Herding Optimization (EHO) was proposed as a nature-inspired algorithm by Jaiprakash and Nanda (2019). The algorithm combined swarm intelligence (the life pattern of elephants living in groups) and evolutionary algorithm (reproduction to create baby elephant). The algorithm has both exploitation (clan updating operator) and exploration (separating operator) capabilities, making it a potential optimization algorithm. The EHO was suitably formulated for clustering analysis by reducing the intra-cluster distance as a cost function. The performance of the algorithm was evaluation based on simulations on three synthetic and six benchmark datasets and compared with RCGA, PSO, and K-means algorithm where it showed a superior level of accuracy in the form of box plots.

The computational time of EHO was also observed to be higher than K means but lower than PSO and RCGA. A comparison of all the clustering algorithms is presented in Table 2.1.

Table 2.1 The analysis of existing clustering algorithms

NO.	Algorithm	Method	Type	Approach	Merits	Demerits
1.	(Van der Merwe & Engelbrecht, 2003b)	PSO	Standard	A PSO-based data clustering method for large dataset clustering.	Have better convergence to lower quantization errors.	Takes more time to converge and also suffers problem to stuck at some local solution.
2.	(Maulik & Bandyopadhyay, 2000)	GA	Standard	Exploited the searching capability of GA to search for suitable cluster centers within the feature space.	The good ability for exploration and global search.	Very sensitive to parameter setting.
3.	(Karaboga & Ozturk, 2011)	ABC	Standard	(ABC) algorithm for benchmark problems data clustering.	The good ability in local search for exploration.	Weak in exploration.
4.	(Zhang et al., 2010)	ABC	Standard	An artificial bee colony algorithm is developed to solve clustering problems which are inspired by the bees' forage behavior.	Good ability to produce a good suboptimal solution.	Lacking in structures which can provide every individual of population with simple memory mechanism.
5.	(Wan et al., 2012)	BFO	Standard	A novel clustering framework based on the Bacterial Foraging (BF) mechanism.	Good ability for exploration and avoiding local optima.	Weak ability to perceive the environment which may be effect the solution quality.
6.	(Senthilnath et al., 2011)	FA	Standard	They measured the performance of FA with respect to supervised clustering problem and the results show that the algorithm is robust and efficient.	Good ability for exploration.	Computational time is high due to too many attractions.
7.	(Hatamlou et al., 2011a)	BB-BC	Standard	An approach that demonstrates the effectiveness and applicability of the BB-BC algorithm in cluster analysis.	Provides high-quality clusters with respect to intra-cluster distance; has a simple structure.	Sensitive to parameter setting.

Table 2.1 Continued

NO.	Algorithm	Method	Type	Approach	Merits	Demerits
8.	(Hatamlou et al., 2011b)	GSA	Standard	Demonstrate the clustering capability of the GSA.	Good ability for exploration and avoiding local optima.	Sensitive to parameter setting.
9.	(Hatamlou, 2014)	Heart	Standard	A new clustering algorithm based on the action of the heart and circulatory system.	Can be easily implemented, few parameters can be manipulated.	Quality of cluster is not very good.
10.	(Saida et al., 2014)	CS	Standard	A new cuckoo search optimization-based algorithm for data clustering.	It is easy to implement, and it manipulates a few parameters.	suffers from the problem of slow convergence and also stuck at local minima.
11.	(Bagirov & Yearwood, 2006)	TS	Standard	TS-Clustering is developed to explore the clustering result	Good ability for exploration.	The number of clusters should be known a priori.
12.	(Hatamlou, 2013)	BH	Standard	A new Blackhole principle-based heuristic optimization algorithm.	Simple structure, easy implementation, and free from issues of parameter tuning.	Require better control for exploration and exploitation to prevent parameter convergence.
13.	(Yan et al., 2012)	HABC	Hybrid	Introduced the crossover operator of GA into ABC to enhance the exchange of information between bees.	Good ability for exploration and avoiding local optima.	Cannot work properly in high dimensional data.
14.	(Senthilnath et al., 2013)	CS-LV	Modified	Exploited the heavy-tail property of levy flight in a hybrid combination with Cuckoo search.	Good ability for exploration for better clustering quality.	It needs more evaluation test.

Table 2.1 Continued

NO.	Algorithm	Method	Type	Approach	Merits	Demerits
15.	(Bouyer et al., 2015)	HCSDE	Hybrid	Improved the capability of the CS algorithm in obtaining a better convergence seed with high precision in a short time. Also improved the CS algorithm capability in solving the number of functional evaluations.	Improves the quality of clustering.	Depends on many parameter.
16.	(Abualigah et al., 2017a)	KHA-HS	Hybrid	A data clustering algorithm based on a hybrid combination of krill herd algorithm (KHA) and harmony search (HS) to improve data clustering.	Good computational efficiency, easy to implement, improves the method for best value detection.	Depends on many parameter.
17.	(Yang et al., 2009)	Harmony-PSO	Hybrid	A PSO and K-harmonic mean-based approach for the data clustering	Fully utilize the merits of both algorithms.	Cannot work properly in high dimensional data.
18.	(Hatamlou et al., 2011c)	GSA-HS	Hybrid	Used gravitational search algorithm to establish the initial solution for clustering problems; then, used a heuristic search algorithm to search around this established to improve its quality.	Improves the quality of clustering.	Sensitive to parameter setting.
19.	(Jensi & Jiji, 2016)	IKH	Modified	Proposed an improved krill herd by introducing a global exploration operator. These modifications improved the ability of the IKH algorithm to quickly converge to optimal solutions.	Improves the quality	Runtime time and not easy to implement.
20.	(Jensi & Jiji, 2015)	MBA-LF	Modified	A hybrid combination of modified bat algorithm with levy flight for efficient data clustering.	Good ability for exploration	It needs more evaluation test.

Table 2.1 Continued

NO.	Algorithm	Method	Type	Approach	Merits	Demerits
20.	(Jensi & Jiji, 2015)	MBA-LF	Modified	A hybrid combination of modified bat algorithm with levy flight for efficient data clustering.	Good ability for exploration	It needs more evaluation test.
21.	(Sun et al., 2006)	QPSO	Hybrid	Explored the applicability of the Quantum-behaved PSO for data clustering.	Improves the quality	Quality of cluster is not very good.
22.	(Chikhi et al., 2014)	GQCS	Hybrid	A new data clustering method based on a hybrid combination of GA and CSA.	The approach fully uses the merits of all algorithms.	Depends on many parameters. Quality of cluster is not very good.
23.	(Boushaki et al., 2018)	QCCS	Hybrid	A new quantum chaotic CSA for data clustering.	The approach fully uses the merits of all algorithms.	Cannot work properly in high dimensional data.
24.	(Emami et al., 2015)	ICAKHM	Hybrid	A method based on the hybrid combination of K-harmonic means algorithm and a modified imperialist competitive algorithm (ICA).	Can reduce the intracluster distance in clustering problems.	This algorithm is generally unstable and its result may or may not be improved.
25.	(Sun & Peng, 2014)	PSO-GSA	Hybrid	A clustering algorithm based on the combination of PSO with GSA for clustering.	Combined the exploitation ability of PSO with the exploration ability of GSA to update velocity equations and improve the clustering performance.	Cannot work properly in high dimensional data.

Table 2.1 Continued

NO.	Algorithm	Method	Type	Approach	Merits	Demerits
26.	(Bouyer, 2016)	KHM-IPSO	Hybrid	A clustering algorithm based on the combination of KHM, IPSO and CS for large data datasets in a faster and accurate manner compared the other algorithms.	Addressed the initialization sensitivity problem of KHM and achieved good convergence to the global optimum.	Its runtime compared to KHM due to using PSO and Cuckoo search optimization too long.
27.	(Mageshkumar et al.)	ACO–ALO	Hybrid	Clustering algorithm for solving data clustering problems. Incorporates Cauchy's mutation operator to avoid the problem of local minima traps.	Reduced intra cluster distance in clustering problems.	Depends on many parameter.
28.	(Yin et al., 2011)	IGSAKHM	Hybrid	Clustering algorithm based on a hybrid combination of KHM and IGSA algorithms.	Can converge quickly to local optima.	Cannot work properly in high dimensional data.
29.	(Ilango et al., 2018)	Hd-ABC	Modified	A new variant of Artificial Bee Colony (ABC) algorithm called History-driven Artificial Bee Colony	Reduced the intracluster distance in clustering problems.	Depends on many parameter.
30.	(Bouyer & Hatamlou, 2018)	ICMPKHM	Hybrid	A clustering method based on a hybrid combination of the improved cuckoo optimization and modified PSO algorithms.	Can solve the local optima problem of KHM and can significantly improve its efficacy and stability.	Cannot work properly in high dimensional data.

2.3 Metaheuristic Algorithms

Optimization is the process of finding the best solution from some sets of available alternatives solutions under certain constraints (Alia and Mandava 2011). This can be achieved by minimizing/maximizing the objective function of the given problem. Optimization techniques are used in real life problems such as scheduling, resource allocation, and many other computer science applications. A large number of algorithms have been developed to solve the optimization problems in recent years. These are broadly classified into two main categories: deterministic and stochastic (Yang, 2008). The former ones produce the same set of solutions if the iterations start with initialization of the same parameters. These are local search algorithms and have the tendency of being easily trapped in a local minimum. Most of the deterministic algorithms used the gradient information. The gradient-based algorithm uses the function values and their derivatives, and work well for smooth unimodal functions (Yang, 2010c). However, it fails on discontinuous functions. To solve this problem, non-gradient-based or gradient-tree algorithms are used as they require only fiinction values.

The stochastic algorithms produce different solutions even if initialized with the same set of parameters (Yang, 2010a). However, these are able to converge to the same optimal solution within a given accuracy. Generally, they are classified into two types: heuristic and metaheuristic. Heuristic means 'to find' or 'to discover by trial and error' (Yang, 2008). They provide quality solutions for an optimization problem within a reasonable amount of time. However, there is no guarantee that optimal solutions are achieved. Further development over the heuristic algorithms is known as meta-heuristic algorithms. The word Meta means 'higher level' and they usually perform better than heuristic algorithms.

All the metaheuristic algorithms use randomization and local search. Randomization helps in avoiding the solutions being stuck in local optima. Intensification and diversification are the two major components of metaheuristic algorithms (Yang, 2008). Intensification intends to search around the current best solutions and selects the best solution. Whereas, diversification avoids solutions being trapped in the local optima and hence increases the diversity of solutions. The best combination of these may ensure the achievement of global optimality. The most

popular metaheuristic algorithms are Genetic Algorithm (GA), Differential Evolution (DE), Particle Swarm Optimization (PSO), Ant Colony Optimization (AGO), Artificial Hone Bee Algorithm (ABC), Firefly Algorithm (FA), Cuckoo Search Algorithm (CS), Grey Wolf Optimization (GWO), etc.

Metaheuristics are the key strategy for the modification and updating of the other heuristically-produced solution. Such solutions are mainly generated when searching for the local optimal (Jourdan et al., 2009).The suffix "meta" in the name is generated from a Greek word which means "upper level methodology"; they are generally better than the simple heuristic approach in performance. Metaheuristics are a conceptual set of all the heuristic approaches which is used to establish the optimal solution of a combinatorial optimization problem. Additionally, metaheuristics use certain balances between randomization and local search to find the near and optimal solutions to a given problem. Local search is a generalized method of finding high quality solutions to hard/complex combinatorial optimization problems within a reasonable length of time. Basically, it is an iterative-based search technique used for the diversification of the neighbouring solutions in a bid to enhance the current solution by local changes (Mirjalili, 2016).

2.3.1 Exploration and Exploitation

The search process of each metaheuristic is dependent on the trade-off between its exploration (diversification) and exploitation (intensification) capabilities. Metaheuristics depends on the local search information to establish better solutions of problems. With too much exploitation capability, a metaheuristic may converge prematurely and often results in a local optimum or a wrong solution. It will also reduce the chances of reaching the global optimum solutions of a complex problem. Hence, there is a need to ensure that there is a fine balance between the intensification and diversification capabilities of metaheuristics.

Metaheuristic techniques can be combined with other concepts to find the best solutions to complex combinatorial optimization problems. Metaheuristics achieves good solutions due to the convergence of all the identified solutions to the optimal solution; the diversification capability, via randomization, ensures that the solution is not trapped at local minima and equally increase the range of the solutions to hard

problems. The provision of the solutions to multi-objective optimization problems is generally difficult; however, a good combination of the explorative and exploitative capabilities of metaheuristics ensures the achievement of the global solution to hard or complex optimization problems and always provides a way of solving large-sized population- based problems by delivering the right solutions in a reasonable amount of time (Greiner et al., 2018).

Metaheuristics are a high-level approach for the diversification of a search space by using different algorithmic methods. It is greatly important that there should be a dynamic balance between the explorative and exploitative capabilities of metaheuristics. Diversification generally refers to the exploration of the search space while exploitation refers to the intensification of the accumulated search experience.

2.3.2 Types of Metaheuristics

Several nature-inspired metaheuristics have been developed in the last two decades and applied to several real-life situations. Metaheuristics are used in recent years to solve several unsupervised optimization problems. People easily picks a metaheuristic method to solve any unsupervised optimization problem at hand (Yang, 2010c) because they guarantee optimal solutions and explores the entire search space with the progress in generations.

Many classification criteria may be used for metaheuristics. This may be illustrated by considering the classification of metaheuristics in terms of their features with respect to different aspects concerning the search path they follow, the use of memory, the kind of neighbourhood exploration used or the number of current solutions carried from one iteration to the next. The metaheuristic classification, which differentiates between Single-Solution Based Metaheuristics and Population-Based Metaheuristics, is often taken to be a fundamental distinction in the literature. Roughly, speaking, basic single-solution based metaheuristics are more exploitation oriented whereas basic population-based metaheuristics are more exploration oriented. The type of meta-heuristic is only a single-solution based metaheuristics and population-based metaheuristics.

2.3.2.1 Single-Solution Based Metaheuristics

In this section, we outline single-solution based metaheuristics, also called trajectory methods. Unlike population-based metaheuristics, they start with a single initial solution and move away from it, describing a trajectory in the search space. Some of them can be seen as ‘‘intelligent’’ extensions of local search algorithms. Trajectory methods mainly encompass the simulated annealing method, the tabu search, the variable neighborhood search, the guided local search and the iterated local search (Metropolis et al., 1953).

- **Simulated Annealing Method**

The origins of the Simulated Annealing method (SA) are in statistical mechanics (Metropolis algorithm)(Černý, 1985). SA is inspired by the annealing technique used by the metallurgists to obtain a ‘‘well ordered’’ solid state of minimal energy (while avoiding the ‘‘metastable’’ structures, characteristic of the local minima of energy). This technique consists in carrying a material at high temperature, then in lowering this temperature slowly. SA transposes the process of the annealing to the solution of an optimization problem: the objective function of the problem, similar to the energy of a material, is then minimized, by introducing a fictitious temperature T, which is a simple controllable parameter of the algorithm.

- **Tabu Search**

Tabu Search (TS) was formalized in 1986 by Glover (Glover, 1986) . TS was designed to manage an embedded local search algorithm. It explicitly uses the history of the search, both to escape from local minima and to implement an explorative strategy. Its main characteristic is indeed based on the use of mechanisms inspired by human memory. It takes, from this point of view, a path opposite to that of SA, which does not use memory, and thus is unable to learn from the past.

- **Variable Neighborhood Search**

Variable Neighborhood Search (VNS) is a metaheuristic proposed by Hansen and Mladenovic (Hansen & Mladenović, 1997). Its strategy consists in the exploration of dynamically changing neighborhoods for a given solution. At the initialization step, a

set of neighborhood structures has to be defined. These neighborhoods can be arbitrarily chosen, but often a sequence N_1; N_2; . . . ; $N_{n_{max}}$ of neighborhoods with increasing cardinality is defined. In principle, they could be included one in the other ($N_1 \in N_2 \in$ . . . $\in N_{n_{max}}$). However, such a sequence may produce an inefficient search, because a large number of solutions can be revisited. Then an initial solution is generated, and the main cycle of VNS begins. This cycle consists of three steps: shaking, local search and move. In the shaking step, a solution Ś is randomly selected in the *n*th neighbourhood of the current solution s. Then, Ś is used as the initial solution of a local search procedure, to generate the solution Ś. The local search can use any neighborhood structure and is not restricted to the set N_n, n = 1, . . . , $N_{n_{max}}$. At the end of the local search process, if Ś is better than s, then Ś replaces s and the cycle starts again with $=$ 1. Otherwise, the algorithm moves to the next neighborhood $n + 1$ and a new shaking phase starts using this neighborhood.

❖ **The Guided Local Search**

In GLS, this memory is called an augmented objective function (Voudouris, 1997). Indeed, GLS dynamically changes the objective function optimized by a local search, according to the found local optima. First, a set of features $ft_n, n = 1, \ldots, n_{max}$ has to be defined. Each feature defines a characteristic of a solution regarding the optimization problem to solve. Then, a cost c_i and a penalty value p_i are associated with each feature. For instance, in the traveling salesman problem, a feature ft_i can be the presence of an edge from a city A to a city B in the solution, and the corresponding cost c_i can be the distance, or the travel time, between these two cities. The penalties are initialized to 0 and updated when the local search reaches a local optimum.

❖ **The Iterated Local Search**

ILS is a metaheuristic based on a simple idea: instead of repeatedly applying a local search procedure to randomly generated starting solutions, ILS generates the starting solution for the next iteration by perturbing the local optimum found at the current iteration (Ebert et al., 1994). This is done in the expectation that the perturbation mechanism provides a solution located in the basin of attraction of a better local optimum. The perturbation mechanism is a key feature of ILS: on the one hand, a too weak perturbation may not be sufficient to escape from the basin of attraction of the

current local optimum; on the other hand, a too strong perturbation would make the algorithm similar to a multi-start local search with randomly generated starting solutions.

2.3.2.2 Population-Based Metaheuristics

Regarding the population-based metaheuristics (P-metaheuristics), they are iterative processes that strives to enhance the number of solutions (Mitchell et al., 1994). In these metaheuristics, the population is first randomly initialized before generating a new population of potential solutions which, based on certain selection criteria, could be integrated into the current solution. Upon meeting a certain termination criterion, the search process is terminated. P-metaheuristics are exploration-based, meaning that they encourage a greater diversification of the search process compared to the single solution metaheuristics. As such, P-metaheuristics, such as those based on evolutionary algorithms (EA), artificial immune systems (AIS), and swarm intelligence (SI) paradigms have grown in popularity with respect to providing solutions to clustering problems.

- **Evolutionary Computation (EC)**

  Evolutionary Computation (EC) is the general term for several optimization algorithms that are inspired by the Darwiniann principles of nature's capability to evolve living beings well adapted to their environment. Usually found grouped under the term of EC algorithms (also called Evolutionary Algorithms (EAs)) (Vent, 1975), are the domains of genetic algorithms (Holland, 1975), evolution strategies (Vent, 1975) , evolutionary programming (Fogel et al., 1966), and genetic programming (Koza & Koza, 1992).

- **Genetic Algorithms (GA)**

Based on population genetics and Darwin's theory of natural selection, genetic algorithms are a type of evolutionary computing that solves problems by probabilistically searching the solution space (Holland, 1975). In contrast to most algorithms that work by successively improving a single estimate of the desired optimum via iterations, GA's work with several estimates at once, which together form

a population. Given an initial population of individuals representing possible solutions to the problem, genetic algorithms simulate evolution by allowing the most fit individuals to reproduce to form subsequent generations. After several generations, convergence to an optimal solution is often accomplished. Determining the fitness of an individual is problem dependent and the fitness function usually incorporates a priori knowledge of the desired optimum.

❖ **Swarm Intelligence (SI)**

Swarm Intelligence (SI) is an innovative distributed intelligent paradigm for solving optimization problems that takes inspiration from the collective behavior of a group of social insect colonies and of other animal societies(Engelbrecht & Wiley, 2006) . SI systems are typically made up of a population of simple agents (an entity capable of performing/executing certain operations) interacting locally with one another and with their environment. These entities with very limited individual capability can jointly (cooperatively) perform many complex tasks necessary for their survival. Although there is normally no centralized control structure dictating how individual agents should behave, local interactions between such agents often lead to the emergence of global and self-organized behavior. Several optimization algorithms inspired by the metaphors of swarming behavior in nature are proposed. Ant colony optimization, Particle Swarm Optimization, Bacterial foraging optimization, Bee Colony Optimization, Artificial Immune Systems and Biogeography-Based Optimization are examples to this effect.

➢ **Particle Swarm Optimization (PSO)**

PSO exploits a population of individuals to probe promising regions of the search space. PSO follows a stochastic optimization method based on Swarm Intelligence (SI) (Kennedy & Eberhart, 1997). The fundamental idea is that each particle represents a potential solution, which it updates according to its own experience and that of neighbours. The PSO algorithm searches in parallel using a group of individuals. Individuals or particles in a swarm, approach to the optimum through its present velocity, previous experience and the experience of its neighbours. PSO searches the problem domain by adjusting the trajectories of moving points in a multidimensional space. The motion of individual particles for the optimal solution is

governed through the interactions of the position and velocity of each individual, their own previous best performance and the best performance of their neighbours.

- **Ant Colony Optimization (ACO)**

The ant colony is an adaptive nature-inspired meta-heuristic optimization method introduced by Dorigo (Colorni et al., 1992; Dorigo et al., 1996). The ant optimization paradigm was related to the behaviour of real ants. Ethnologists have studied how blind animals, such as ants, can establish the shortest paths from their nest to food sources. Pheromone is an aromatic material. Ants lay down this in some quantity when they are on their way to food and are back to the home. The possibility of ants to follow a particular pheromone path depends upon the pheromone intensity. Researchers have shown that ants identify the shortest path from their home to a food source by pheromone trail following behaviour. Pheromone is the source of information among individual ants regarding paths. A moving ant lays some pheromone on the ground, thus making the path. The pheromone gradually dissipates over time. It is reinforced, as other ants use the same path. Therefore, efficient trails increase their pheromone level over time while the poor ones reduce to nil.

- **Gravitational Search Algorithm (GSA)**

Gravitational Search Algorithm (GSA) was first introduced by (Rashedi et al. 2009), which is inspired by the laws of gravitation and motion. GSA could be considered as a collection of agents having masses proportional to their fitness function value. All masses attract each other through gravitational force. The gravitational force is directly proportional to product of masses and inversely proportional to distance between masses. A heavier mass which has small distance generates more attraction force. The heavier masses are possibly close to the global optimum.

- **Grey Wolf Algorithm**

Grey wolf algorithm (GWA) is an efficient optimization algorithm that is inspired by behaviours of grey wolves (Mirjalili et al., 2014). It mimics the leadership hierarchy and hunting mechanism of grey wolves in nature. Four types of grey wolves such as alpha, beta, delta and omega are employed for simulating the leadership hierarchy. In addition to these, three main steps of hunting, searching for prey,

encircling prey and attacking prey are used. In GWA, the search process starts with random population of grey wolves (candidate solutions). During the course of iterations, alpha, beta and delta wolves estimate the approximate position of the prey. Each candidate solution updates its distance from the prey. Candidate solutions tend to converge towards the prey. The GWA produces the best candidate solution at end of iteration.

2.3.2.3 Multi-Population-based metaheuristics

Many real-world problems have been recently solved using many natural or biological inspired population-based techniques (Yildiz, 2012, 2013a, 2013b). However, the last decade witnessed the use of population-based methods to solve dynamic optimization problems (Blackwell & Branke, 2004; Branke, 1999; Cruz et al., 2011; Yang et al., 2007). Being that population-based methods deal with a set of solutions scattered over the whole solution space, this feature helps them to monitor changes by allocating each solution from the population to a different section of the solution space (Yang et al., 2007). When solving dynamic optimization problems, the major problem encountered is the controlling of the solution diversity; hence, the combination of the population-based techniques with several mechanisms have been proposed to ensure population diversity (Branke, 1999). For instance, Self-Organizing Scouts (SOS) has been proposed by Branke (1999) as a multi-population evolutionary algorithm for solving the Moving Peaks Benchmark (MPB). This algorithm divides the population into 2 subgroups (small and large). For the smaller populations, the goal is to monitor the most promising peaks over time while the larger population will keep searching for new peaks. The performance of the proposed algorithm was positive when applied to the MPB.

A multiswarm PSO was proposed by (Blackwell & Branke, 2004, 2006) in which the swarm was partitioned into a mutable interacting subset of swarms which interacts locally by exchanging algorithmic parametric information. However, the global interaction is based on the anti-convergence mechanism that strived to eliminate the worst swarm from its peak in order to re-initialize it in the solution space. The results of the proposed algorithm, when applied on the MPB, were positive. A clustering PSO for the MPB was developed by (Yang & Li, 2010). This algorithm

tracks and locates multiple peaks using a hierarchical clustering method. The algorithm also achieved positive results.

A cooperative PSO approach (CPSO-S) was suggested by (Van den Bergh & Engelbrecht, 2004). This approach splits the solution vector into sub-vectors which will individually be optimized by a specific swarm. The best solutions discovered by each swarm is used to build the complete solution vector. This approach is based on an initially proposed method using a GA (Potter & De Jong, 1994). Then in (Baskar & Suganthan, 2004) a method that involves 2 swarms parallelly searching for a solution using frequent passing of information was developed by . A Master-Slave multi-population for PSO was proposed by (Niu et al., 2005) to ensure the particles' diversity. Several other works have also been reported in this regard, such as (Yildiz, 2013a) where cooperating swarms which use a diversity strategy to exchange information are used.(Hongwei et al., 2010) used a fuzzy multi-population cooperative GA for multi-objective transportation (El-Abd et al., 2010) suggested a discrete cooperative PSO for FPGA placement and (El-Abd & Kamel, 2010) proposed a cooperative PSO with the migration of heterogeneous probabilistic models. A study presented by (Akbari & Ziarati, 2011) presented a cooperative approach to bee swarm optimization, while (Guo et al., 2011) proposed a multi-population cooperative cultural algorithm which brought the competition cooperative GA into the population space of the cultural algorithm. These are mainly stochastic search techniques that are based on the principles of the individual and collective behaviour of insect swarms (Zhou et al., 2019). They are robust, efficient, and adaptive search methods that produces near to optimal solutions and are endowed with a great implicit parallelism. Contrarily, data clustering can be formulated as a global optimization problem.

2.4 Metaheuristics based Levy Flight (LF)

- The Basic Concepts of Levy Flight (LF)

Levy flight (LF) refers to a group of non-Gaussian random processes in which the distribution of its stationary increments follows a Levy stable distribution (Haklı & Uğuz, 2014). Paul Pierre Levy is a French mathematician who provided the first insight into Levy motion (Lévy, 2001); hence, the term 'Levy' in LF was taken after his name. The term 'flight' in this concept is taken as the maximum distance (in a straight line)

cover by an object in motion between 2 points without any halt or directional variation. Levy walk and LF are used interchangeably in the literature. In scenarios where birds or animals have little or no information of where to find their food, it has been observed that LF can efficiently provide such information with randomly distributed targets compared to BM (Brownian motion) which is efficient in scenarios with enough and more predictable targets (Reynolds & Rhodes, 2009; Yang & Deb, 2010).

Levy flight (Chechkin et al., 2008) can be defined as a type of arbitrary processes that is characterized by a jump size that adheres to the levy probability distribution function. Its name was derivative of a French mathematician named Paul Pierre Levy. The distribution is simple power-law formula $L(s)|s|^{-1-\beta}$ where $0 < \beta \leq 2$ is an index. The Levy distribution can be defined using an uncomplicated mathematical definition, as seen below (Yang & Deb, 2013):

$$L(s) \sim |s|^{(-1-\beta)}, where\ \beta\ (0 < \beta \leq 2) \qquad 2.11$$

where β and s represents an index and the step length, respectively. This study utilized a Mantegna algorithm for a symmetric Levy stable distribution to generate the sizes of the random steps. The term 'symmetric' in this concept implies that the step size will assume either a positive or negative value. The step length s in the Mantegna's algorithm can be calculated thus:

$$s = u/|v|^{(1/\beta)} \qquad 2.12$$

where u and v are drawn from normal distributions; i.e.,

$$u \sim N(0, \sigma\_u^2), \quad v \sim N(0, \sigma\_u^2) \qquad 2.13$$

Where

$$\sigma_u = \frac{\tau(1+\beta)\sin\frac{\pi\beta}{2}}{\tau[\left(\frac{1+\beta}{2}\right)\beta 2^{\frac{\beta-1}{2}}}, \quad \sigma_v = 1 \qquad 2.14$$

The distribution for s follows the anticipated Levy distribution for $|s| \geq |s\_0|$, where s_0 represent the least step length and $\tau(.)$ represent the Gamma function which is estimated thus:

$$\tau(1+\beta) = \int_0^{\infty} t^{\beta} e^{-1} dt \quad 2.15$$

The Levy distribution is used to generate the step sizes in the proposed technique. This is aimed at exploiting the search area. The step sizes are calculated thus:

$$step(t) = 0.01 \times s(t) \times rand(0,1) \quad 2.16$$

where t represents an iteration counter, $s(t)$ is estimated as shown in Equation (2.12) using Levy distribution, while $rand(0,1)$ is a random value ranging from [0,1].

The step sizes in the Levy flights are too aggressive; this implies that they can often generate new solutions which are off the domain or on the boundary. Since the movement equation represented in the BH algorithm is a stochastic method search for new better positions within the search space, therefore, 0.01 multiplier is used in Equation (2.16) to reduce the step sizes when they get large. The positions of the stars are updated in the LBH as follows:

$$x_t(t+1) = x_t(t) + (step(t) \times (x_{BH} - x_t(t))) \quad 2.17$$

where x_t is an individual star in iteration t while $step(t)$ is the actual step sizes generated using Equation (10). x_{BH} denotes the current best solution or the black hole.

Levy flight is characterized by an important parameter of β, whereby each star is a solution and an arbitrary number is produced as β between 0 and 2. Its different values may result in dissimilar outcomes. Therefore, larger values of β pose a higher likelihood to result in jumps to unexplored areas (i.e. higher exploration) and avoidance of being trapped in local optimums. However, smaller values will provoke the new positions to be viewed as near the obtained solutions (i.e. higher exploitation). The BH algorithm is particularly well-perceived for its excellent local search ability (Piotrowski et al., 2014), but within the surround of the optimum point, it is characterized by a low convergence rate. This is due to higher exploitation rate compared to the exploration rate.

- **Properties and Attributes of LF**

LF is a tool commonly used to describe abnormal stochastic processes. Mathematically, LF is Markovian and their statistical distribution limit arises from identical independent randomly distributed variables based on the generalized central limit theorem. Some of the characteristics of LF include:

1. Stable distribution

As a stable distribution, the sum of 2 independent random variables with a β stable distribution and index β is also β stable with the same index β. However, this invariance property is not applicable to differing values of β.

2. Infinite mean and infinite variance

LF has an infinite mean and variance. A stable distribution is said to have infinite variance if it has fatter tails than the GD (Yang, 2010a). An infinite variance distribution is mainly characterized by having the average of its independent draws being no less indeterminate than a single draw. The progression of tails with infinite mean and variance to zero is slow while that of the normal distribution is faster.

3. Heavy tailed probability distribution

As a distinct type of generalized RW, the step length in LF during a walk is described by a heavy-tailed probability distribution or LD. Here, a heavy tail implies a gentler fall of the tail of LD compared to a GD (Yang & Deb, 2013); however, more heavy-tailed distribution is exponentially unconstrained.

4. Capability of escaping local optimum

Being that LD variance portrays divergence attributes, there is a chance of having enormous long jumps. On all scales, classical flights are similar as they display numerous smaller jumps intermixed with long searches. For this type of flight, the advantage is that it allows for an efficient and effective search of the distant solution spaces when solving global optimization tasks. It also prevents the algorithm from being trapped at a local minimum especially when the solution space is enormous.

5. Faster algorithmic convergence

The use of Levy Flight (LF) reduces the chances of searching the previously searched areas in the solution space. As such, it improves the algorithmic performance by ensuring that it is not trapped in any local optima so that it can search the unexplored areas of the search space. With LF, the required number of iterations to lunch the program is reduced by a factor of 104 compared to Gaussian distribution (GD), and by a factor of 108 compared to the use of ES combined with LF. This improves the overall algorithmic convergence speed. This is expressed mathematically in the subsequent sections.

6. Random numbers implementation with LF

There are 2 phases of implementing random numbers with LF, the first phase involves the selection of the random direction from a uniform distribution while the second phase considered the generation of the steps that follow the selected Levy Distribution (LD).

7. Search efficiency of LF

LF concept has been observed in the foraging behavior of various species, such as eagle, spider monkey, honeybee, shark, etc. Other physical processes that exhibit LF include the diffusion of fluorescent molecules, as well as the noise and cooling characteristics under a set of conditions. LF-based frameworks have been shown to produce better results when there is no prior information about resource availability compared to the non-LF frameworks. LF-based frameworks also work better in scenarios where targets are hard to detect or where the targets distribution is meager. However, the fine-tuning capability of LF is lower than that of Gaussian distributed in situations with small to mid-range search spaces. For the GD-based algorithms, problem search is controlled by varying the mean and standard deviation (σ), but this mechanism is not applicable due to the infinite variance of LD.

- **LF in metaheuristics**

Most of the latest metaheuristics are based on LF (Fister et al., 2013). Such algorithms have been successfully applied in several science and engineering problems. The use of LF in these metaheuristics is described in the following paragraphs.

The FPA was developed based on inspiration from the pollination event of plants where reproduction occurs via the transfer of pollen grains from the source flower to the recipient flower (Yang, 2012). Several studies have provided the pseudocode and rules of the FPA (Chakravarthy & Rao, 2015; Yang et al., 2013b). The global pollination process involves the carriage of the flower pollen gametes by certain agents (pollinators) over a long distance as they move across distances, exhibiting Levy walk characteristics (Yang et al., 2016).

Regarding the CS algorithm which was presented by Yang and Deb (2009), each egg in a nest is a potential solution while a cuckoo egg denotes a new solution. In the CS, the basic concept is the replacement of the not-so-good solutions with the new and potentially better solutions (Rajabioun, 2011). The mathematical analysis, pseudocode, and governing rules of the CS have been provided by Yang and Deb (2009) . The cuckoos' consecutive steps form an RW process which follows a heavy-tailed power-law step length distribution. Ideally, a cuckoo's egg may resemble that of some bird species, and if this is the case, the cuckoo's egg may less likely be noticed by the host bird. Hence, fitness is a function of the variation in solutions (Chawla & Duhan, 2014; Yang & Deb, 2013). The CS has exhibited a better number of function evaluations in terms of mean and SD compared to GA and PSO on a small set of test functions (Yang & Deb, 2010). As per Rehman et al. (2016), the LF in the CS brings about the possibility of finding all the optima for multimodal functions in a solution space.

There are 2 stages in eagles' foraging activity (Yang et al., 2013a). While searching for food, the eagle flies randomly like an LF and performs LW in the entire space. Upon the detection of a target, the eagle moves to a chase stratagem in order to catch the target. The ES ensures a good mix of exploration with an efficient and intensive local search method. Local minima entrapment is avoided by first executing

an exploratory search before an exploitation search. This process is iteratively repeated. Two different algorithms are used in the ES at different stages and at different iteration times. LF is employed for global exploration to provide enough randomness to explore the solution space efficiently. The second step involves the use of an effective local optimizer to establish the local best using the least number of evaluation functions. This step is faster compared to the global exploration stage. ES has been proven to achieve better efficiency and success rate compared to PSO and other metaheuristics (Yang & Deb, 2010).

The FA was presented by Yang based on inspiration from the light flashing pattern of fireflies. The pseudocode and the idealized rules of the FA were also provided by (Yang, 2010a, 2010b). A firefly's level of attractiveness is a function of its brightness which is dependent on the encoded objective function (Marichelvam et al., 2013).

The BA was developed based on the echolocation pattern of bats (echolocation can be likened to a sonar) (Yang, 2010d). when searching for targets, bats, especially microbats, makes a loud sound (short-pulsed) and wait for the sound to hit the target and return to their ears after a short while. Based on the returned sound and the timing, the bats can determine the distance to the target. In the BA, the echolocation capability of the bats is the objective function that needs to be optimized; hence, optimization frameworks which can imitate this process in establishing the optimal solution to a problem can be formulated (Chawla & Duhan, 2014). The pseudocode of BA and its guiding rules have been provided by Yang (2010d). To generate new solutions, the frequencies, loudness and the bats' pulse emission rates are adjusted, while the acceptability of the proposed solution is a function of its quality or the loudness and pulse rate which are dependent on the closeness of the solution to the global best solution (Yang, 2013).

Richer and Blackwell (2006) used LF in PSO and modified PSO by replacing the particles' motion with LF to provide an effective solution to optimization problems (Chawla & Duhan, 2018; Kennedy & Eberhart, 1995). The performance of the LF-based PSO was evaluated on a set of 9 different benchmark functions where the LF-

based PSO was observed to achieve a better convergence speed compared to the normal PSO due to the fat tail characteristics of its LD.

(Haklı & Uğuz, 2014) this study hybridized PSO with LF to address the problem of local minima and the inability to perform a well global search in the original PSO. The SPSO and LFPSO were compared on 21 benchmark functions with 30 and 50 dimensions where the LFPSO was more robust and produced better average results in most of the tested benchmark functions. The performance of the LFPSO algorithm was also compared to that of other PSO variants. The evaluation of the experimental results also showed the LFPSO to be more efficient compared to the benchmarked PSO variants. However, the proposed LFPSO was also compared with some of the recent and well-known population-based optimization techniques and found to perform better than most of the methods but close to the ABC algorithm.

2.5 Analysis of the Previous Work

The metaheuristic algorithms discussed previously generally have their various strengths and weaknesses. Most of these techniques require many iterations and perform poorly if there is no adequate parameter tuning. Moreover, metaheuristics are designed to search for near-optimal solutions, as these do not have the capacity to provide an optimal solution. Nevertheless, metaheuristics have some advantages that can be applied to a wide range of problems, either independently or in combination with other traditional techniques. Moreover, most metaheuristics have mechanisms for information sharing, which are responsible for enhancing quick convergence. Further, NI inspired optimization techniques provide a very effective way of handling complex problems and are good substitutes for existing techniques that normally get trapped in local optima . In addition, most of these algorithms have the capacity to escape from local optima which provides the ability to locate near-optimal solutions in a reasonable amount of time. Also, most metaheuristics are simple to understand, design and implement. In Table 2.2, some of the specific advantages and disadvantages of these metaheuristic algorithms presented in this study are outlined.

Table 2.2 Summary of the Classification of Snapshot Metaheuristics Agorithms

Algorithm	Advantages	Disadvantages
GA	GA is easy to implement.	GA has high likelihood of getting trapped in the local maxima
	GA has the ability to handle random types of objectives and constraints (Bousquet et al., 2016).	GA does not have a standard method for defining a good fitness function. The best solutions majorly depend on the problem defined fitness function and hence the fitness function must be very accurate (Bousquet et al., 2016).
	GA can be used independently to solve a given problem. It does not depend on other algorithms or heuristics for it to function well	In GA, premature convergence seldom occurs, thus losing the population diversity
	GA can be used to handle problems whose constraints and objective functions are nonlinear or discontinuous	GA does not have a standard termination criterion; neither does it have a standard method for adjusting its parameters. Therefore, parameter fne tuning is necessary for optimal solution realization (Bousquet et al., 2016).
	GA uses simple operators and can be used to solve problems that have high computational complexity, such as the travelling salesman problem and data clustering problem as well (Bousquet et al., 2016).	GA can be time consuming, especially for problems with large number of variables
ACO	The construction process for ACO is inherently parallel, as ant builds solutions independently and simultaneously (Selvi & Umarani, 2010).	ACO probability distribution changes with iteration.
	Distributed computation in ACO evades premature convergence	Although convergence in ACO is guaranteed, convergence time is undefined (Selvi & Umarani, 2010).
	ACO can be used to efficiently handle Travelling Salesman Problem and related problems	It is difficult to theoretically analyse the behaviour of ACO, since ACO is based on sequences of random decisions of different independent artificial ants

Table 2.3 Summary of the Classification of Snapshot Metaheuristics Agorithms

PSO	PSO is very simple to implement (Bai, 2010).	In PSO, solutions are more likely to converge prematurely and consequently loses the population diversity (Bai, 2010).
	PSO is very robustness and has excellent ability to control parameters	PSO suffers from partial optimism
	PSO is computationally efficient when compared with mathematical algorithm and other heuristic optimization techniques	PSO has the potential to easily fall into local optimum in high-dimensional space
	PSO is useful in scientific research and in engineering and therefore has wide coverage in its application to solving real world problems	
GWO	The Grey wolves conclude the hunt by attacking the prey when it ceases to move. It allows the position of its search agents to be updated based on the location of the alpha, beta, and delta; and attack towards the prey(Fister Jr et al., 2013).	It faces the problem of premature convergence due to the problem of stagnation of wolf pack in local optima (Fister Jr et al., 2013).
CSA	CSA has the ability to converge to a true global optimum (Fister Jr et al., 2013).	CSA produces low classification accuracy (Qu & He, 2015).
	CSA can handle local and global search. It makes use of Levy flight as a strategy for global search (Fister Jr et al., 2013)	CSA has low convergence rate (Qu & He, 2015).
FPA	FPA is flexible, simple, easy to implement, has few parameters and can be used to handle both single and multiple objective optimisation problems (Zhao & Zhou, 2016).	It has the problem of slow convergence, low precision and easy to fall into a local optimum (Wang et al., 2017).
ABC	Does not require external parameters such as the crossover rate and mutation rate as in case of GA. Global search ability in the algorithm is implemented by employing neighbourhood based source production mechanism, which is a comparable to mutation process.	ABC algorithm has premature convergence in the later stage of its search and the classification accuracy of its best obtained value may not be high enough to meet the requirements
BH	It has a simple structure and it is easy to implement (Zuwairie et al., 2018).	BH easily fall into local optimum in high-dimensional space (Kumar et al., 2015).

From Table 2.2, it can be seen that the existing metaheuristic algorithms have not coverd all the features. However, all these algorithms do not work properly on high dimensional data and they take more time to converge. This will result undiscovered search space and the algorithm trapped at local minima. It is evident from the summary of the analysis presented that the existing algorithms still need improvements in order to deliver an efficiency algorithm. Moreover, an existing clustering algorithm namely BH, have promising results on normal datasets, but there are still some issues that need to be addressed such as exploration and high dimensionality. In the present research, levy flight on the multi-population black hole (MLBH) has been proposed to overcome these issues. MLBH is an algorithm that combines the levy flight and the multi-population in the black hole algorithm.

BH superiority has been conformed against several state-of-art metaheuristics algorithms (Hatamlou, 2013) The BH algorithm has proven successful in solving data clustering problems even though the evaluation performance showed this approach to perform better than the other metaheuristics in optimization processes (Munoz et al., 2018). Moreover, BH does not require any controlling parameter for balancing the global search and the local search abilities of the algorithm (Ezugwu, 2020), therefore, it has a simple structure and it is easy to implement. These two mentioned characteristics have led the researchers to apply BH algorithm for solving different optimization problems,

Black hole algorithm is weak to perform global search perfectly in the big problem spaces . In other words, the movement of the stars depends mainly on generating step sizes by using a Uniform distribution values, which may lead to generate almost same steps. Additionally, the weak balancing between the exploration and exploitation increases the chances of the algorithm to trap in the local optima (Kumar et al., 2015). Therefore, the absorption process of the BH algorithm should be improved, which is the main contribution of this study.

2.6 Review on BH algorithm

As per (Li & Pei, 2015), the BH can be applied based on the membrane system. (Tsai et al., 2015) introduced the BH algorithm to enhance the clustering speed using both software and hardware. They introduced an input data solution which exceeds the

memory of such a system. To solve this problem, MapReduce Black Hole (MRBH), an efficient clustering algorithm, was presented to leverage the strength of the BH and the MapReduce programming model. Using MapReduce, MRBH will partition a large dataset into numerous small datasets before clustering them in parallel.

The original BH need to be extended for solving combinatorial optimization problems. One of the extensions of BH algorithm is called Binary Black Holes (BBH) and it is used to solve multiple instances of known benchmarks obtained from the O library (Pashaei & Aydin, 2017). Another binary version of BH called BBHA, which is used to solve feature selection problem in biological data (Rubio et al., 2016). The BH has been actively hybridized with other algorithms. For example, the BH is combined with the stars gravities information and applied to unmanned combat aerial vehicle (UCAV) planning (Heidari & Abbaspour, 2014). The BH is also used as an operator and hybridized with GSA in order to prevent facing premature convergence and to improve the abilities of GSA in exploration and exploitation (Doraghinejad & Nezamabadi-pour, 2014; Mohammed et al., 2016).

In another study, the BH is hybridized with a HS algorithm to solve the problem of BH. In this framework, the BH is used to produce an initial clustering solution to a problem while the HS algorithm is applied to improve the solution's quality (Chandrasekar & Krishnamoorthi, 2014; Eskandarzadehalamdary et al., 2014).

In addition, a bisecting k-means algorithm is hybridized with BH to improve the performance of bisecting k-means algorithm (Eskandarzadehalamdary et al., 2014). Moreover, Black Hole Artificial Bee Colony (BHABC) algorithm has been introduced in 2016. In this new algorithm, the BH gives a high exploration ability while maintaining the original exploitation ability of the ABC algorithm (Bansal, 2016).

Genetic algorithm operators have been used to improve the diversity of BH (Yaghoobi & Mojallali, 2016). Other studies have shown that chaotic features can also be used to enhance the performance of BH and as such, a Chaotic Inertia Weight Black Hole (CIWBH) algorithm has been developed (Aslani et al., 2015).

A white hole operator has been introduced in BH to avoid premature convergence in BH (Suad Khairi et al., 2016) In general relativity, theoretically, if black holes exist, then, it should be possible to reverse the equations governing them to get the

opposite of black hole, which is the white hole (Mirjalili et al., 2016). As opposed to black hole agent in the BH, the white hole was assigned to the worst agent in the population.

The BLA was developed based on inspiration from black holes. This algorithm has three main steps; birth of black holes, calculating forces, and hawking radiation. The experimental results on different benchmarks indicate that the performance of the proposed algorithm is better than PSO, AFS and RBH-PSO. Six benchmark functions with different levels of complexity are used to evaluate the performance of the proposed algorithm (Premalatha & Balamurugan, 2015).

The BH is used in engineering design of electromagnetic devices (Bouchekara, 2013). The new optimization technique based on the black hole phenomenon is called the black-hole-based optimization technique. To show the effectiveness of the proposed technique, it has been demonstrated on a magnetizer by optimizing its pole face to obtain a desired magnetic flux density distribution.

The BH has been introduced into the generalized constitutive law civil engineering. The model identification problem is transformed into a parameter back analysis problem which represents a typical and complicated optimization task. The BH algorithm was introduced in this study to improve the efficiency of the conventional optimization method. Its mechanism involved the combination of the generalized constitutive law for an elastic–plastic constitutive model with the BH algorithm. A new back analysis method has been proposed for model identification of rocks lining underground roadways in a coal mine in a bid to solve a parameter back analysis for the elastic-plastic constitutive model (Gao et al., 2016a). The slope instability of embankment is complicated and may develop locally within the embankment, near the facing, or through the foundation soil as local, general, deep-seated, or surficial failure. The BH is used to analyze the stability of one high embankment slope for one airport in the plateau loess area (Gao et al., 2016b).

In the field of operations research, the BH has been employed by Soto et al., in which, they presented two new systems for online control of enumeration strategies based on bat algorithm and BH (Soto et al., 2017b). The BH is used to improve

performance in the exploration of the search tree and updating the enumeration strategy online (Olivares et al., 2016).

Industrially, the design, control, and performance evaluation of induction motors are dependent on circuit parameters. Despite the capability of the conventional methods to produce accurate electrical parameters (such as resistance) measurements, the swarm intelligence-based methods are for real-world optimization problems. (Sharma & Kapoor, 2017) proposed the use of the disrupted Black Hole Artificial Bee Colony (DBHABC) algorithm for the optimization of induction motor parameter estimation.

One of the commonest optimization problems is the set covering problem (SCP). Multiple instances of this problem have been solved using the BH, with known benchmarks obtained from the OR-library (Rubio et al., 2016). Set covering problems were also solved in 2016 by employing a recent nature-inspired metaheuristic based on the black hole phenomena (Soto et al., 2016). A multi-dynamic binary black hole algorithm for resolving the set covering problem has been introduced (García et al., 2017). The same problems were also solved by cuckoo search and BH (Soto et al., 2017a).

A new method of adjusting metaheuristic-based classifiers based on BH algorithm has bee proposed. The aim of the method is to obtain results close to those obtained by using manual noise elimination methods. The proposed method was evaluated using the MAHNOB HCI Tagging Database and the results show the BH to achieve an accuracy of 92.56% over 30 executions when used to optimize the feature vector of the SVM (Munoz et al., 2018).

Lastly, the differential evaluation (DE) algorithm is a population-based meta-heuristic which was developed to address complex real-world optimization problems. However, a variant of DE called Black-hole Gbest DE algorithm (BHGDE) was developed based on the black-hole (BH) phenomenon. With the incorporation of the Black-Hole, the exploration capability of BHGDE was improved while still retaining the original exploitation capability of DE (Sharma et al., 2019).

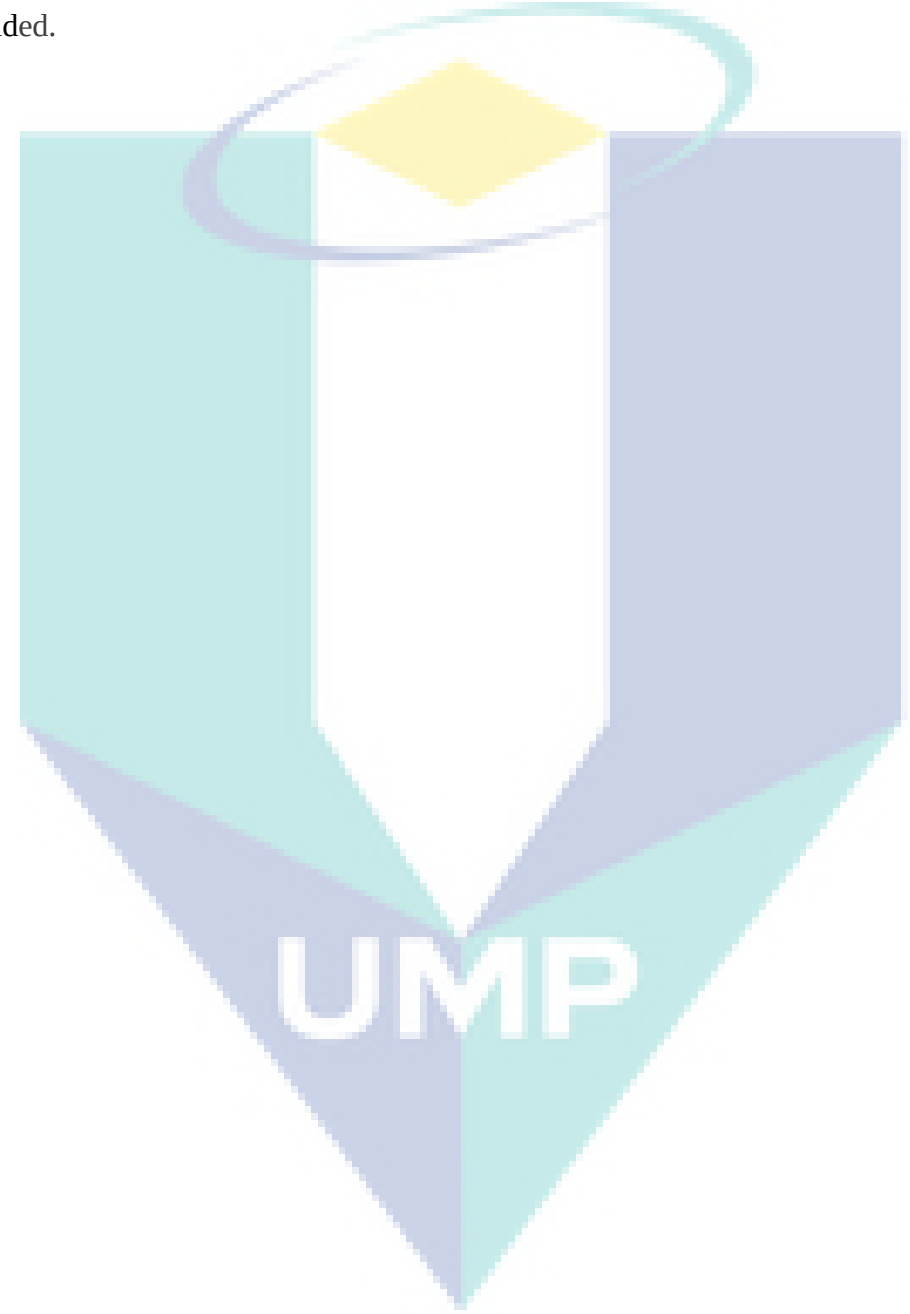
2.7 Gap Analysis

It can be observed from Table 2-1 and Table 2-2 that in all standard meta-heuristics, the number of iterations usually increases in a bid to find an optimal solution due to the random initialization at the initial step. Furthermore, all metaheuristics are characterized by two major problems; first, they require several parameters which are difficult to tune; second, they often require several iterations due to the large search space (rendering them computationally prohibitive). Therefore, most of the recently proposed approaches are just an extension of the capabilities of the original algorithm through hybridization (in most cases) with either K-means algorithm or any other algorithm. However, hybridized algorithms are characterized by high complexity and often requires more computation. To avoid promoting a certain meta-heuristic, other approaches depend on new concepts inspired by recent theories such as chaos and quantum theory to enhance performance. All the meta-heuristics perform differently when they are used to solve different optimization problems. One algorithm may perform better than the other in solving one problem and it may perform unsatisfactorily in other set of problems. We conclude that black hole algorithm is population-based same as particle swarm optimization, firefly, genetic algorithm, BAT algorithm, and other evolutionary methods. It is free from parameter tuning issues like genetic algorithms and others. It does not suffer from premature convergence problem. This implies that black hole is potentially more powerful in solving NP-hard (e.g. data clustering problem). Based on the review in this study, is weak to perform global search completely especially in the big problem spaces due to its limited exploration and exploitation capability (Kumar et al., 2015). The standard version of BH algorithm does not perform the global search well, it is only performed when there is a star with fitness lower than the even horizon (R). Which means only small number of stars are re-generated for the exploration purposes, while in some problems there is a need for exploring the search space more than what BH does.

2.8 Summary

This chapter has introduced and briefly described the basic concepts of meta heuristic, exploration and exploitation, and types of meta heuristic. Special attention has been given to meta heuristic based levy flight, single and multi-population meta

heuristic algorithms. Moreover, clustering since it has been clarified in this chapter as well as clustering optimization, clustering challenges, clustering categories, high dimensional clustering. Next, a survey of the state of art of existing meta heuristics clustering algorithm has been presented. Finally, an overview of the black hole algorithm along with the justification and the gap analysis for black hole algorithm is provided.



CHAPTER 3

RESEARCH METHODOLOGY

3.1 Introduction

In previous chapter, the concept of clustering, the problem of clustering, the metaheuristic, and analysis of the existing data clustering algorithms were introduced. Finally, the BH was explained in detail with the pros and cons then the subsection ends with a review on the black hole algorithms. This chapter presents a full details description of the design methods of the current study. This chapter specifically describes the phases involved for designing clustering algorithm, including its two modified versions, the performance test criteria of the new variant of BH will be presented at the end of this chapter.

As remainder, this chapter describes the research methodology used to develop the proposed algorithm in solving data clustering problem. Section 3.2 offers an elaborated detail of research methodology adopted as well as the tools for testing and validation. Section 3.3 presents a full description of the original black hole algorithm. Section 3.4 then explain the Multiple Levy Flight Black Hole algorithm for high dimensional data clustering. Moreover Section 3.5 presents the high dimensional MLBH followed by Section 3.6 will analysis the test function, datasets and the evaluation measures. Finally, Section 3.9 summarize and conclude this chapter.

3.2 Research Methodology

Most real-world problems can be considered as NP-hard problems because there are no exact methods that could efficiently solve the problem within an acceptable range of computational time. One promising method is metaheuristic since they return good

quality solution quite reasonably (Talbi, 2009). Nonetheless, many metaheuristic methods have a stochastic component, which then leads to different solutions over multiple runs even if the initial solution is the same. At the end, it is very difficult to study their behaviour or characteristic analytically (Bartz-Beielstein et al., 2010). As shown in Figure 3.1, the phases are problem understanding and literature review, design and development and benchmarking and analysis of this study.

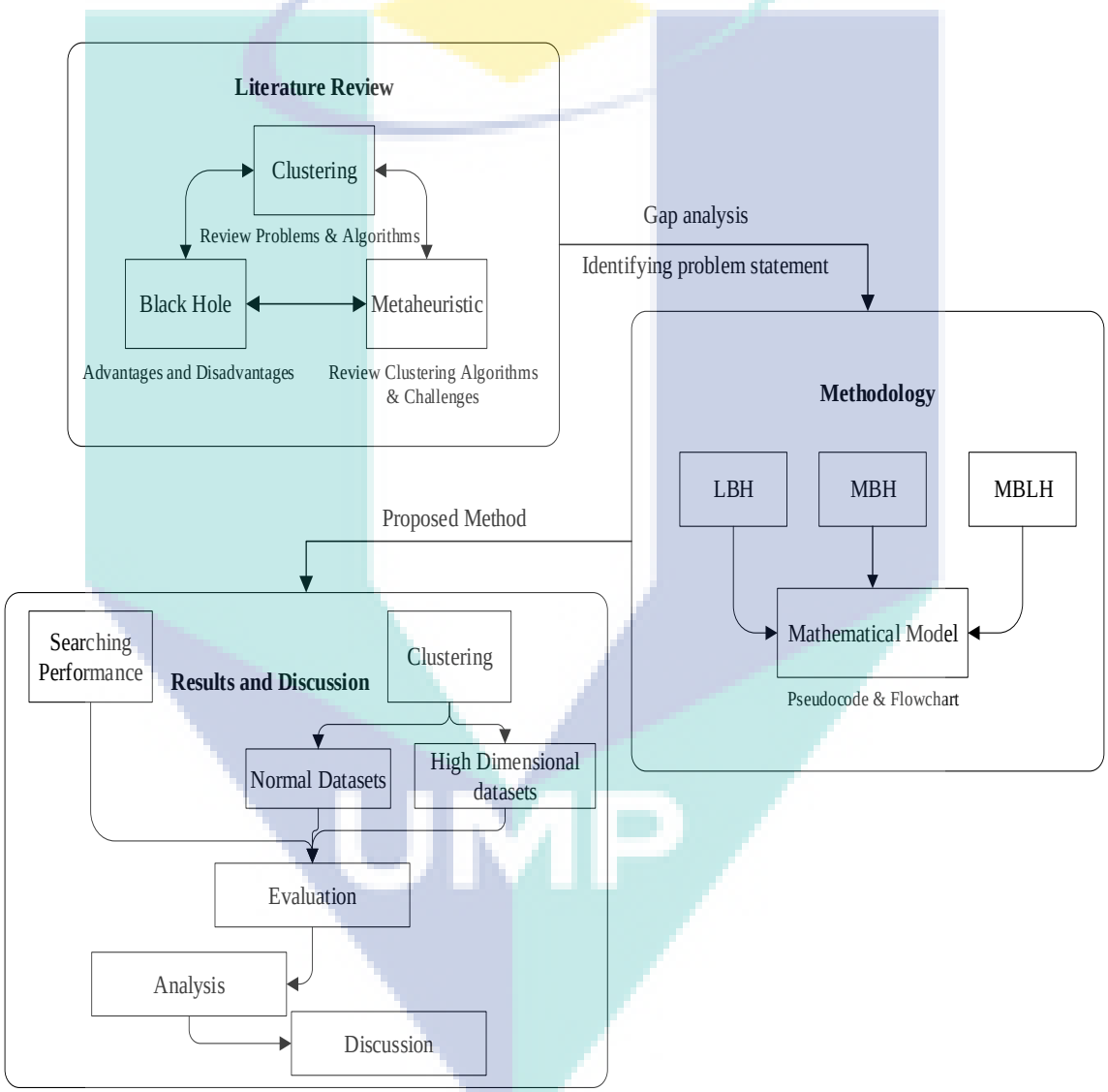


Figure 3.1 The research process

3.2.1 Literature Review Phase

This phase begins with identifying the most relevant works. Particularly, this phase concentrates on understanding the challenges in developing effective and efficient modify BH for data clustering. This is achieved by carrying out a literature review, exploring the existing approaches and finding the research gaps. Then, the proposed solution for the identified gap is presented in the following phase.

3.2.2 The Methodology

The data clustering problem with its challenges have been addressed in the previous phase. In addition, the two major drawbacks of BH algorithm have been identified, they are: The weak exploration capabilities of BH algorithm, and the weak balancing between the exploration and exploitation which may increase the chances for the algorithm to trap in the local optima. In this phase, a new variant of BH algorithm is proposed to overcome the above-mentioned issues for the problems of optimization and data clustering. Moreover, the proposed algorithm called 'Multiple Levy Flight Black Hole (MLBH)' is developed to handle the high dimensional datasets. There are two main differences between MLBH and the original BH, first, the exploration is enhanced by using Levy Flight. While the second, is that the balancing between exploration and exploitation is enhanced by developing a new multi-population architecture. These two modifications are 'Levy Flight Black Hole Algorithm (LBH)' and 'Multiple Black Hole (MBH). The new variant MLBH, with the modifications LBH, and MBH are explained in detail in this chapter. The notations defined are as follows:

i.	μ	is position or shift parameter;
ii.	γ	scale parameter that controls the scale of distribution;
iii.	$rand$	an arbitrary number within the range[0,1];
iv.	$\oplus$	indicates entry-wise multiplications;
v.	S	is a monumental process;
vi.	u and v	are random numbers produced by a normal distribution;
vii.	τ	is the standard gamma function;
viii.	β	Levy flight parameter;
ix.	$x_i(t)$	the location of the star;
x.	x_{BH}	the location of the black hole in the search space;

xi.	c	is a constant;
xii.	N	is the number of stars (candidate solutions) in the population;
xiii.	(SC)	Search Counter;
xiv.	SC_{max}	is the maximum value of SC;
xv.	probability ($prob_{replace}$)	every star death is characterized with a new replacement star;
xvi.	r_g	the ratio is used to mix between the two ways;
xvii.	BH_G	the global best black hole;
xviii.	X_i	represents a new star;
xix.	P	population;
xx.	X_R	represents a randomly selected star from another randomly selected population;
xxi.	F_i	represents the current feature needs to be normalized;
xxii.	Min_i and Max_i	represent the minimum and the maximum value for that feature respectively;
xxiii.	MI	is a well-known filter based feature selection method;
xxiv.	I	represents the value of weight of individual feature;
xxv.	H	denotes the entropy value;
xxvi.	Entropy	is calculated by the summation of all the probability distribution of values of the feature, multiplied by the natural log of those probability distribution;
xxvii.	x	represent a value of the set X;
xxviii.	$p(x)$	represents the probability distribution of x;

3.2.3 Result and Discussion

In this phase, the developed algorithm is verified by testing it based on two aspects; first, the proposed algorithm has been verified based on nine benchmark mathematical functions. Then, the proposed algorithm (MLBH) with the two modifications (LBH and MBH) are evaluated based on two types of benchmark datasets (normal and high-dimensional), in order to demonstrate that the proposed algorithm perform well. The normal datasets are: Iris, Wine, Glass, Cancer, Vowel and Contraceptive Method Choice (CMC), which are available in the repository of the machine learning databases. On the other hand, the high dimensional datasets are: Colon tumour, Breast Cancer and central nervous system (CNS). Since the objective of

this thesis is to improve the effectiveness and efficiency of the clustering, this phase provides different means of evaluation as shown in Figure 3.2.

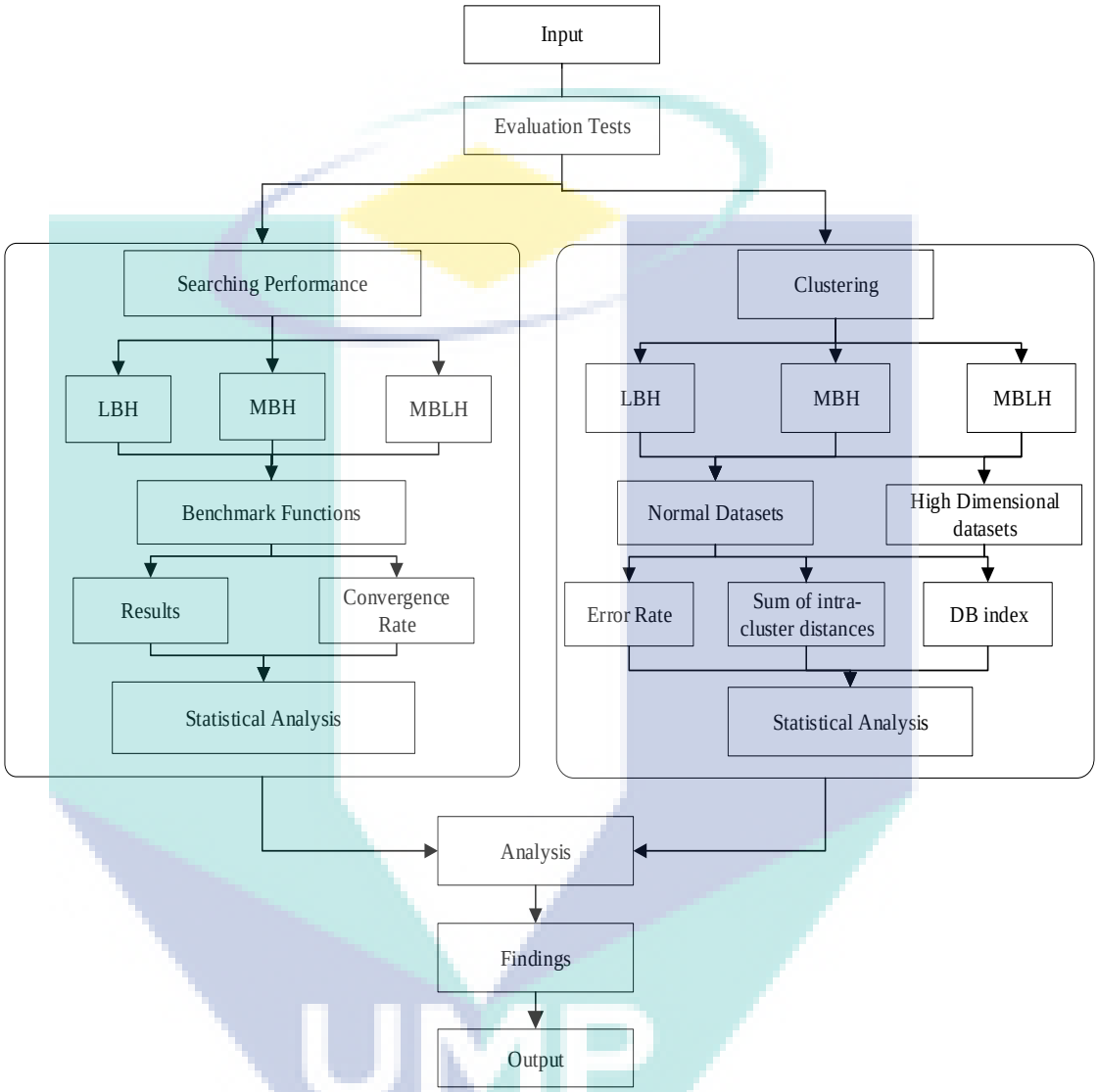


Figure 3.2 Experiment evaluation process

3.3 The Original Black Hole (BH)

3.3.1 Black Hole Phenomena

The black hole concept was established by Dr John Michel and Pierre Pierre Simon de Laplace in the eighteenth century when they depended on Newton's law to invent the concept of a star becoming invisible to the human eye. This concept was not recognized as black hole until in 1967 when John Wheeler, an American physicist, first

referred to this concept of mass collapsing as a black hole (Talbi, 2009). A black hole is formed in nature when a massive star collapses in a process called mass collapsing. The black hole has a strong gravitational force that no form of light can escape from it. This strong gravitational force is because a large matter has been squeezed into a tiny space such that anything that crosses its boundary will be trapped.

The sphere-shaped boundary of a black hole in space is referred to as the event horizon and the radius of this event horizon is referred to as the Schwarzschild radius. The speed of escape at this radius is equal to the speed of light, and once light passes through, it cannot escape. The Schwarzschild radius (R) is calculate as follows: $R = \frac{2GM}{c^2}$, where G = the gravitational constant ($6.67 * 10^{-11} N * (\frac{m}{kg})^2$), M = mass of the black hole, and c = speed of light. If a star moves towards the event horizon or crosses the Schwarzschild radius of the black hole, it will be trapped into the black hole and will disappear permanently. The effect of black hole on objects that surrounds it proves its existence (Giacconi, 2001; Pickover, 1998).

- **Black hole behaviour**

A black hole is a region of space-time (x, y, t) with a strong gravitational field that nothing can escape from it. According to the theory and principle of general relativity, "a sufficiently compact mass will deform space-time to form a black hole". There is a mathematically defined surface around a black hole called an event horizon which marks the point of no return. Because this hole can absorb all lights that hits it, it is referred to as black hole (Schutz, 2003). There are three independent physical properties of a black hole, its mass (M), charge (Q), and angular momentum (J). A charged black hole repels like charges like any other charged object in a given space. Although the simplest black holes have mass, however, they lack angular momentum and electric charge.

3.3.2 Black Hole Algorithm

The black hole concept is simply a region of space with so much mass concentrated in it such that no nearby object can escape its gravitational pull. Anything that falls into a black hole (including light) is permanently gone. Figure 3.3 presents the steps of the Black Hole algorithm.

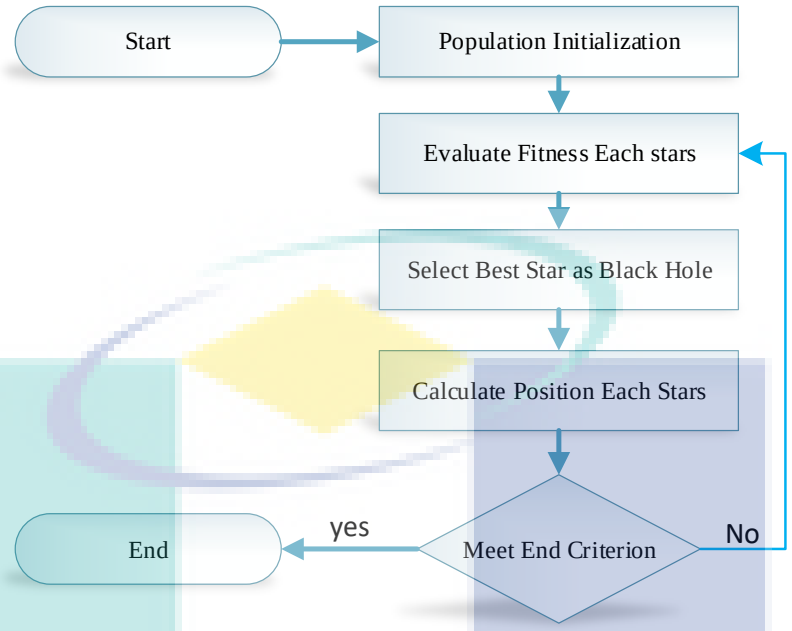


Figure 3.3 The Black Hole Algorithm

BH algorithm consists of three main components. First, Black Hole, which represents the best candidate – or solution – among all the candidates at each iteration. Second, the Stars which denotes the other normal solutions or candidates. The creation of the black hole is not random, and it is one of the real candidates of the population. Finally, Movement component, all the candidates are moved towards the black hole based on their current location and a random number.

The mathematical model of BH can be summarized in three main stages, as follows:

Stage 1: Initialization and Fitness value calculation

1. Initial population: $P(x) = \{x_1^t, x_2^t, x_3^t x_4^t, \ldots, x_n^t\}$ randomly generated individual solutions are placed in the solution space of some problem/function.

2. Determine the total fitness of the population as equation 3.1:

$$f(x_i) = \sum_{i=1}^{pop\_size} eval(p(t) \qquad 3.1$$

Where f_i represents the fitness function for the position of each star x_i, while $eval(x_i)$ represents the defined objective function which either be maximized or minimized. The best star or candidate in terms of the fitness value is then determined and selected as the Black Hole (x_{BH}). After initializing the first black hole and stars, the black hole starts absorbing the stars around it and all the stars start moving towards the black hole.

Stage 2: Absorption rate of the stars by the BH

The black hole starts absorbing the stars around it and all the stars start moving towards the black hole. The absorption of stars by the black hole is formulated as equation 3.2:

$$x_i(t+1) = x_i(t) + rand \times \left(x_{BH} - x_i(t)\right) \; i = 1.2.\cdots N, \qquad 3.2$$

Where $x_i(t+1)$ and $x_i(t)$ = location of the i_{th} star at iteration t and $t+1$ respectively, x_{BH} = location of the BH in the solution space, $rand$ = random number in the range $[0, 1]$ which is generated using a uniform distribution, and N = number of individual solutions in the population. When a star is moving towards the BH, it may reach a location with a lower cost compared to the BH; in such case, the BH will move to the location of that star and vice versa. Then BH algorithm will then proceed with the BH in the new location, pulling the stars to its new location.

Stage 3: Probability of Crossing the Event Horizon during Moving Stars

In the BH algorithm, the probability of a moving star crossing the event horizon of BH is used to gather more optimal data points from the solution space of a problem. Each star (individual solution) that crosses the BH's event horizon will be drawn into the BH and whenever a star (an individual solution) dies, another star will be nominated and randomly distributed in the search space and a new search will be initiated in the search space. The next iteration can only commence when all the stars have been moved. The radius of the BH's event horizon in the algorithm is calculated using the equation 3.3:

$$R = \frac{f_{BH}}{\sum_{i=1}^{N} f_i} \qquad 3.3$$

Where f_{BH} = the BH's ffitness value, f_i = fitness value of the i_{th} star, N = number of stars in the population. A population that has a less number of stars than the allowed minimum number of stars will be omitted. When R is greater than the distance between an individual solution and the BH (the best candidate), that candidate is collapsed, giving room for the creation of a new candidate which will be randomly distributed in the search space.

BH Algorithm

Input: Dataset or Test Function, $MaxItr$, $PopSize$, $Upper$, $Lower$
Output: Best Solution X_{BH}
Procedure:
    Define Objective Function $f(x_i)$
    Initialize all the stars x_i in the population via Uniform Distribution
    Evaluate the fitness value of each star X in the population via f
    Set the best star in the population as Black Hole x_{BH}
    While $itr \leq MaxItr$
        For each star X_i ***in*** the population
            Update the positon of each star X_i via equation 3.2

$$x_i(t+1) = x_i(t) + rand \times \left(x_{BH} - x_i(t)\right)$$

            Check the boundaries of each star X_i
            Evaluate the fitness value of the star X_i in the population via f
        End For
        Calculate the event horizon via equation

$$R = \frac{f_{BH}}{\sum_{i=1}^{N} f_i}$$

        For each star X_i ***in*** the population
            If X_i crosses the event horizon (R) ***Then***
                Remove the star X_i
                Generate a new star via Uniform Distribution
            End If
        End For
        Set the best star in the population as Black Hole X_{BH}
    Loop
    Return X_{BH}

Figure 3.4 Pseudocode of BH algorithm

3.4 Multiple Levy Flight Black Hole (MLBH) Algorithm

As mentioned previously in the methodology of this thesis, a new variant of BH algorithm is design in this thesis, which is called (MLBH). MLBH consists of two main

modifications LBH and MBH. In next subsection, LBH, MBH, and MLBH are explained in details.

3.4.1 Levy Flight Black Hole (LBH) Algorithm

The BH algorithm has been recently developed by (Hatamlou, 2013), it simulates the movement of the stars surrounding the black hole stars. In the standard version of BH, all stars are moved based on the distance between the position of the star and the position of the black hole (best solution), multiplied by a random number generated by a uniform distribution in range [0,1], as shown in equation 3.2 section 3.3.2. This equation represents the exploration or the global search ability of BH. The $rand$ parameter in equation 3.2 section 3.3.2 may leads to almost same values, which cause to move the stars into a close position to the current. Therefore, the formula for moving the stars to explore the solution space causes them to over-scatter and leads to slow convergence and may leads to a trapping in the local optima.

A new modification in this thesis is proposed to overcome the issue of the movement equation. The long jumps have been undertaken via Levy distribution in order to ensure effectual use of the search space in comparison with BH. Previously investigated works have aimed to improve BH, whereby the current proposal calls for BH to perform random walks and global search. Levy flight, in particular, improves the global search capacity for the BH algorithm, preventing one to be stuck in local minima. Additionally, the proposed modification enhances the global search ability of BH algorithm as per the new equation of star movements underlined. As BH algorithm is incapable of attaining the optimum results in a specific number of iterations, an efficient Levy-flight selection is imperative to avoid being stuck in local optimum as it results in improved global and local search capability concomitantly.

Hence, the suggested algorithm is designed in a manner that it allows the BH algorithm's local search ability, which will improve the method's efficiency in generating the optimal resolution and accelerating the convergence rate. Some examples of Levy flight compared with the Brownian walk (random) have been displayed in Figure 3.5(Haklı & Uğuz, 2014) . After the first movements around a point, sudden jumps are encountered; it generates the simultaneous local and global search. The suggested method describes the creation of an arbitrary population of stars, which then requires their respective cost to be calculated. The black hole is next marked in the

ensuing stage, which is followed by movement operations and stars disappearing in the black hole.

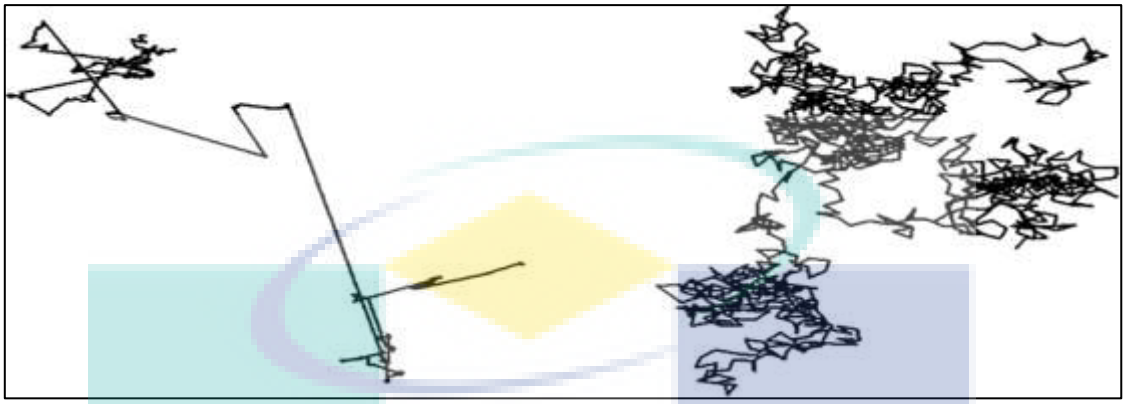


Figure 3.5 Motion path in Levy flight and Brownian (random) walk (Haklı & Uğuz, 2014)

The pseudocode and flowchart of LBH are presented in Figure 3.6 and Figure 3.7 respectively.

LBH Algorithm

Input: Dataset or Test Function, $MaxItr$, $PopSize$, $Upper$, $Lower$
Output: Best Solution X_{BH}
Procedure:
    Define Objective Function $f(x_i)$
    Initialize all the stars x_i in the population via the uniform distribution
    Evaluate the fitness value of each star X in the population via f
    Set the best star in the population as Black Hole x_{BH}
    While $itr \leq MaxItr$
        For each star X_i ***in*** the population
            Update the positon of each star X_i via the following equations :

$$step(t) = 0.01 \times s(t) \times rand(0,1)$$
$$x_t(t+1) = x_t(t) + (step(t) \times (x_{BH} - x_t(t)))$$

            Check the boundaries of each star X_i
            Evaluate the fitness value of the star X_i in the population via f
            Set the best star in the population as Black Hole x_{BH}
        End For
        Calculate the event horizon via the following equation:

$$R = \frac{f_{BH}}{\sum_{i=1}^{N} f_i}$$

        For each star X_i ***in*** the population
            If X_i crosses the event horizon (R) ***Then***
                Remove the star X_i
                Generate a new star via uniform distribution
            End If
        End For
        Set the best star in the population as Black Hole X_{BH}
    Loop
    Return X_{BH}

Figure 3.6 Pseudocode of LBH

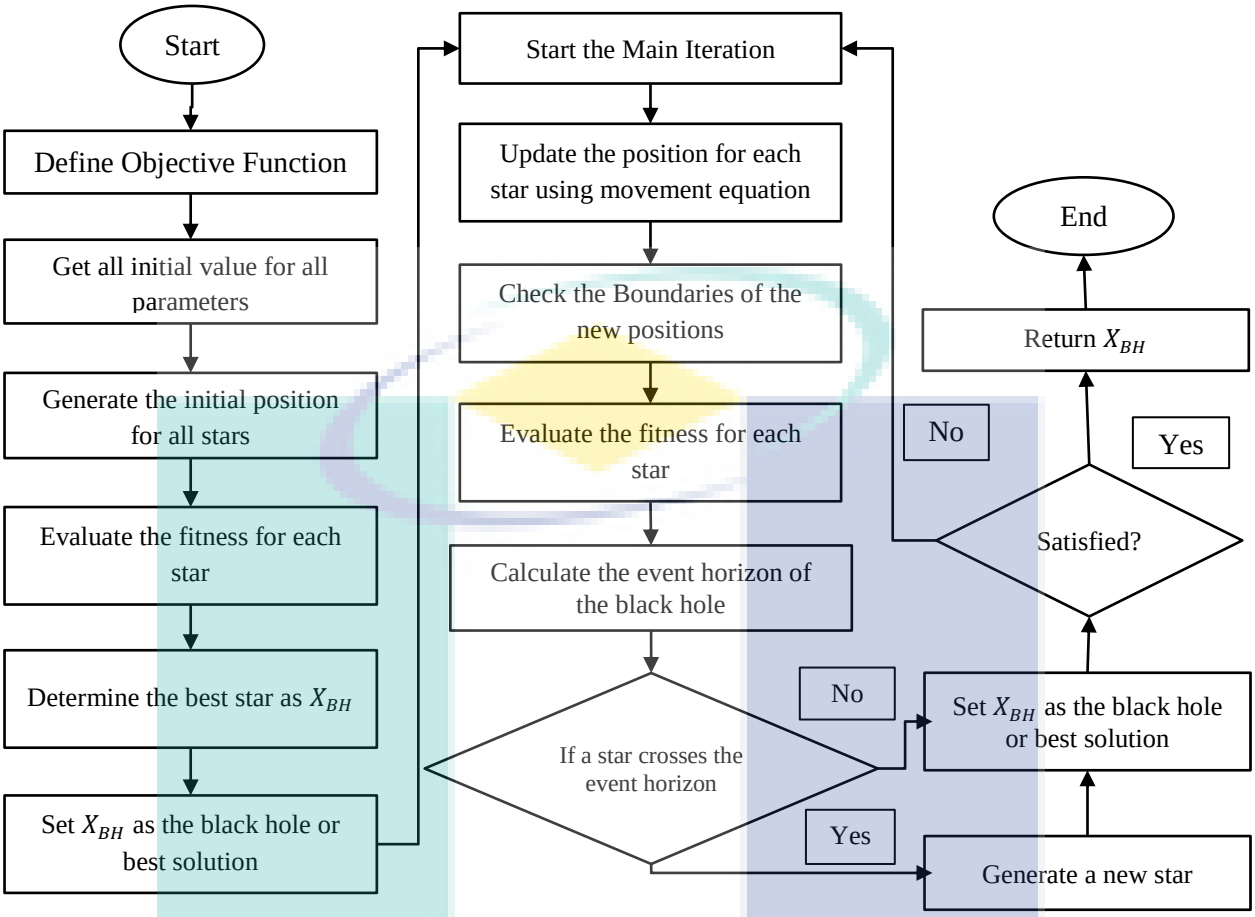


Figure 3.7 Flowchart of LBH algorithm

3.4.2 Multiple Black Hole (MBH) Algorithm

The standard version of BH algorithm does not perform the global search well, it is only performed when there is a star with fitness lower than the event horizon (R). Which means only small number of stars are re-generated for the exploration purposes, while in some problems there is a need for exploring the search space more than what BH does. Therefore, in case of the exploitation capabilities, being performed more than the exploration capabilities; the chances of trapping in the local optimum are increased. An enhanced version of BH algorithm was proposed and called as the "Multiple Black Hole (MBH) Algorithm" for the problem of data clustering. MBH is based on the original BH algorithm but uses multiple populations instead of a single one. Each population is composed of a number of candidate solutions (stars) that undergoes random generation in the search space. Then, the populations are initialized and each of their fitness values is assessed, whereby the best candidate having the best fitness value

is chosen as the black hole, while the rest reverts to become normal stars. As the black hole is capable of absorbing stars around it, such process of star absorption occurs after the black hole and stars are initialized, at which the stars move. The absorption process has been formulated as seen below:

$$x_i(t+1) = x_i(t) \times rand \times \left(x_{BH} - x_i(t)\right) \; i = 1.2.\cdots N, \qquad 3.4$$

Where $x_i(t+1)$ and $x_i(t)$ are the location of the i_{th} star at iteration t and $t+1$, x_{BH} is the location of the black hole in the search space, c is a constant, $rand$ is a random number in the interval $[0, 1]$, and N is the number of stars (candidate solutions) in the population.

A population must be omitted if the number of its stars becomes less than the minimum allowed number of stars in a population. At each iteration, there will be a probability of generating a new population ($prob_{generating\_population}$), which will help to explore the entire search space and avoid the local minima at a minimum number of iterations (speedup the convergence to global optima in early iterations).

$$prob_{generating\_population} = \frac{rand}{number\ of\ populations} \qquad 3.5$$

Where $rand$ is a random number in the interval[0.1]. The solutions of the new population are generated in two ways:

1) Arbitrarily in the search space, and

2) Arbitrarily chosen from other populations.

The ratio r_g is used to mix between the two ways and is formulated as equation 3.6:

$$r_g = \frac{itr}{max\ iterations}, \qquad 3.6$$

Where itr is the iteration of generating the new population and max iterations refers to the total number of iterations. Therefore, the searching process during the early iterations is considered to be a global search (r_g) is small and the solutions are arbitrarily generated in the search space. As the iterations go on, it becomes a local search (r_g) is getting bigger and the solutions are taken from other populations. Note

that the value of r_g can be also selected as constant. Thus, in order to generate a new population, there are two cases: if r_g is less than 50% of the total no. of iterations then generate a new random population, otherwise, generate the population based on the position of the global best black hole (BH_G) as shown in the equations 3.7 and 3.8:

$$Population\ (P)\begin{cases} r_g \leq 0.5\ \ then\ generate\ random\ population \\ otherwise,\ \ generate\ based\ on\ BH_G \end{cases} \quad 3.7$$

$$Population(P).X_i = X_i + (BH_G - X_R) * rand \quad 3.8$$

Where X_i represensts a new star. In the population P, while X_R represents a randomly selected star from another randomly selected population, $rand$ is a random number in range $[0, 1]$. The key processes for the enhanced BH algorithm are subsequently summarized using the Figure 3.8 pseudocode, while the flowchart is given in Figure 3.9.

MBH Algorithm

1. **Inputs:** Dataset or Test Function, $Num\_of\_Pops, PopSize, Max\_Sols, Min\_Sols, MaxItr, SC_{max}, Upper, Lower$
2. **Output:** Best Solution BH_G
3. **Procedure**
4. Define Objective Function $f(x_i)$
5. Initialize all populations of stars with random locations in the search space, as follows:
   $$pop(p).x_i = (Upper - Lower) \times rand + Lower$$
6. Evaluate the objective function for each star in each population using the define objective function
7. Determine the best star in each populations $pop(p).x_i$
   *EvaluationNum=0; MaxEvaluation = P * Num_of_Pops * MaxItr;*
8. ***While*** $itr \leq MaxItr$
9. ***For*** $p = 1$ *to* Num_of_Pops
10. ***For*** each star X_i ***in*** Pop (p) *MaxEvaluation*
11. Move the star $pop(p).x_i$ via the following equation:
    $$pop(p).x_i(t+1) = pop(p).x_i(t) \times rand \times \left(pop(p).x_{BH} - pop(p).x_i(t)\right)$$
12. Check the boundaries of $pop(p).x_i$
13. Evaluate the objective function for $pop(p).x_i$
14. EvaluationNum + 1;
15. If EvaluationNum ≥ MaxEvaluation
    Go to line 40
16. ***End For***
17. Determine the best star in Pop (p) as ($pop(p).x_{BH}$) *as follows:*
    $$pop(p).x_{BH} = \text{Min } pop(p)$$
18. *Calculate the value of the event horizon for* $pop(p)$ *via the following equation:*
    $$pop(p).R = \frac{f(pop(p).x_{BH})}{\sum_{i=1}^{N} f(pop(p).x_i)}$$
19. ***For*** each star X_i in Pop (p)
20. ***If*** $f(pop\ (p).X_i\ ) < (pop(p).R)$ ***Then***
21. Re-generate a new $pop\ (p).X_i$ via the uniform distribution
22. EvaluationNum + 1;
23. If EvaluationNum ≥ MaxEvaluation
    Go to line 40
24. ***End If***
25. ***End For***
26. ***End For***
27. Set the global best solution overall population as the global best black hole (BH_G)
28. ***For*** $p = 1$ *to* Num_of_Pops
29. Calculate the probability of generating or replacing the population, as follows:
    $$prob_{generating_{population}} = \frac{rand}{no.of\ population}$$
30. ***If*** $rand < prob_{generating\_population}$ ***Then***
31. Calculate the generating ratio r_g as follows:
    $$r_g = \frac{itr}{max\ iterations}$$
32. ***If*** $r_g \leq 0.5$ ***Then***
33. Generate new $pop(p)$ of stars with random locations in the search space via the uniform distribution
    $$pop(p).x_i = (Upper - Lower) \times rand + Lower$$
34. ***Else***
35. Generate new $pop(p)$ based on BH_g as follows:
    $$pop(p).x_i = pop(p).x_i + (BH_G - pop(r_1).x_{r2}) * rand$$
36. ***End If***
37. ***End If***
38. ***End For***
39. ***Loop***
40. Determine the best solution in all populations as (BH_G)
41. Return BH_G

Figure 3.8 Psuedocode of MBH

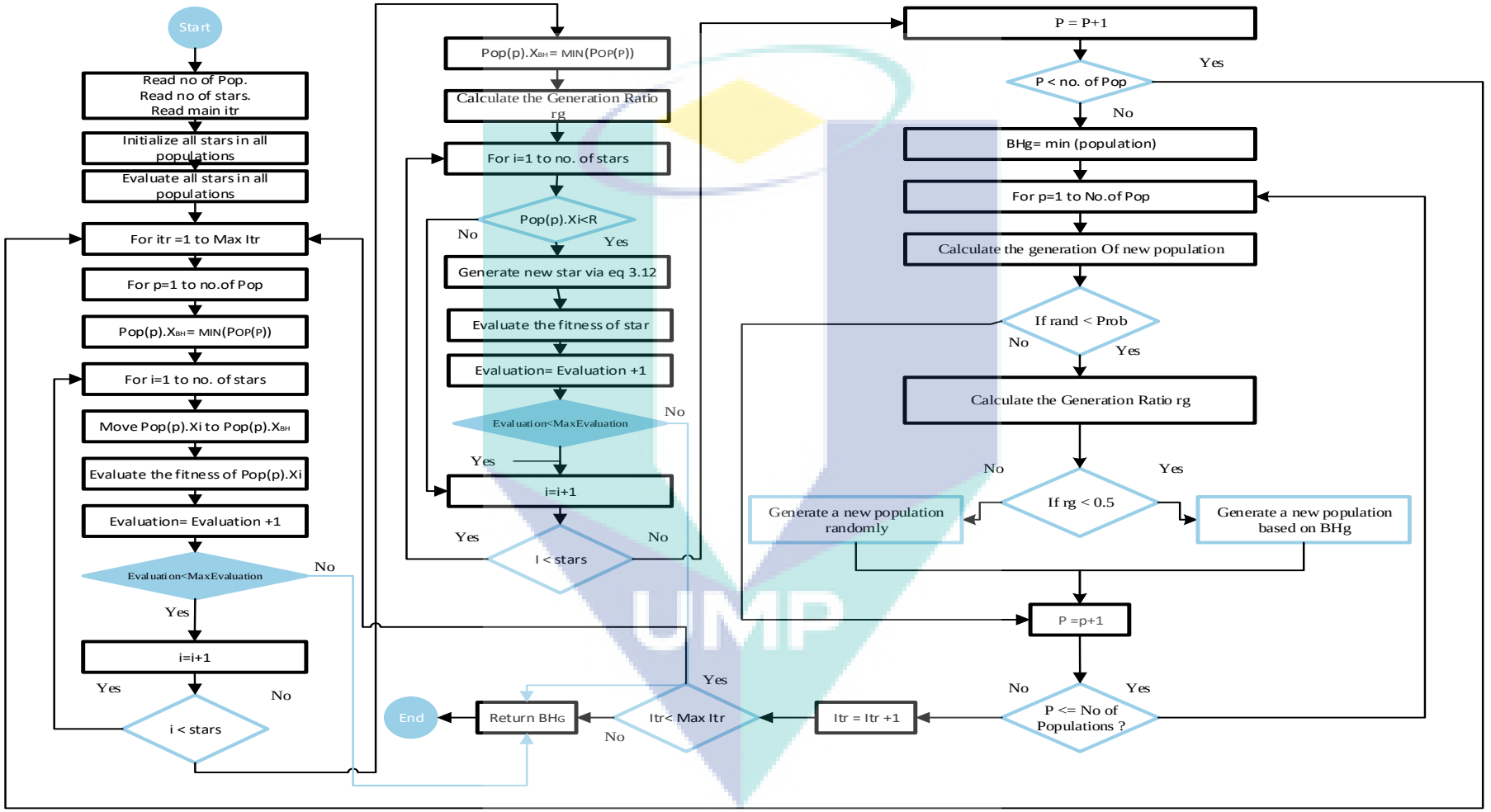


Figure 3.9 Flowchart of MBH algorithm.

3.4.3 Multiple Levy Flight Black Hole

Multiple Levy Flight Black Hole (MLBH) algorithm which is the main contribution of this thesis is presented and explained in this subsection. MLBH is a new variant of Black Hole algorithm, the main difference between the original BH and MLBH is that MLBH contains a new mechanism for exploring the search space more than the original BH algorithm. Therefore, the chances of falling in the local optima are less when using MLBH for the global optimization problems in general, and data clustering problems in particular due to its ability of visiting positions which are not explored by the standard version of BH.

In the previous two subsections, two enhancements or modifications for the original BH algorithm, they (LBH and MBH) were explained in details. Although these two modifications had enhanced the global search ability of the original BH, these two modifications have their drawbacks. These drawbacks are:

i. LBH: The Levy Flight formula generates the step sizes in the movement equation in LBH. The values of the step sizes depend mainly on the value of the β, as it gets larger, the stars jump for new positions far from the current local best or the previous positions, and vice versa. Although the large value of β enhances the exploration rate of BH algorithm, but it may keep generating large values even when the stars or the solutions converged towards the optimal solutions. Therefore, there is a chance when the exploration rate of LBH is more than exploitation which leads to premature convergence.

ii. MBH: The main structure of MBH consists of two main parts, the generating/mixing the populations, and the movement of the stars towards the black hole or the current best solution inside each population. Therefore, for each population, there is a chance to fall in the local optima due to the low exploration rate of the stars in the populations, which leads to search in area near to the current local best solution and the rest parts of the search space remain undiscovered. However, MBH enhances the exploration of the original BH by using a multi population architecture.

In order to overcome the above mentioned issues, MLBH combines both LBH and MBH. In other words, the multi-population structure of MBH decreases the chances

of premature convergence in LBH by generating several populations instead of one, meaning that the populations help each other when any one failed to perform a good local search. Moreover, the Levy Flight distribution in LBH enhances the movements of stars in MBH, which enhances the global search ability in MBH and decreases the chances of falling in local optima. Therefore, MLBH is more stable than the original BH, LBH and MBH, because the above-mentioned reasons. Figure 3.10 below depicts the different between BH, LBH, MBH, and MLBH.

It is worth to mention that the mathematical model of MLBH is the same of MBH with Levy Flight distribution equation in the movements of stars. Therefore, there is no specific mathematical model for MLBH. **Error! Reference source not found.**, Figure 3.11, and Figure 3.11 present the pseudocode, the block diagram of the flowchart of MLBH.

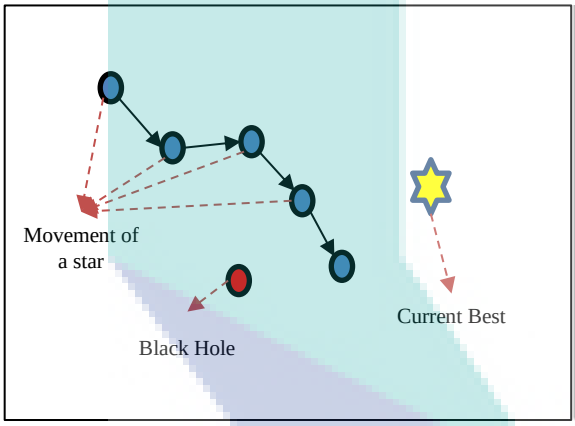


Figure 3.10a. The movement of a star in BH

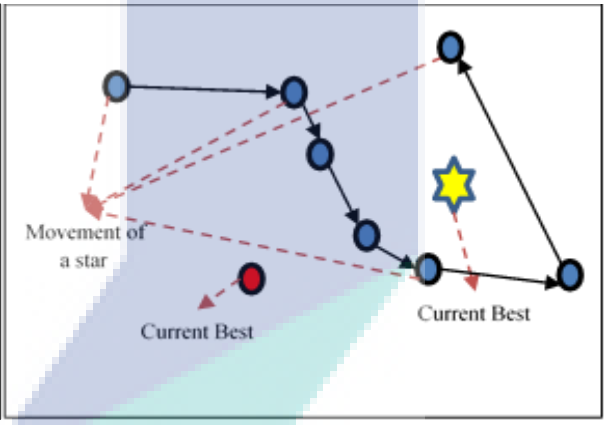


Figure 3.10b. The movement of a star in LBH

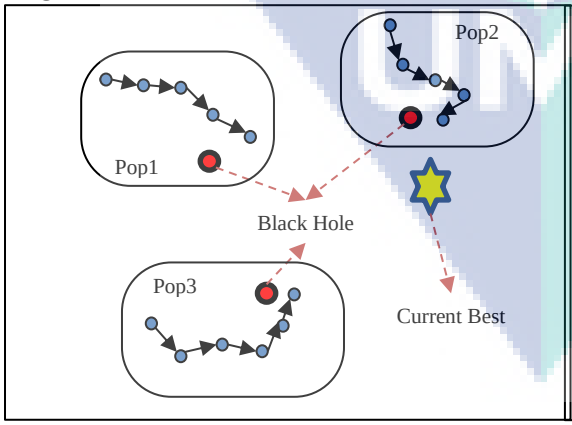


Figure 3.10c. The movement of a star in MBH

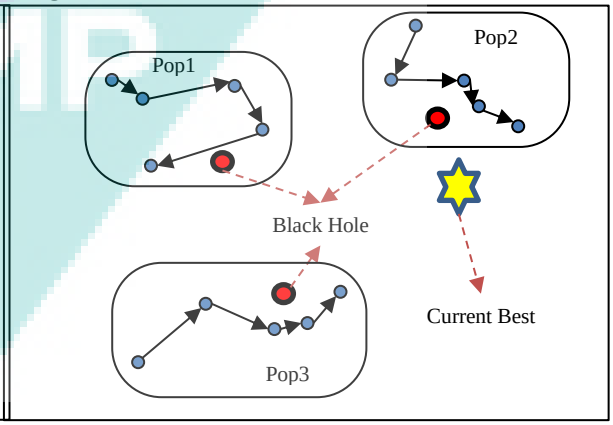


Figure 3.10d. The movement of a star in MLBH

Figure 3.10 Graphical illustration of a star movment in BH, LBH, MBH, and MLBH

It can be seen from the graphical illustration above, that movements of an individual star or solution is different between each algorithm. The drawback of LBH is clear in Figure 3.10b, when star moved to a far position from the optimal solution and the current black hole because the Levy Flight generated a large. Moreover, the drawback of MBH is clear in Figure 3.10c, when the star moved to a position close to the current black hole. In other words, the star tried to find a better position near to the current best solution instead of moving towards the optimal solution (i.e., less exploration). In MLBH, the stars in each population moved towards different position based on different generated step sizes.

MLBH Algorithm

Inputs: Dataset or Test Function, $Num\_of\_Pops, PopSize, Max\_Sols, Min\_Sols, MaxItr, SC_{max}, Upper, Lower$
Output: Best Solution BH_G
Procedure
  Define Objective Function $f(x_i)$
  Initialize all populations of stars with random locations in the search space, as follows:
    $pop(p).x_i = (Upper - Lower) \times rand + Lower$
  Evaluate the objective function for each star in each population using the define objective function
  Determine the best star in each populations $pop(p).x_i$
  *EvaluationNum=0; MaxEvaluation = Number_of_stars * MaxItr;*
  While $itr \leq MaxItr$
    For $p = 1$ *to* Num_of_Pops
      For each star X_i ***in*** Pop (p) *MaxEvaluation*
        Move the star $pop(p).x_i$ via the following equation:
          $step(t) = 0.01 \times s(t) \times rand(0,1)$
          $x_t(t+1) = x_t(t) + (step(t) \times (x_{BH} - x_t(t)))$
        Check the boundaries of $pop(p).x_i$
        Evaluate the objective function for $pop(p).x_i$
        EvaluationNum + 1;
        If EvaluationNum ≥ MaxEvaluation
          Go to line 40
      End For
      Determine the best star in Pop (p) as ($pop(p).x_{BH}$) *as follows:*
        $pop(p).x_{BH}$=Min $pop(p)$
      Calculate the value of the event horizon for $pop(p)$ *via the following equation:*
        $$pop(p).R = \frac{f(pop(p).x_{BH})}{\sum_{i=1}^{N} f(pop(p).x_i)}$$
      For each star X_i in Pop (p)
        If $f(pop\ (p).X_i\ ) < (pop(p).R)$ ***Then***
          Re-generate a new $pop\ (p).X_i$ via the uniform distribution
          EvaluationNum + 1;
          If EvaluationNum ≥ MaxEvaluation
            Go to line 40
        End If
      End For
    End For
    Set the global best solution overall population as the global best black hole (BH_G)
    For $p = 1$ *to* Num_of_Pops
      Calculate the probability of generating or replacing the population, as follows:
        $$prob_{generating_{population}} = \frac{rand}{no.of\ population}$$
      If $rand < prob_{generating\_population}$ ***Then***
        Calculate the generating ratio r_g as follows:
          $$r_g = \frac{itr}{max\ iterations}$$
        If $r_g \leq 0.5$ ***Then***
          Generate new $pop(p)$ of stars with random locations in the search space via the uniform distribution
            $pop(p).x_i = (Upper - Lower) \times rand + Lower$
        Else
          Generate new $pop(p)$ based on BH_g as follows:
            $pop(p).x_i = pop(p).x_i + (BH_G - pop(r_1).x_{r2}) * rand$
        End If
      End If
    End For
  Loop
  Determine the best solution in all populations as (BH_G)
  Return BH_G

Figure 3.11 Psuedocode of MLBH

Based on the aforementioned description of BH, Figure 3.11 depicts the complete algorithm as the backbone of the BH for data clustering. Unlike the standard BH, the Levy Flight formula, which is utilized to generates the step sizes in the movement equation in LBH. The values of the step sizes depend mainly on the value of the β, as it gets larger, the stars jump for new positions far from the current local best or the previous positions, and vice versa.

Line 5 represents the initialization process, which is done by multi population instead of initial population (Equation 3.13), which can carry out the overall search at higher speed. Specifically, line 6 defines the initial evaluation and the selection of the initial best BH candidate. The evaluation steps calculate the fitness of all the stars in the swarm. The BH are evaluated using the objective function specified earlier in Equation 2.10. The best BH with highest fitness value, is selected and considered as the best BH. As the best BH is selected, the iteration is initiated in line 8 where the LBH cycle until the condition is reached. This consider as the stopping condition. During this iteration from line 22 to line 29 the best BH here is improved in each generation.

The process of improvement is progressing in line 30 when best BH has better intensity than the previous best BH, then best BH will move toward the current best BH using Equation 3.4, the movement is executed using the discrete Levy in Equation 3.9, and 3.10 respectively. Then, in line 32, In case of best BH is not improved in the current generation, using Equation 3.5 a new population will generate using Equation 3.4. The solutions of the new population are generated in two ways: 1) arbitrarily in the search space, and 2) arbitrarily chosen from other populations. The ratio r_g is used to mix between the two ways in presented in line 34. Finally, line 41 represents the selection the best BH with the best fitness value. Based on the aforementioned description of the MLBH, Figure 3.13 depicts the complete algorithm as the backbone for BH clustering algorithm. Unlike the BH, the MLBH introduce the new movement mechanism aims to solve the exploration convergence issues in the BH.

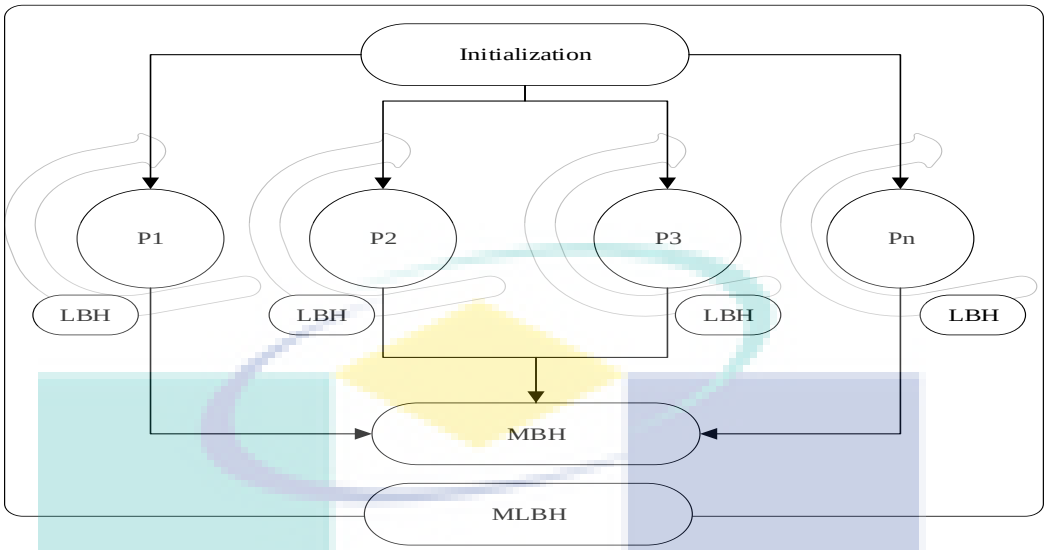


Figure 3.11 The block diagram of MLBH

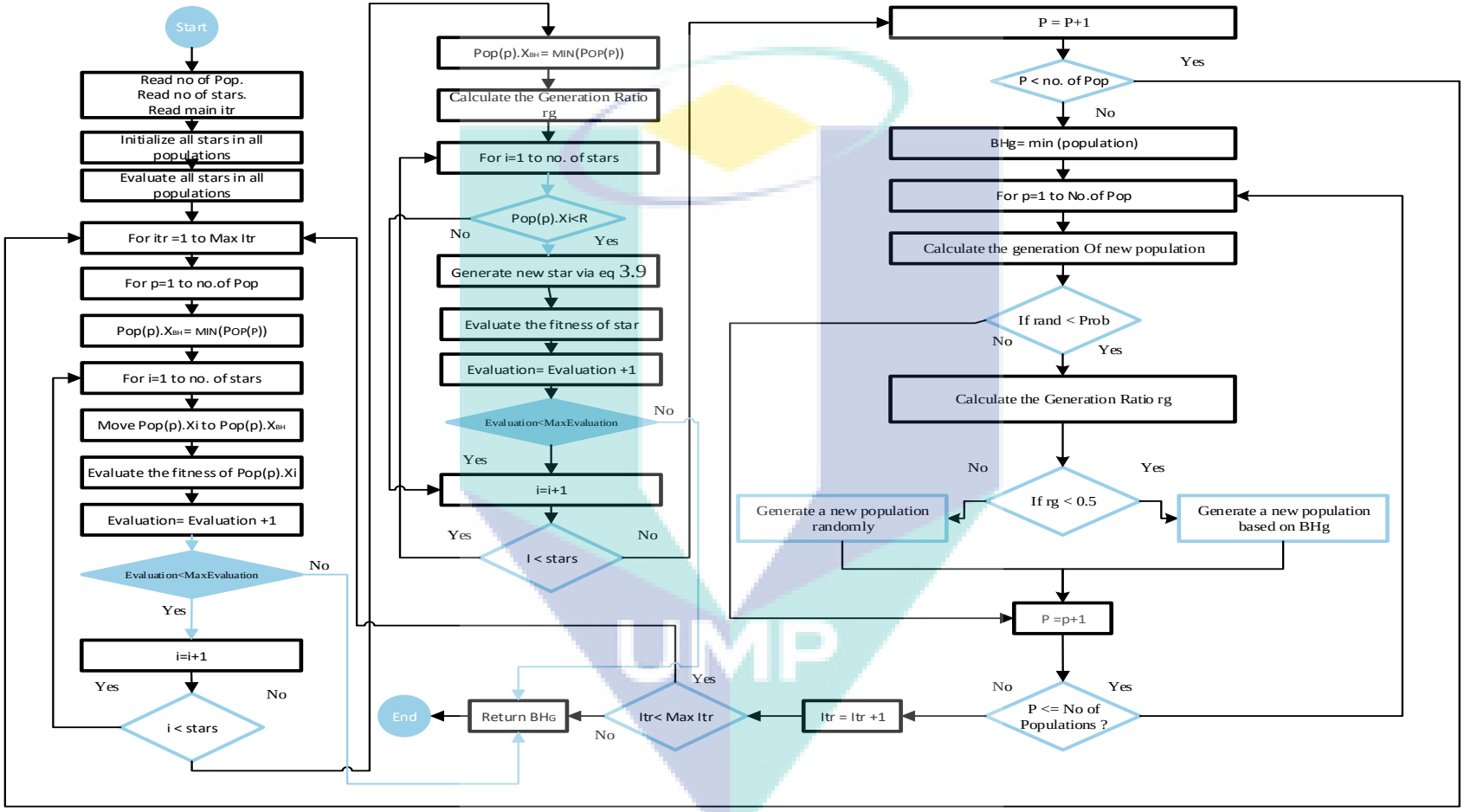


Figure 3.12 Flowchart of MLBH

3.5 Clustering of High Dimensional Datasets

In the previous section, the MLBH algorithm has been explained in details. The goal of MLBH is to solve the optimization problems by searching for the best decision variables which maximize/minimize the objective function. MLBH also can be implemented for solving the problem of data clustering by searching for the high quality cluster in normal datasets. The main different between the normal and high dimensional datasets is the size or the number of features in the datasets. High dimensional datasets mean the datasets with hundreds or thousands features, which need for an efficient and powerful clustering algorithm. In this section, MLBH algorithm is integrated with Mutual information to determine the most relevant features in the datasets is used to find the best centroid clusters. The general framework structure of MLBH for high dimensional datasets is presented in Figure 3.13.

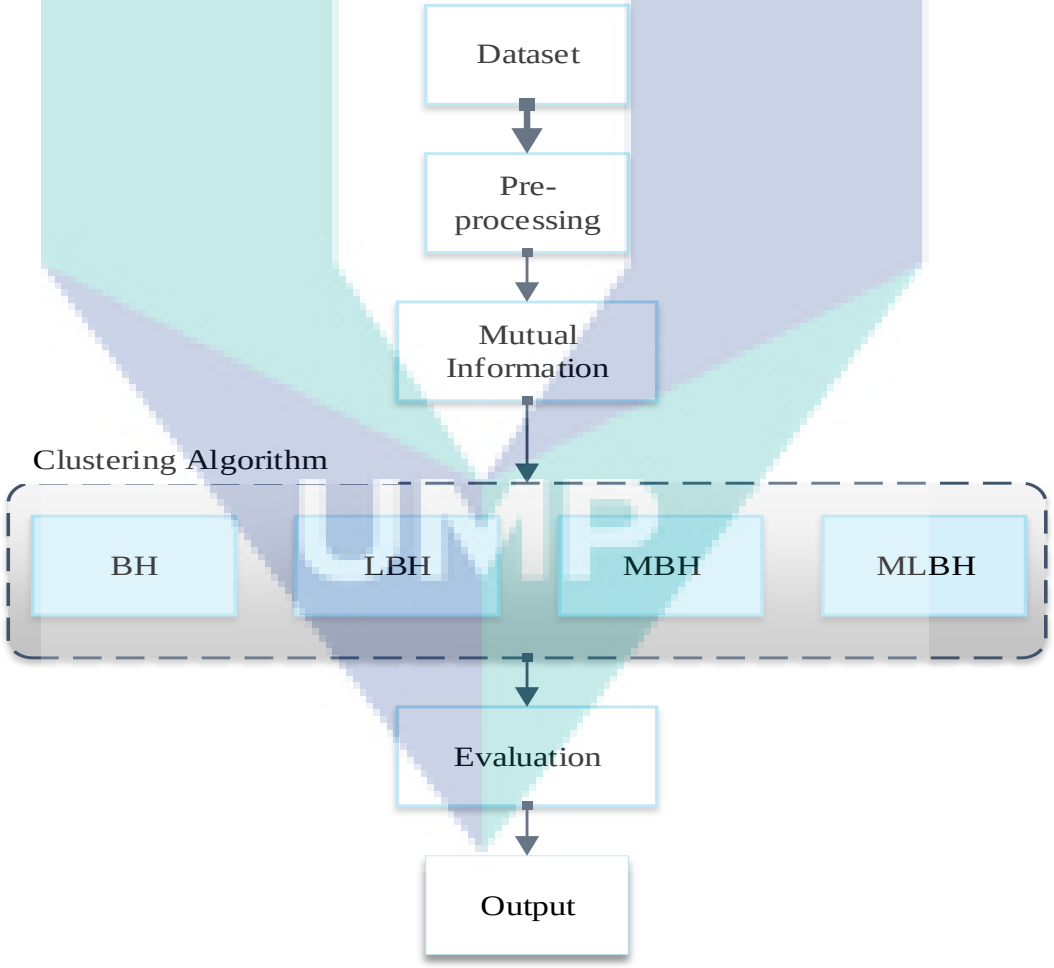


Figure 3.13 The framework of MLBH for High dimensional datasets.

As shown in Figure 3.13, the MLBH for High dimensional consists of five main components; they are: Datasets, Data Pre-processing, Mutual Information, Clustering Algorithm, and Evaluation Metrics.

3.5.1 Datasets

Managing and analysis of medical big data involve many different issues regarding their structure, storage and analysis. For reducing the dimensions, selecting features and classification in such datasets.

3.5.2 Pre-processing

In this step, the whole dataset is pre-processed and cleaned from the noise. Pre-processing consists of two steps, scaling and normalization. In the scaling step, the dataset is converted from string representation into numerical representation. In other words, any feature or class label represented by a string will be converted into a numerical value. The second step is normalization. The normalization cleans the noises from the dataset and decreases the differences in the ranges between the features. In this work, Min-Max normalization method is implemented, as equation 3.9:

$$F_i = \frac{(F_i - Min_i)}{(Max_i - Min_i)} \qquad 3.9$$

Where F_i represents the current feature needs to be normalized, Min_i and Max_i represent the minimum and the maximum value for that feature respectively.

3.5.3 Mutual Information (MI)

Mutual information (MI) is a well-known filter-based feature selection method. In the filter-based methods, variables are ranked without any form of dependence on the classifier, and some of such performance measures are Fisher score, Pearson correlation coefficient, and Information theory-based measures. Such techniques are advantageous because they are easy to implement, computationally less expensive, and provides a more generalizable feature subset since they are not dependent on any classifier or cluster algorithm. Having said that, their major problem is that they cannot exploit specific machine learning algorithm characteristics which are intended for use, and as such, rarely achieve the highest level of classification accuracies.

MI refers to the specific information shared by 2 variables. Through entropy, the conveyable information from a variable can be quantified, but the major point of interest is the level of variables overlaps in the recorded variables. This is important when considering the effectiveness of one variable in the prediction of the other; a higher level of shared information implies that a similar information source is being measured:

$$I(X:Y) = H(X) - H(X|Y) = H(X) + H(Y) - H(X,Y) \quad 3.10$$

Where I represents the value of weight of individual feature, while H denotes the entropy value. Entropy is calculated by the summation of all the probability distribution of values of the feature, multiplied by the natural log of those probability distribution, as equation 3.11:

$$H(X) = -\sum p(x)\log p(x) \quad 3.11$$

Where x represent a value of the set X, while $p(x)$ represents the probability distribution of x. The resulting weight values of the MI are tested so if it less than th it will be deleted. The weights of features generated by MI are real values, therefore, the weights are more than a threshold value are selected, the rest are removed or unselected. The threshold value in this study is zero. Meaning that, any feature with zero weight is removed from the dataset.

3.5.4 Clustering Algorithm

In this step, the clustering algorithm is applied to find the best intra cluster distance in the dataset. The clustering algorithm depends on the resulted subset of features from the previous step. There are four clustering algorithms in this study, the original BH, LBH, MBH, and MLBH. All of these algorithms are implemented and examined based on the same subset of features in order to evaluate them.

3.5.5 Evaluation Step

The results of the clustering algorithms are evaluated based on three main measurements; they are: sum of intra destines error rate, and davies-bouldin (DB) index. These measurements are explained in details in the next section.

Our evaluation focuses on two related goals. Firstly, we compare the performance of our algorithm against existing algorithms from the literature. Then, we verify our findings using statistical analysis. In our evaluation, we note that the comparative performances with the same number of objective function evaluation are not possible for metaheuristic-based algorithms (i.e. most implementations are not publically available; hence, the settings of each of the algorithm parameters are beyond our controls).

3.6 Test Functions, Datasets, and Evaluation Metrics

3.6.1 Evaluation on Benchmark Test Functions

In order to further verify that LBH and MBH has a better exploration than the standard BH, it has been evaluated on a set of multi-model type of objective functions in a multi-dimensional space as defined in Refs. (Jaddi et al., 2017; Niu et al., 2007; Zhang et al., 2017). The functions with their main characteristics in terms of Name, Dimensions (D), Upper and Lower Boundaries (UB, LB) and the value of optimal solution (Opt) are stated in Table 3.1. The benchmark test functions used in this study are classified into two main classes, Unimodal and Multimodal. The function is called 'Unimodal' when it has only single global optima needs to be found, while 'Multimodal' represents the functions with more than one global optima. The algorithm has a good exploration ability when it performs better on the multimodal function than the other algorithms, while the exploitation is evaluated based on the unimodal test functions.

Table 3.1 Benchmark Test Functions

Func.	Name	Test	D	LB	UB	Opt
f_1	Sumsquare	$f(x) = \sum_{i=1}^{D} ix_i^2$	30	-100	100	0
f_2	Rastrigin	$f_2(x) = \sum_{i=1}^{N}\{x_i^2 - 10\cos(2\pi x_i) + 10\}$	30	-5.12	5.12	0
f_3	Quartic	$f_3(x) = \sum_{i=1}^{n} ix_i^4 + random(0,1)$	30	-1.28	1.28	0
f_4	Ackley	$f_4(x) = -20e^{-0.02\sqrt{D^{-1}\sum_{i=1}^{D} x_1^2}} - e^{D^{-1}\sum_{i=1}^{D}\cos(2\pi x_i)} + 20 + e$	30	-32	32	0
f_5	Alpine No.1	$f_5(x) = \sum_{i=1}^{D}\lvert x_i \sin(x_i) + 0.1x_i\rvert$	30	-10	10	0
f_6	Griewank	$f_6(x) = \sum_{i=1}^{Dim}\frac{y_i^2}{4000} - \prod_{i=1}^{Dim}\cos\left(\frac{y_i}{\sqrt{i}}\right) + 1$	30	-600	600	0
f_7	Penalized	$f_7(x) = \sum_{i=1}^{Dim-1}(y_i - 1)^2 \times (1 + sin^2)(3\pi y_{i+1}) + (y_{Dim} - 1)^2(1 + sin^2(2\pi y_{Dim})) + sin^2(3\pi y_1)$	30	-50	50	0
f_8	Zakharov	$f_8(x) = \sum_{i=1}^{n} x_i^2 + (\frac{1}{2}\sum_{i=1}^{n} ix_i)^2 + (\frac{1}{2}\sum_{i=1}^{n} ix_i)^4$	30	-5	10	0
f_9	Sphere	$f_9(x) = \sum_{i=1}^{N} x_1^2$	30	-100	100	0

3.6.2 Clustering Datasets

To evaluate the performance of proposed algorithms (LBH, MBH) for data clustering, six datasets have been used. The datasets, namely, Iris, Wine, Glass, Cancer, Contraceptive Method Choice (CMC) and Vowel. All data sets are available from UCI machine learning laboratory. The datasets utilized in this particular study are displayed in Table 3.2.

I. Iris Dataset

The dataset consisted of 150 arbitrary samples of flowers having four features from the iris. They were differentiated into 3 groups of 50 instances, whereby each group represented a form of iris plant (Setosa, Versicolor and Virginica).

II. Wine Dataset

The dataset elucidated the quality of wine using the physicochemical properties, in which they were grown in the identical region in Italy but sourced from three cultivars, respectively. Each of the three types of wine was linked to 178 instances, with 13 numeric attributes representing the quantities of 13 components elicited in them.

III. CMC Dataset

The dataset was generated by TjenSien Lim, which is a subset of Indonesia's 1987 National Contraceptive Prevalence Survey. The sample size consisted of married women who were either not pregnant or not in the know of their pregnancy during the interview period. It featured the issue of predicting the recent contraceptive method choice (i.e. no use, long-term method, or short-term methods) according to a woman's demographic and socioeconomic attributes.

IV. Cancer Dataset

The dataset was a representation of the Wisconsin breast cancer database, consisting of 683 instances having 9 components. They included: Clump Thickness, Cell Size Uniformity, Cell Shape Uniformity, Marginal Adhesion, Single Epithelial Cell Size, Bare Nuclei, Bland Chromatin, Normal Nuclei, and Mitoses. Each of the instances was possibly of one class, either benign or malignant.

V. Glass Dataset

The dataset consisted of 214 objects with nine features, which were: refractive index, sodium, magnesium, aluminium, silicon, potassium, calcium, barium, and iron. The data sampling was done using six groups of glass, which were: float processed building windows, non-float processed building windows, float-processed vehicle windows, containers, tableware, and headlamps.

VI. Vowel Dataset

The dataset was comprised of 871 Indian Telugu vowel sounds, inclusive of three attributes that corresponded to the first, second and third vowel frequencies, as well as six overlapping classes.

Table 3.2 The main characteristics of the used datasets

Datasets	No. of classes	No. of features	No of instances	Size of classes
Iris	3	4	150	50,50,50
Wine	3	13	178	59,71,48
CMC	3	9	1473	629,334,510
Cancer	2	9	683	444,178
Glass	6	9	214	70,17,76,13,9,29
Vowel	6	3	871	72,89,172,151,207,180

3.6.3 Evaluation Measures for Normal Datasets

- **Sum of Intra Cluster Distances:** The process of evaluating the results of a clustering algorithm is called cluster validity assessment. A good clustering method will produce high quality clusters with high intra cluster similarity and low inter cluster similarity. Sum of Intra Cluster Distances (SICD) is the most known evaluating criteria for clustering data. Less value of SICD means higher quality clustering is performed (Neshat et al., 2012).

$$F(O.Z) = \sum_{i=1}^{N} \sum_{j=1}^{K} \left\| O_i - Z_j \right\|^2 \quad 3.12$$

Euclidean distance between each gene vector in a cluster and the centroid of that cluster is calculated and summed up. Here, in K clusters C_i $(1 \leq i \leq K)$, each of N gene vector x_j are clustered on the basis of distance from each other of these cluster centers x_i $(1 \leq i \leq K)$.

- **Error Rate (ER)** as an external quality measure: The percentage of misplaced data objects, which is formulated as:

$$ER = \frac{Number\ of\ misplaced\ objects}{total\ number\ of\ objects\ within\ dataset} 100 \quad 3.13$$

3.7 High Dimensional Datasets

MLBH, LBH and MBH have been evaluated based on three publicly available microarray datasets, which are, colon tumor data (Alon et al., 1999), breast cancer microarray data (Van't Veer et al., 2002) and central nervous system (CNS) dataset (Pomeroy et al., 2002). Gene expression datasets comprise thousands of genes and hundreds of conditions for mining functional and class information is becoming highly significant. Genes that behave similarly may be co-regulated and belong to a common pathway or a cellular structure. Clustering genes, groups similar genes into the same cluster based on a proximity measure. In gene-based clustering, the genes are treated as objects and samples are the features. The following sections will be briefly give an introduction about these datasets, while more detailed information can be found at http://www.rii.com/publications/2002/vantveer.html, the data resources are detailed in Table 3.3, which displays the main characteristics of these datasets.

Table 3.3 The main characteristics of high dimensional datasets

Dataset	Size	No. of Samples	Cluster Number
Colon tumour	2,000	62	7
Breast cancer	1,213	34	7
CNS	7,129	52	7

3.7.1 Colon Tumour

This dataset represents the expression levels for 6500 human genes across 62 samples used by Alon et al. (1999), it represents colon adenocarcinoma specimen collected from several patients, while normal tissues were also obtained from some of these patients. The top 2000 genes with highest intensity across the samples were selected. The resulting dataset contains 2000 genes across 40 tumorous and 20 normal colon tissues. Note that this dataset does not contain negative values and only 1909 of the 2000 genes are unique.

3.7.2 Breast Cancer

This gene microarray data contains 97 patient samples, among which 46 samples are from patients who had developed distance metastases within 5 years

(labelled as 'relapse'), the remaining 51 samples are from patients who remained healthy from the disease after their initial diagnosis for interval of at least 5 years (labelled as 'non-relapse'). The number of genes in this dataset is 24,481. The detailed information regarding this dataset is available from http://www.rii.com/publications/2002/vantveer.html.

3.7.3 CNS

This microarray data was originally provided by Pomeroy et al. (2002). The dataset employed in the experiments is the dataset C used to analyse the outcome of the treatment, which contains 60 patient samples, 21 are survivors (patients who were alive after treatment), and 39 are failures, (patients who succumbed to their disease). There are 7129 genes in the datasets. The associated data resource is http://www.broad.mit.edu/mpr/CNS/.

3.8 Evaluation Measure for High Dimensional Clustering

In order to evaluate and select an optimal clustering scheme namely compactness and separation is proposed (DeRisi et al., 1996). Compactness is defined as the number of each cluster that should be as close to each other as possible. A common measure of compactness is the variance. Separation is defined, as the clusters themselves should be widely separated. There are three common approaches to measure the distance between two different clusters: distance between the closest members of clusters, distance between the most distant members and distance between the centers of the clusters. The most widely used cluster validity indices for clustering high dimensional dataset is discussed next section.

3.8.1 Davies-Bouldin Index

Davies-Bouldin (DB) index is based on the similarity measure of clusters (R_{ij}) whose bases are the dispersion measure of a cluster (S_i) Si and the cluster dissimilarity measure (d_{ij}). DB index measures the average of similarity between each cluster and its most similar one. As the clusters have to be compact and separated, the lower DB index means better cluster configuration. It is a function of the ratio of the sum of within-cluster distance to between cluster separations. The DB index according (Davies & Bouldin, 1979; Pal & Bezdek, 1995) is defined as

$$DB = \frac{1}{K}\sum_{i=1}^{K} R_i \quad 3.14$$

Where $R_i = j = 1 \ldots . K^{max}(R_{ij}), i = 1 \ldots . . K$. The similarity measure of clusters (R_{ij}) is computed using equation 3.15:

$$R_{ij} = \frac{S_i + S_j}{d_{ij}} \quad 3.15$$

Where $d_{ij} = d(v_i, v_j)$the cluster dissimilarity measure and Si are computed as equation 3.16:

$$S_i = \frac{1}{C_i}\sum_{x \in C_i} d(x, v_i) \quad 3.16$$

3.8.2 Intra Cluster Distance

The process of evaluating the results of a clustering algorithm is called cluster validity assessment. A good clustering method will produce high quality clusters with high intra cluster similarity and low inter cluster similarity. Sum of Intra Cluster Distances (SICD) is the most known evaluating criteria for clustering data (refer to section 3.6.3 for more details).

3.9 Summary

This chapter presented the research methodology adopted in this thesis. It is experimental based methodology to develop effective and efficient BH algorithm in order to solving problem of data clustering. First, the general steps in the research methodology were discussed. Then detailed description in each phase were provided including the tools used for validation. The two modifications of the Black hole were explained. The next chapter will describe the new variation of BH for clustering algorithms.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

In chapter three, the proposed variant of Black Hole (BH) algorithm has been designed, which is called Multiple Levy Flight Black Hole (MLBH) algorithm. MLBH consists of two main modifications, Levy Flight Black Hole (LBH), and Multiple Black Hole (MBH). Each modification can be considered as an individual algorithm for the optimization problems in general, and data clustering in particular. Therefore, MLBH with the two main components should be tested in order to evaluate their performance, in terms of finding the optimal solution of the mathematical benchmark test functions and searching for the best clustering results.

In this chapter, the results of all the above-mentioned algorithms are presented and discussed. Overall, the chapter is divided into three main sections; searching performance Clustering performance in both data types the normal and high dimensional datasets and the last part is the analysis and finding. The chapter started by evaluating LBH, MBH, and MLBH based on nine benchmark test functions, then the convergence rate for the two modification and the new variant of BH were presented. The clustering performance section divided into normal datasets and high dimensional datasets.

4.2 Experimental Settings

The performance of LBH, MBH and MLBH were evaluated by carrying out two main types of experiments. In the first experiment, all algorithms had been evaluated in terms of searching for the optimal solutions in a set of nine benchmark test functions. This set was divided into two main groups, unimodal and multimodal test functions.

The unimodal test functions (i.e., functions with only one global optima) were used to test the local search ability of the proposed algorithms. While the multimodal test functions (i.e., functions with more than one global optima) were used to test the global search ability of the algorithms, in other word, to test the ability of the algorithms to escape being trapped in local minima. Table 3.1 presented these test functions in the previous chapter.

In this experiment, the comparison stage is done by benchmarking against nine well-known metaheuristics they are:

- Big Bang–Big Crunch (BB-BC) (Erol & Eksin, 2006).
- Artificial Bees Colony (ABC) (Karaboga & Basturk, 2007).
- Particle Swarm Optimization (PSO) (Zambrano-Bigiarini et al., 2013).
- Levy Firefly Algorithm (LFFA) (Yang, 2010a).
- Grey Wolf Optimizer (GWO) (Mirjalili et al., 2014).
- Gravitational search algorithm (GSA) (Rashedi et al., 2009).
- Cuckoo Search (CS) (Yang & Deb, 2009).
- Black hole (BH) (Hatamlou, 2013).

The experiments for all algorithms were executed in 30 different runs as recommended by Farah and Belazi (2018), and the best, worst, mean, error rate, and standard deviation. Additionally, Table 4.1 showed the specific/default parameters for the metaheuristic algorithms used in this experiment. The parameter setting used in those algorithms are collected from the respective researcher works.

In the second experiment, all algorithms had been evaluated in terms of solving the problem of data clustering. This experiment was divided into two sub-experiment, first, the algorithms had been evaluated based on six normal-dimensional benchmark

datasets. Section 3.6.2 presented these datasets in details. The results are measured based on sum of intra clusters, and the value of error rate. The comparison stage was done by comparing the results of LBH, MBH, and MLBH against several well-known metaheuristic clustering algorithms: K-Means, Particle swarm optimization (PSO), Big Bang-Big Crunch Algorithm (BB-BC), Gravitational Search Algorithm (GSA), Black Hole Algorithm (BH) (Hatamlou, 2013), Genetic Algorithm (GA) , Tabu Search Algorithm (TS) , Ant Colony Optimization (ACO), Krill Herd (KH), Improved Krill Herd Algorithm (IKH) (Jensi & Jiji, 2016), Cuckoo Search algorithm (CS), Quantum Chaotic Cuckoo Search Algorithm (QCCA) (Boushaki et al., 2018) Artificial Bee Colony (ABC) (Yan et al., 2012), The Bat algorithm (BA), Harmony Search Algorithm (HS) (Abualigah et al., 2017b), Artificial Bee Colony and levy Distribution (ABCL) (Ghafarzadeh & Bouyer, 2016), K-harmonic means Algorithm and Imperialist Competitive Algorithm (ICAKHM) (Bouyer & Hatamlou, 2018), Grey Wolf Optimizer with Levy Flight steps (Kumar et al., 2017).

We need to mention that the population size is 25, which means 24 stars and one black hole in the original algorithm. But for the MBH and MLBH we only instead of choosing one black hole and the 24 stars we use 5 populations in each population 4 stars and one black hole so the number of population doesn't increase the number of stars.

Table 4.1 Parameter settings

Method	Parameters	Definition	Value
General	n	Population Size	25
	i	Iterations	1000
	N	No. of Runs	30
LFFA	β_0	Attractiveness	1.0
	γ	Light absorption coefficient	1.0
	α	Step scaling factor	0.2
	δ	Decreasing the value of a	0.96
PSO	ω	Inertia weight	0.742
	c_1, c_2	Personal Learning Coefficient	1.42
ABC	N	No. of Source	Size / 2
	$Limit$	Limit for scout bees	50
GWO	a	Balancing parameter	(2 –> 0.1)
GSA	G_0	Gravitational Constant	100
	β	Selection pressure	20
	ε	A small constant	2.22e-16
CSA	β	Levy exponent	1.5

The second part of the second experiment was the testing of ability of all proposed algorithms for handling and solving the problem of data clustering for high-dimensional datasets. The evaluation had been measured based on three high-dimensional datasets, Section 3.7 presented these datasets in details. The results are measured based on sum of intra clusters, the value of error rate, and Davies-Bouldin (DB) index. In this experiment, the three proposed algorithms had been compared against the original BH algorithm, and against each other as well.

Statistical analysis based on Friedman (Daniel, 1990) and Wilcoxon Signed Rank (Wilcoxon, 1945) will be conducted. This is to determine the significance of the results of the undertaken work. The rationale for adopting the Friedman and Wilcoxon Signed Rank stemmed from the fact that the obtained results are not normally distributed. This presented the need for non-parametric test.

Basically, the null hypothesis (H_0) for the Friedman test is that there is no significant difference between the terms of the obtained results for the selected datasets for the results sample at 95% confident level. Alternatively, the alternative hypothesis (H_1) is that there is a significant difference in terms of the results median. This means that the results median distribution is not equal (less or greater) for the sample. As Friedman test gives a general observation for all the results, a Wilcoxon signed rank test is needed to compare original BH results with the results of other proposed methods (LBH, MBH, and MLBH) individually.

4.3 Experimental Results

Cheng and Lien (2012) previously conducted experiments on all functions with a 500,000 maximum number of function evaluations. They reported any value less than $1e^{-12}$ as 0. To maintain comparison consistency, LBH, MBH, and MLBH were also tested using these same conditions. This section presents the results of the proposed algorithms, after executing and recording all the experiments over the 9 benchmark test functions, the outcomes showed that the proposed LBH, MBH and MLBH exerted superior performance and could reach the optimal solution for most test functions. Table 4.3 presents the results of LBH, MBH and MLBH and the other metaheuristics over the 9 test functions.

It is obvious to notice that MLBH outperformed all the other algorithms including the original BH in all benchmark test functions; especially the multimodal test functions (f_2, f_4, f_6). Meaning that, the optimized global search of BH by Levy flight (LBH) had enhanced the exploration ability of the algorithm and guided the stars towards better positions. The stars visit positions far from the black hole candidate or the current best solution, due to the possibility of generating varying step sizes via Levy Flight. Therefore, the stars avoid the possibility of trapping in the local optima. In conclusion, LBH has overcome the issue of the weak exploration and achieved the desired results of the first objective.

Moreover, MBH algorithm enhances the LBH algorithm for a better exploration and exploitation capabilities of the original BH algorithm; because of the multi-population architecture increases the chances of discovering better position at the first 50% of the iterations, and controls the search process by replacing the worst population with a completely new generated solutions/stars or with a new mixed population from the other populations. Therefore, MBH controls the amount of global search and local search abilities in the BH algorithm. As a result, MLBH is more stabilized version due to the two modifications LBH and MBH. In conclusion, MLBH has overcome the issue of the weak balancing between the exploration and exploitation in the original BH algorithm and achieved the desired results of the second objective. Figure 4.1 portrays a summarized comparison between all algorithms in terms of the number of successful tests.

4.3.1 Test functions comparison

For highlighting LBH, MBH and MLBH having superior exploration in comparison with the standard BH, further verification has been undertaken via a set of multi-model type of objective functions in a multi-dimensional space.

Table 4.2 Results of LBH, MBH and MLBH over benchmark test function from f1 to f9

Fun.	Statistics	BH	LBH	MBH	MLBH
	Best	0.00348	0.00330	**0.00000**	**0.00000**
f_1	Mean	0.09160	0.07189	**0.00000**	**0.00000**
	Std. Div.	0.00847	0.00819	**0.00000**	**0.00000**
	Best	0.00845	0.00839	**0.00000**	**0.00000**
f_2	Mean	0.08394	0.08384	**0.00000**	**0.00000**
	Std. Div.	0.01945	0.01955	**0.00000**	**0.00000**
	Best	0.02348	0.02337	0.01657	0.00697
f_3	Mean	0.03154	0.03146	0.02815	0.01999
	Std. Div.	0.00284	0.00259	0.00133	0.00129
	Best	1.2293	0.020580523	**0.00000**	**0.00000**
f_4	Mean	3.1853	0.069228159	**0.00000**	**0.00000**
	Std. Div.	0.024199	0.019449	**0.00000**	**0.00000**
	Best	0.00481	4.91E-05	**0.00000**	**0.00000**
f_5	Mean	0.08741	2.48E-04	**0.00000**	**0.00000**
	Std. Div.	0.03847	0.00031	**0.00000**	**0.00000**
	Best	0.001584	**0.00000**	**0.00000**	**0.00000**
f_6	Mean	0.009612	**0.00000**	**0.00000**	**0.00000**
	Std. Div.	0.084123	**0.00000**	**0.00000**	**0.00000**
	Best	0.12245	**0.00000**	0.12239	**0.00000**
f_7	Mean	0.26640	**0.00000**	0.26636	**0.00000**
	Std. Div.	0.05789	**0.00000**	0.05785	**0.00000**
	Best	3.59778	3.49999	2.89768	**0.00000**
f_8	Mean	3.94558	3.93899	3.98789	**0.00000**
	Std. Div.	0.87565	0.87558	0.85965	**0.00000**
	Best	0.00094	0.01745	0.00000	**0.00000**
f_9	Mean	0.00845	0.04478	0.00000	**0.00000**
	Std. Div.	0.05491	0.00648	0.00000	**0.00000**

Table 4.3 Results of LBH, MBH and MLBH over benchmark test function from $f1$ to $f9$

Fun.	Statistics	BB-BC	ABC	PSO	LFFA	GWO	GSA	CSA	BH	LBH	MBH	MLBH
	Best	4.1458	2.79E-16	3195.407	0.0077476	5.14E-30	0.00156	4.97E-04	0.00348	0.00330	**0.00000**	**0.00000**
f_1	Mean	5.9475	2.72E-16	4811.7969	0.21006	1.21E-28	0.02943	0.00105	0.09160	0.07189	**0.00000**	**0.00000**
	Std. Div.	2.1354	8.51E-12	588.3249	0.34752	1.76E-28	0.08790	4.41E-04	0.00847	0.00819	**0.00000**	**0.00000**
	Best	2.1049	5.68E-14	0.024484	4.94E-10	**0.00000**	0.90001	**0.00000**	0.00845	0.00839	**0.00000**	**0.00000**
f_2	Mean	3.3085	8.83E-13	1.1077	5.99E-08	**0.00000**	1.00043	**0.00000**	0.08394	0.08384	**0.00000**	**0.00000**
	Std. Div.	3.5478	2.76E-12	0.55972	5.18E-08	**0.00000**	0.90536	**0.00000**	0.01945	0.01955	**0.00000**	**0.00000**
	Best	3.45892	0.11531	1.3389	0.00409	1.85E-04	0.06348	0.01741	0.02348	0.02337	0.01657	0.00697
f_3	Mean	5.48953	0.19593	6.9606	0.02542	4.47E-04	0.08815	0.02845	0.03154	0.03146	0.02815	0.01999
	Std. Div.	0.83211	0.05549	0.6477	0.02312	2.11E-04	0.04413	0.00148	0.00284	0.00259	0.00133	0.00129
	Best	1.5829	0.02058	1.9877	0.0634	0.0692	2.86E-05	0.9900	1.2293	0.020580523	**0.00000**	**0.00000**
f_4	Mean	3.8331	0.15442	2.9439	1.9994	0.0366	0.0002763	0.9989	3.1853	0.069228159	**0.00000**	**0.00000**
	Std. Div.	1.0422	0.00000	0.037191	0.00013675	5.49E-10	0.556324	0.0497715	0.024199	0.019449	**0.00000**	**0.00000**
	Best	0.00064	5.04E-08	3.1901	0.0049316	4.60E-41	0.00493	5.82E-05	0.00481	4.91E-05	**0.00000**	**0.00000**
f_5	Mean	1.06309	1.76E-06	4.9583	0.010317	4.66E-05	0.02171	2.48E-03	0.08741	2.48E-04	**0.00000**	**0.00000**
	Std. Div.	1.79308	3.35E-06	1.4454	0.0039505	0.00017085	0.00928	0.00048	0.03847	0.00031	**0.00000**	**0.00000**
	Best	**0.00000**	4.44E-16	0.36776	3.20E-07	**0.00000**	**0.00000**	0.00019	0.001584	**0.00000**	**0.00000**	**0.00000**
f_6	Mean	**0.00000**	5.79E-04	1.6518	1.51E-06	**0.00000**	**0.00000**	0.00048	0.009612	**0.00000**	**0.00000**	**0.00000**
	Std. Div.	**0.00000**	0.0067	1.0598	1.88E-06	**0.00000**	**0.00000**	0.00082	0.084123	**0.00000**	**0.00000**	**0.00000**
	Best	0.89765	3.82E-16	8.8242	**0.00000**	0.0065555	15.3769	0.14548	0.12245	**0.00000**	0.12239	**0.00000**
f_7	Mean	0.56432	7.86E-16	11.8403	**0.00000**	0.024756	32366.20	1.16473	0.26640	**0.00000**	0.26636	**0.00000**
	Std. Div.	0.00318	1.61E-16	11.8403	**0.00000**	0.013532	59623.51	0.40721	0.05789	**0.00000**	0.05785	**0.00000**
	Best	4112.205	1.89E+02	246.5546	4.6417	4.40E-29	4214.467	3.55676	3.59778	3.49999	2.89768	**0.00000**
f_8	Mean	267.3249	2.48E+02	389.7976	17.9621	6.38E-27	345.7899	4.78767	3.94558	3.93899	3.98789	**0.00000**
	Std. Div.	189.7456	3.47E+01	72.833	6.80E+00	1.39E-26	189.7867	0.89787	0.87565	0.87558	0.85965	**0.00000**
	Best	2.12461	0.00432	1.2945	0.00128	0.00000	0.04871	0.57843	0.00094	0.01745	0.00000	**0.00000**
f_9	Mean	3.98452	0.00645	2.7707	0.00300	0.00000	0.06643	0.76741	0.00845	0.04478	0.00000	**0.00000**
	Std. Div.	2.64871	0.03184	1.0831	0.00105	0.00000	0.00384	0.68817	0.05491	0.00648	0.00000	**0.00000**

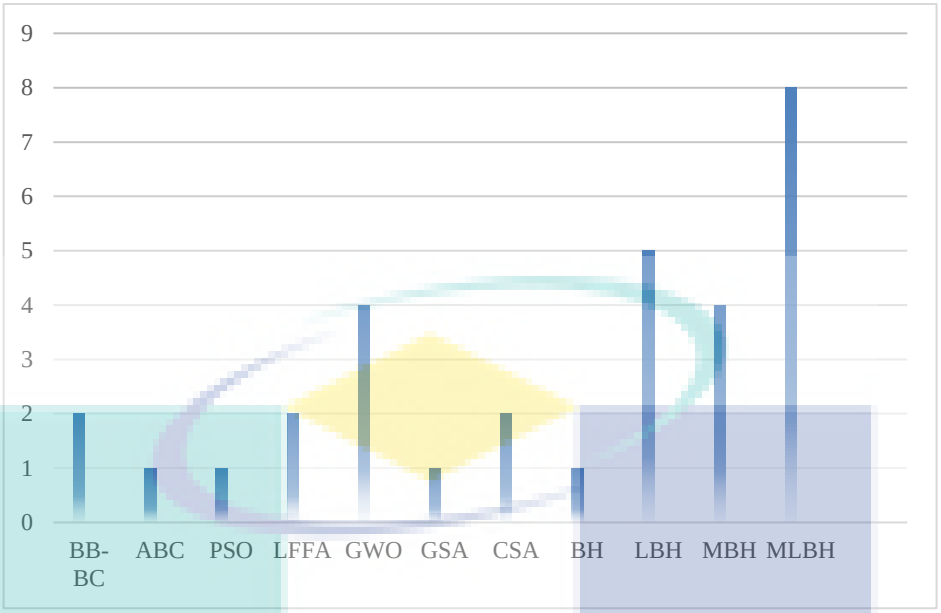


Figure 4.1 The results of all tests

All algorithms were evaluated 30 run times on 9 test functions, which means each algorithm were ran for a total of 270 times. Figure 4.1 illustrates that the MLBH arrived at the best solution for 8 tests, out of 9, while LBH ranked second with 5 best solutions. ON the other hand, GWO and MBH managed to solve 4 tests each, BB-BC, LFFA and CSA with 2 tests, and finally, PSO, ABC and BH solved only 1 tests, respectively.

4.3.2 Statistical Analysis for the Experimental Results

Although the statistical results presented in Table 4.3, which provides a first insight into the performance of MLBH, a pair-wise statistical test is typically used for a better comparison. For this purpose, by using the results obtained from 30 runs of each algorithm, a Wilcoxon Signed-Rank Test is performed with a statistical significance value of $\alpha = 0.05$. A Wilcoxon signed rank test is performed to compare the obtained results of the proposed algorithm such as LBH, MBH and MLBH against the original BH.

Generally, the null hypothesis (H_0) for Wilcoxon signed Rank test indicates that there is no significant median distribution between the mean pair of samples. The results are compared with other methods at a 95% level of confident. Here, if the Wilcoxon statistic is less or equal to the alpha ($\alpha = 0.05$), then, H_0 will be rejected. To perform the statistical calculations, the SPSS statistics Software Version 25 and

Microsoft Excel 2016 are used. In Table 4.4, the statistical results of LBH, MBH and MLBH algorithms compared to BH are given for f_6. Also the MLBH results compared to LBH and MBH are provided.

Table 4.4 Wilcoxon test

Comparison	Experiments setting			P-value	Decision
	Itre.	Run	S.size		
LBH vs BH	250	20	10	0.133895	Accept H_0
			20	0.000053	Reject H_0
			30	0.000051	Reject H_0
			40	0.000048	Reject H_0
MBH vs BH			10	0.000043	Reject H_0
			20	0.000053	Reject H_0
			30	0.000064	Reject H_0
			40	0.000053	Reject H_0
MLBH vs BH			10	0.000066	Reject H_0
			20	0.033895	Reject H_0
			30	0.000053	Reject H_0
			40	0.000051	Reject H_0
LBH vs BH	500	20	10	0.000094	Reject H_0
			20	0.000065	Reject H_0
			30	0.000067	Reject H_0
			40	0.000049	Reject H_0
MBH vs BH			10	0.000053	Reject H_0
			20	0.124158	Accept H_0
			30	0.000053	Reject H_0
			40	0.000043	Reject H_0
MLBH vs BH			10	0.000094	Reject H_0
			20	0.000094	Reject H_0
			30	0.000065	Reject H_0
			40	0.000067	Reject H_0
LBH vs BH	1000	20	10	0.000053	Reject H_0
			20	0.000273	Reject H_0
			30	0.000053	Reject H_0
			40	0.000080	Reject H_0
MBH vs BH			10	0.000050	Reject H_0
			20	0.000078	Reject H_0
			30	0.000053	Reject H_0
			40	0.000036	Reject H_0
MLBH vs BH			10	0.000012	Reject H_0
			20	0.000053	Reject H_0
			30	0.000273	Reject H_0
			40	0.000053	Reject H_0

Table 4.4 presented the Wilcoxon Signed Rank test for comparison of LBH vs BH, MBH vs BH, MLBH vs BH, MLBH vs LBH and MLBH vs MBH, the following observations are acknowledged as follows:

- LBH vs. BH: the Wilcoxon signed rank test in Table 4.4 shows that there is a significant difference between the obtained results of LBH against the original BH. Which indicates that all cases have been associated with rejected hypothesis.

- MBH vs. BH: the Wilcoxon signed rank test in Table 4.4 shows that there is a significant difference between the obtained results of MBH against the original BH.

- MLBH vs. BH: the Wilcoxon signed rank test in Table 4.4 shows that there is a significant difference between the obtained results of MLBH against the original BH

- MLBH vs. LBH: the Wilcoxon signed rank test in Table 4.4 shows that there is a significant difference between the obtained results of MLBH against the LBH

- MLBH vs. MBH: the Wilcoxon signed rank test in Table 4.4 shows that there is a significant difference between the obtained results of MLBH against the MBH

The time complexity of the three suggested algorithms is calculated as follows:

For LBH algorithm, it has the exact same time complexity of the standard version of BH, as it does not add any new loops or controlling parameters, which is :

$$O(N.T) \tag{4.1}$$

Where N represents the number of solutions, while T represents the number of iterations.

For MBH and MLBH, it almost have the same time complexity for BH and LBH, however, it requires two loops for the populations. In the first loop, the stars in each population are moved or updated, while in the second loop, the population are checked for the possibility of the replacing by new populations. It can be calculated as follows:

$$O(P.N.T) \qquad 4.2$$

4.3.3 Convergence Rate Analysis

The convergence curves for several test functions of LBH, MBH and MLBH and the other algorithms are provided in Figure 4.1- Figure 4.7 for the first 100 iterations. According to Figure 4.2, it can be noticed that LBH has converge more faster than all other algorithms in the comparison in test functions of (f_1 to f_6), the convergence of BH by Levy flight (LBH) had enhanced the exploration ability of the algorithm and guided the stars towards better positions rate. Which means that the stars avoid the possibility of trapping in local optima. Furthermore, the convergence rate of the MBH and the other algorithms in the comparison for the test function of (f_1 to f_6), is depicted in Figure 4.3, it is worth to mention that the convergence rate of MlBH has faster convergence than all other algorithms in the comparison for the six test function, which is clear evidence that the MlBH is capable to controls the global search as well as the local search abilities in the BH algorithm.

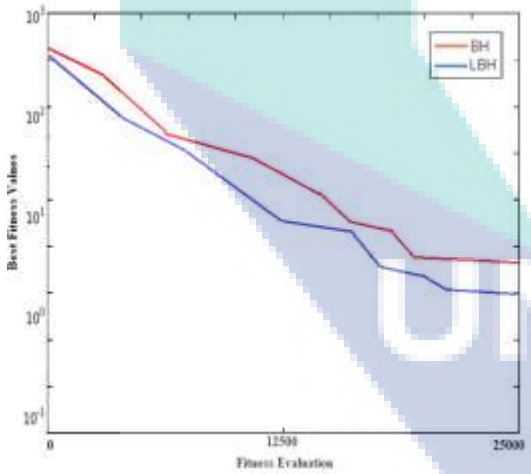


Figure 4.2a The convergence of (f_1)

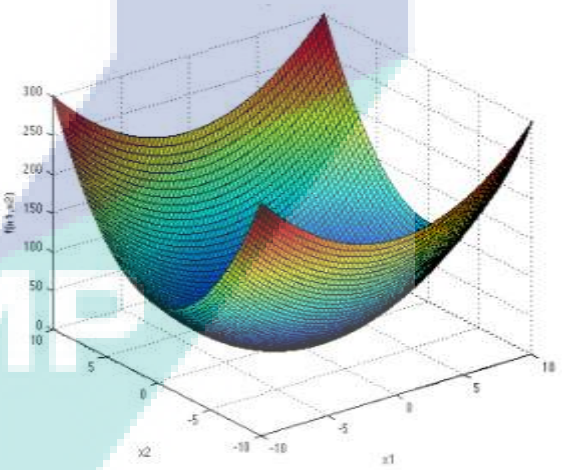
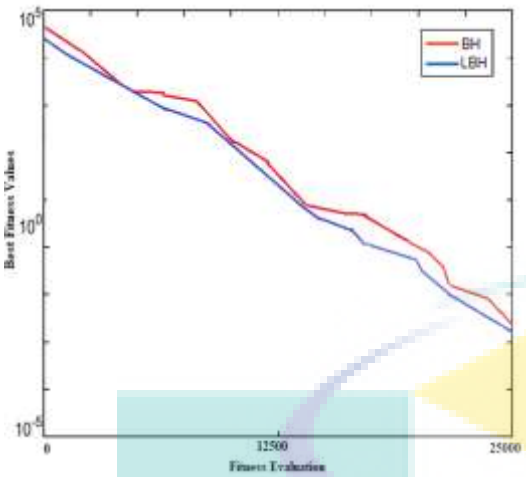


Figure 4.2b The 3D plot of sumsqaure (f_1)

Figure 4.2 The 3D plot of sumsqaure (f_1)with the convergence analysis of LBH and BH algorithm.



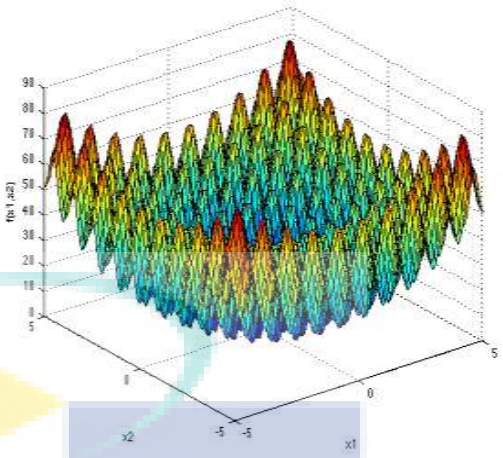
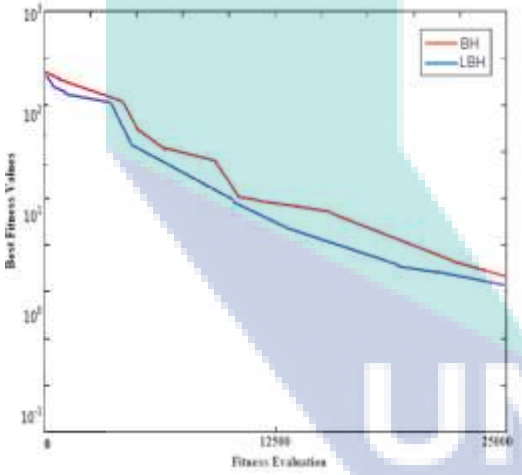


Figure 4.3a The convergence of (f_2)

Figure 4.3b The 3D plot of Rastrigin (f_2)

Figure 4.3 The 3D plot of Rastrigin (f_2)with the convergence analysis of LBH and BH algorithm.



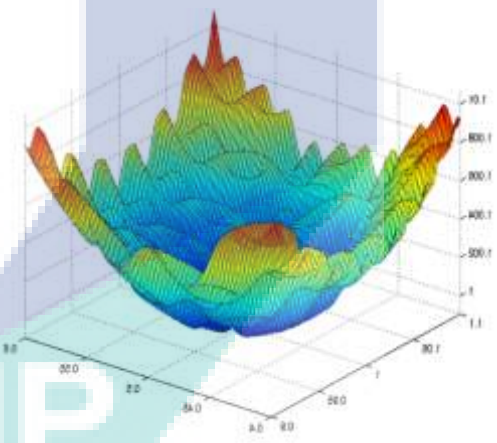


Figure 4.4a The convergence of (f_3)

Figure 4.4b The 3D plot of Quatric (f_3)

Figure 4.4 The 3D plot of Quatric (f_3)with the convergence analysis of LBH and BH algorithm.

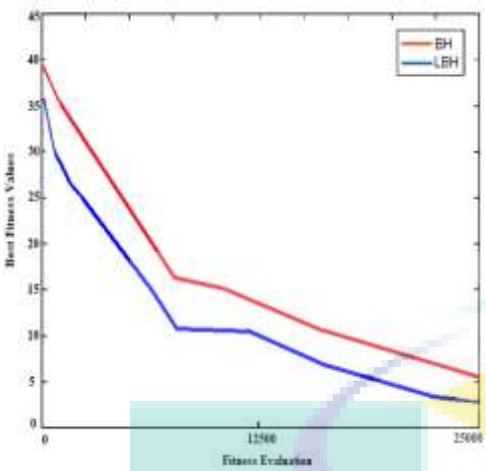


Figure 4.5a The convergence of (f_4)

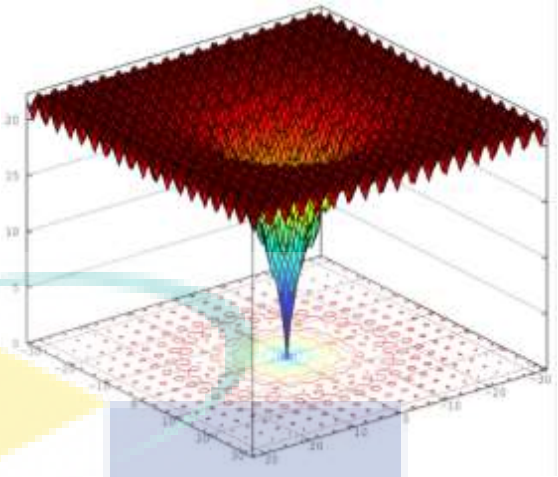


Figure 4.5b The 3D plot of Ackley (f_4)

Figure 4.5 The 3D plot of Ackley $(f_4)$with the convergence analysis of LBH and BH algorithm.

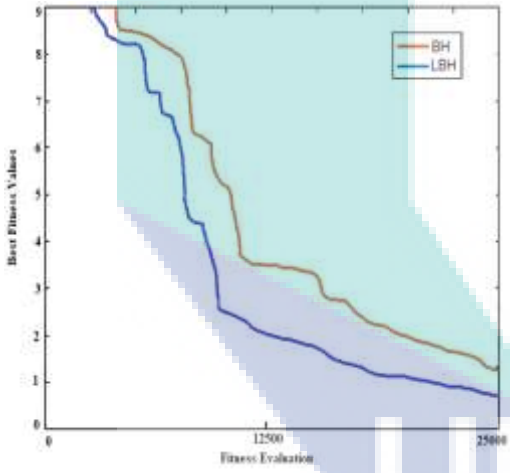


Figure 4.6a The convergence of (f_5)

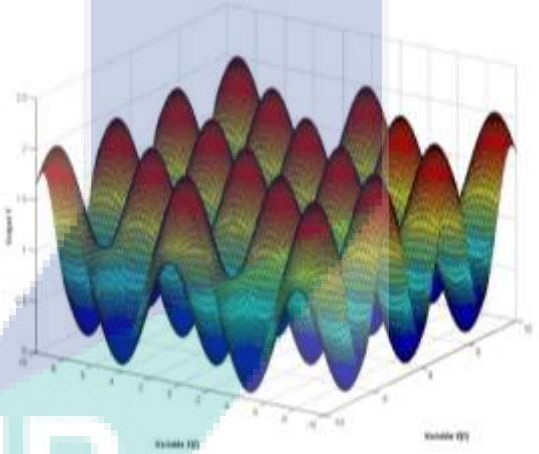
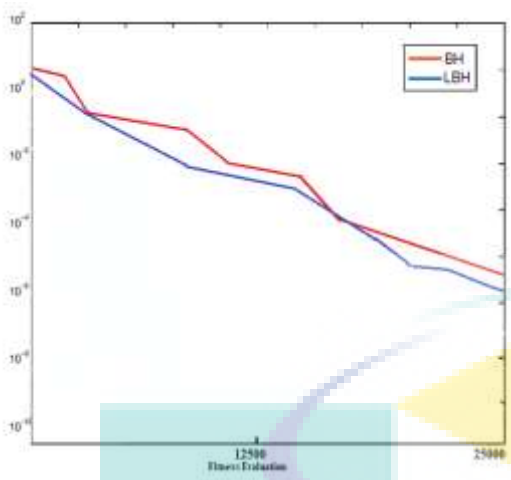


Figure 4.6b The 3D plot of Alpin N1 (f_5)

Figure 4.6 The 3D plot of Alpin N1 $(f_5)$with the convergence analysis of LBH and BH algorithm.



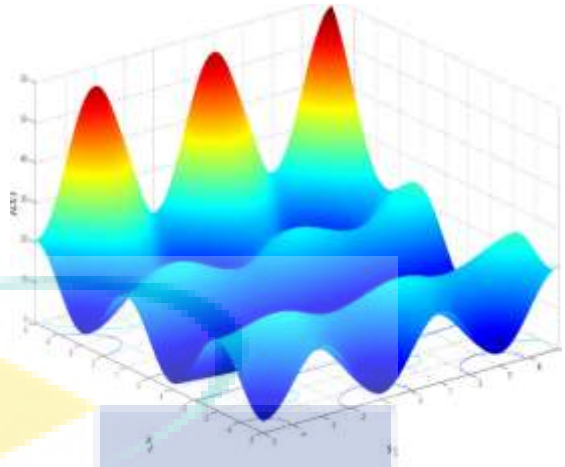
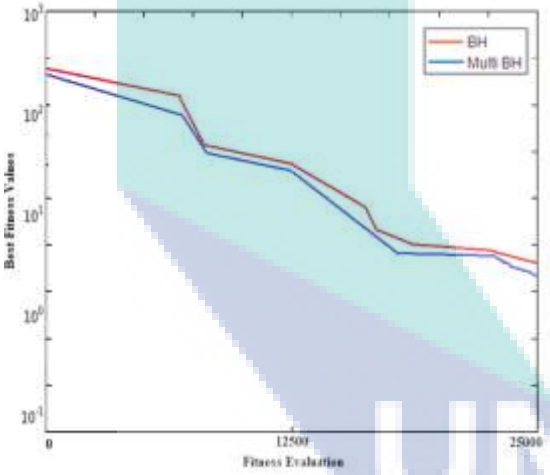


Figure 4.7a The convergence of (f_6) Figure 4.7b The 3D plot of Griewauk (f_6)

Figure 4.7 The 3D plot of Griewauk (f_6)with the convergence analysis of LBH and BH algorithm.



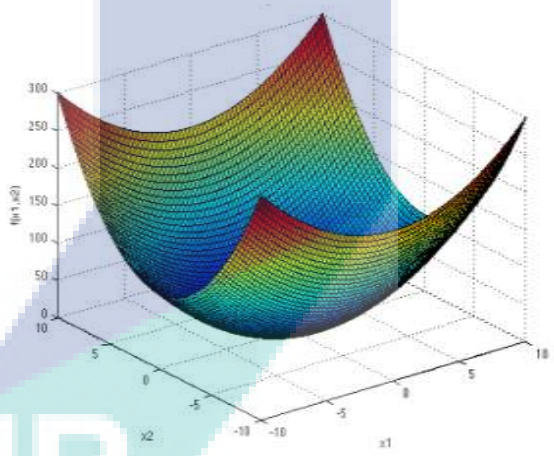
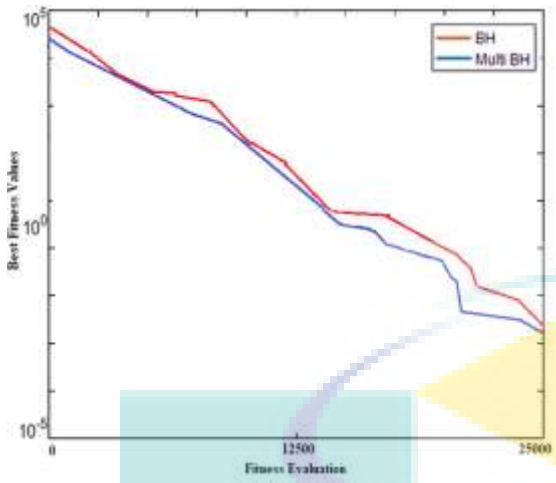


Figure 4.8a The convergence of (f_1) Figure 4.8b The 3D plot of sumsqaure (f_1)

Figure 4.8 The 3D plot of sumsqaure (f_1)with the convergence analysis of MBH and BH algorithms



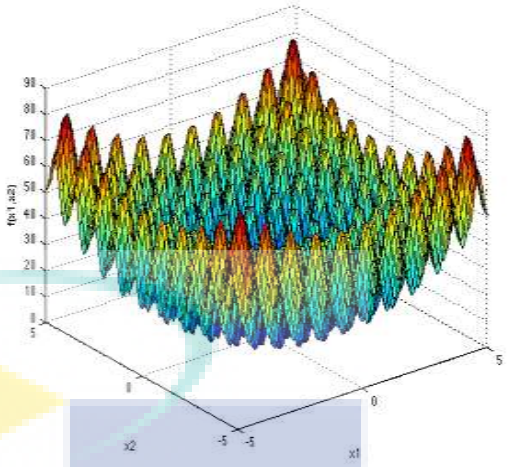
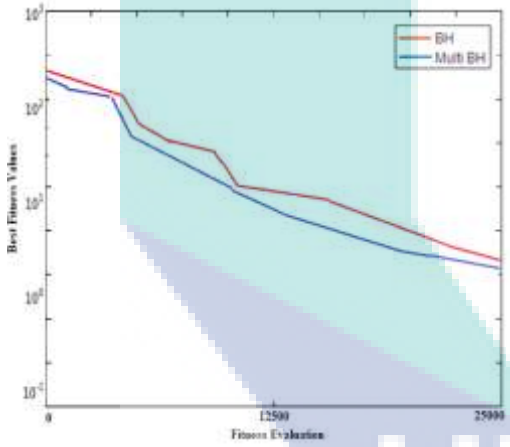


Figure 4.9a The convergence of (f_2)

Figure 4.9b The 3D plot of Rastrigin (f_2)

Figure 4.9 The 3D plot of Rastrigin (f_2)with the convergence analysis of MBH and BH algorithm.



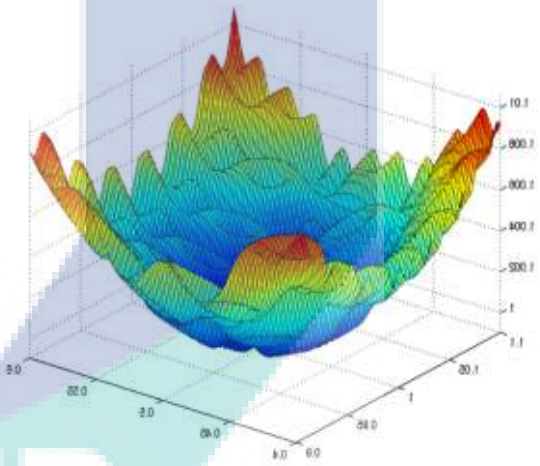


Figure 4.10a The convergence of (f_3)

Figure 4.10b The 3D plot of Quatric (f_3)

Figure 4.10 The 3D plot of Quatric (f_3)with the convergence analysis of MBH and BH algorithm.

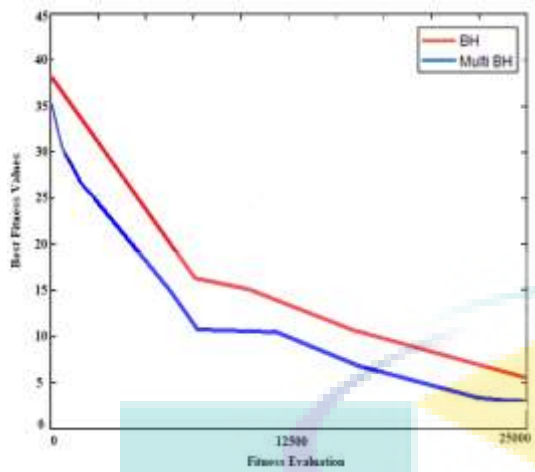


Figure 4.11a The convergence of (f_4)

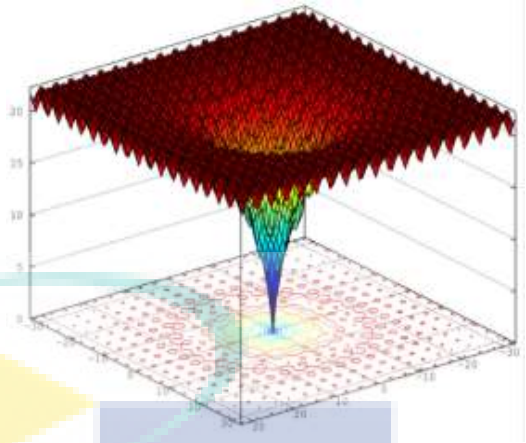


Figure 4.11b The 3D plot of Ackley (f_4)

Figure 4.11 The 3D plot of Ackley (f_4)with the convergence analysis of MBH and BH algorithm.

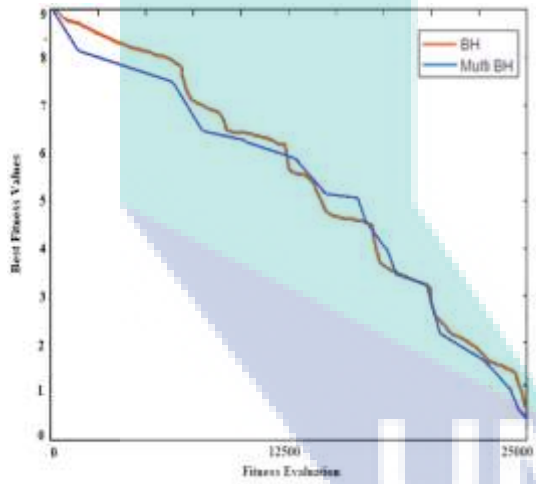


Figure 4.12a The convergence of (f_5)

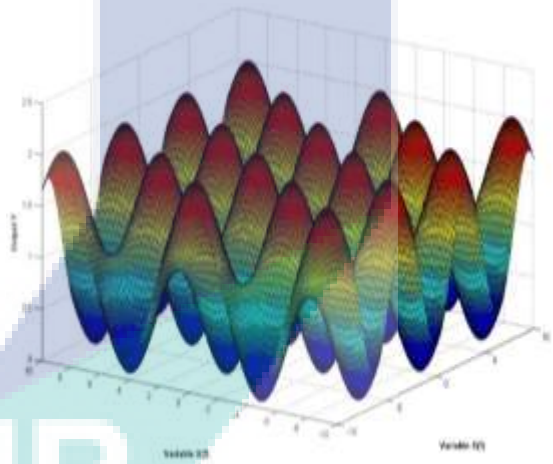
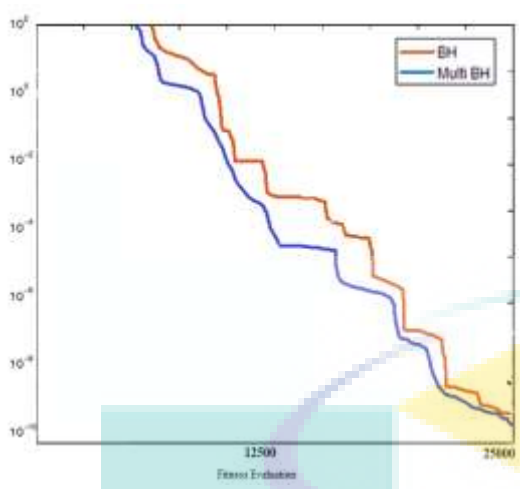


Figure 4.12b The 3D plot of Alpin N1 (f_5)

Figure 4.12 The 3D plot of Alpin N1 (f_5)with the convergence analysis of MBH and BH algorithm.



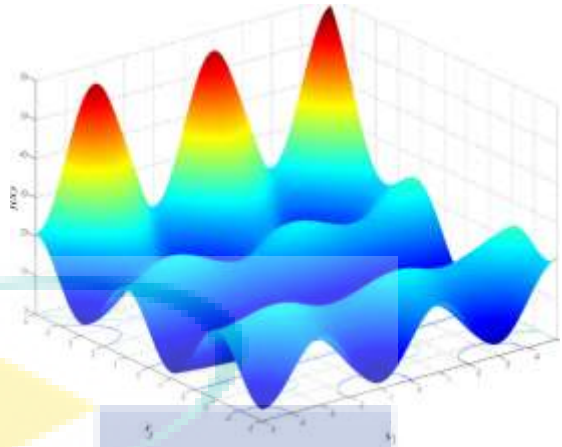
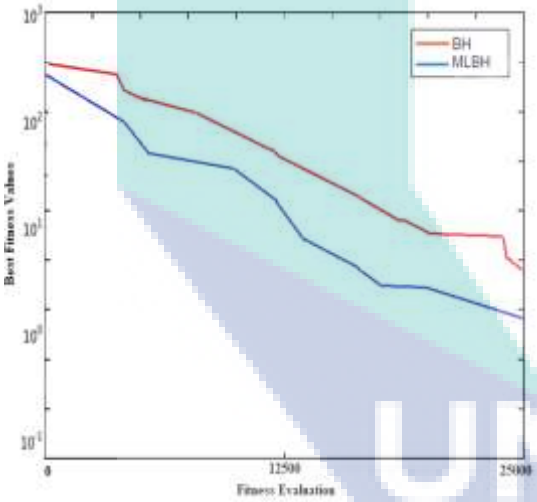


Figure 4.13a The convergence of (f_6) Figure 4.13b The 3D plot of Griewauk (f_6)

Figure 4.13 The 3D plot of Griewauk (f_6)with the convergence analysis of MBH and BH algorithm.



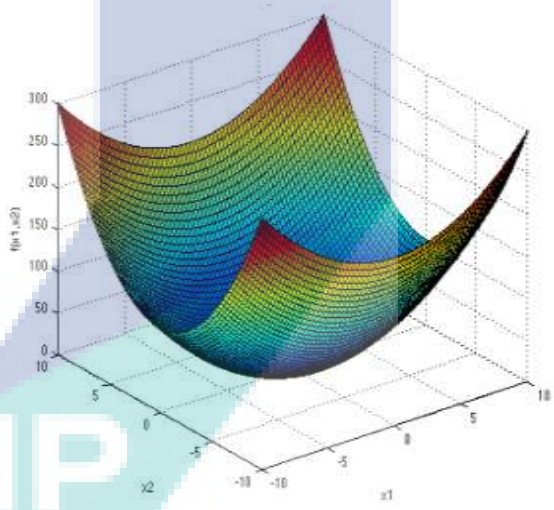
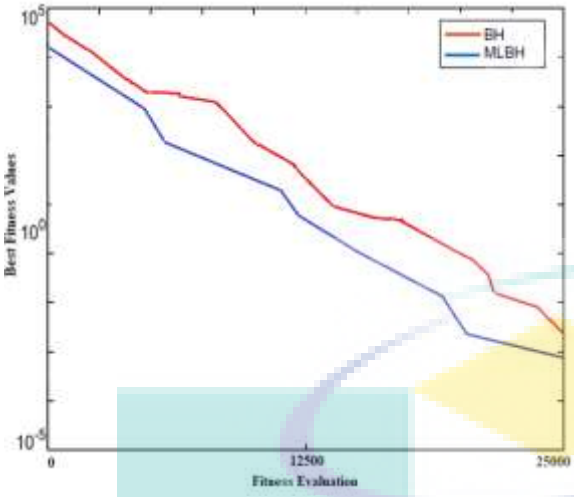


Figure 4.14a The convergence of (f_1) Figure 4.14b The 3D plot of sumsqaure (f_1)

Figure 4.14 The 3D plot of sumsqaure (f_1)with the convergence analysis of MLBH and BH algorithm.



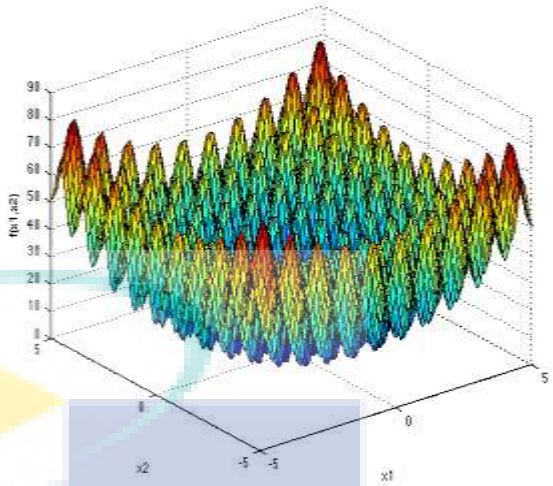
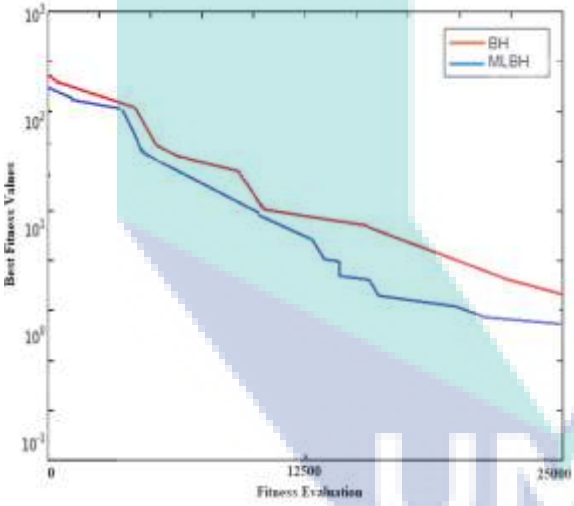


Figure 4.15a The convergence of (f_2)

Figure 4.15b The 3D plot of Rastrigin (f_2)

Figure 4.15 The 3D plot of Rastrigin (f_2)with the convergence analysis of MLBH and BH algorithm.



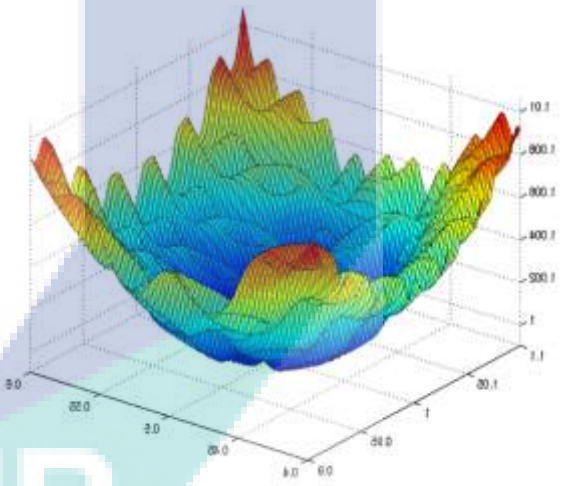
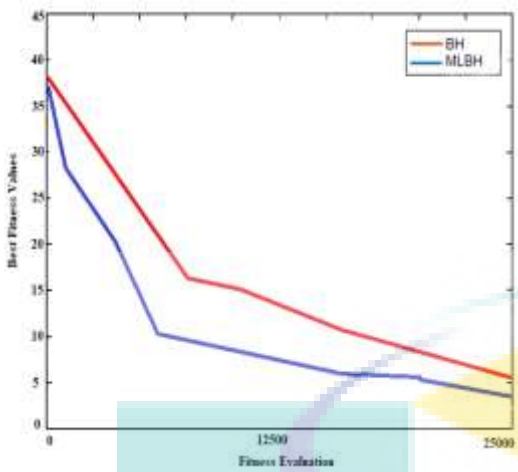


Figure 4.16a The convergence of (f_3)

Figure 4.16b The 3D plot of Quatric (f_3)

Figure 4.16 The 3D plot of Quatric (f_3)with the convergence analysis of MLBH and BH algorithm.



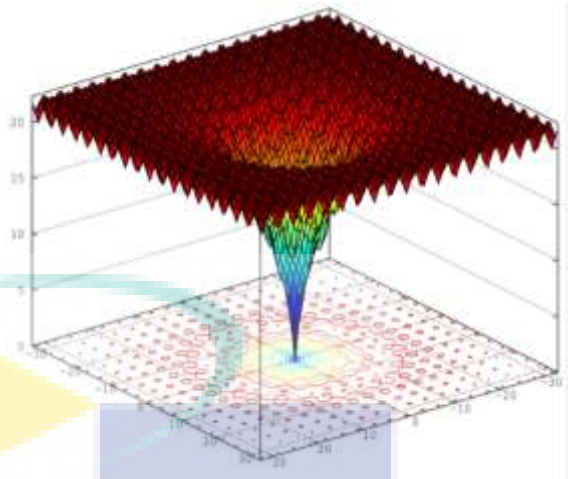
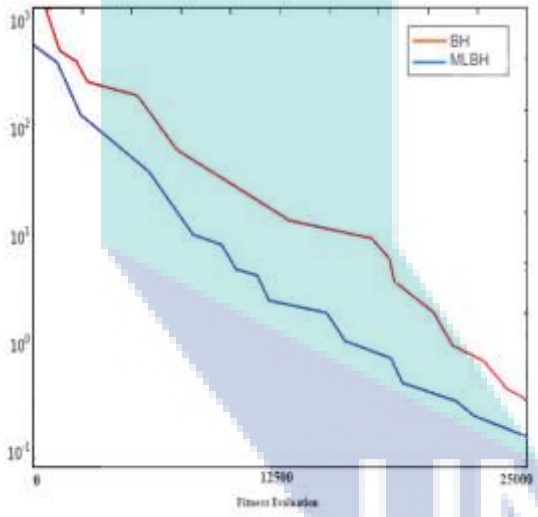


Figure 4.17a The convergence of (f_4) Figure 4.17b The 3D plot of Ackley (f_4)

Figure 4.17 The 3D plot of Ackley (f_4)with the convergence analysis of MLBH and BH algorithm.



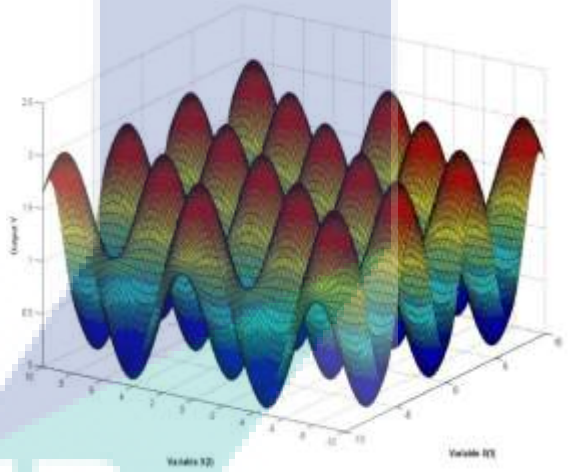
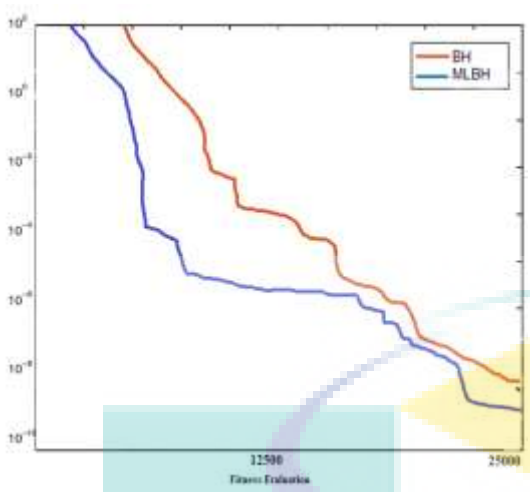


Figure 4.18a The convergence of (f_5) Figure 4.18b The 3D plot of Alpin N1 (f_5)

Figure 4.18 The 3D plot of Alpin N1 (f_5)with the convergence analysis of MLBH and BH algorithm.



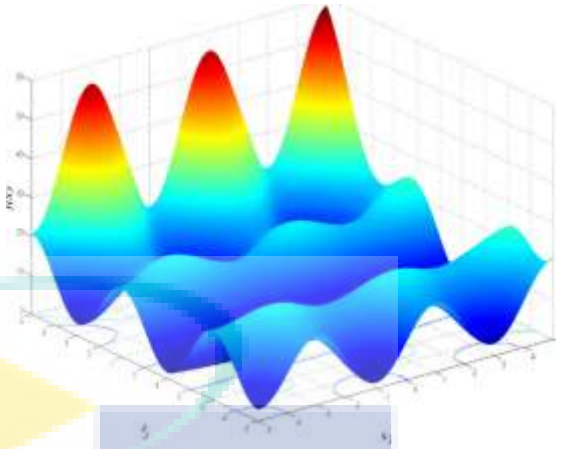


Figure 4.19a The convergence of (f_6) Figure 4.19b The 3D plot of Griewauk (f_6)

Figure 4.19 The 3D plot of Griewauk (f_6)with the convergence analysis of MLBH and BH algorithm.

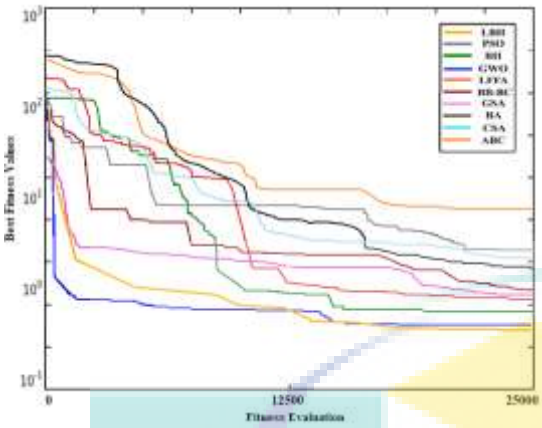


Figure 4.20a The convergence of (f_1)

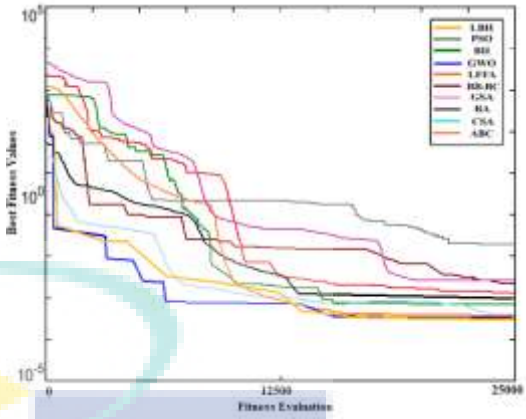


Figure 4.20b The convergence of (f_2)

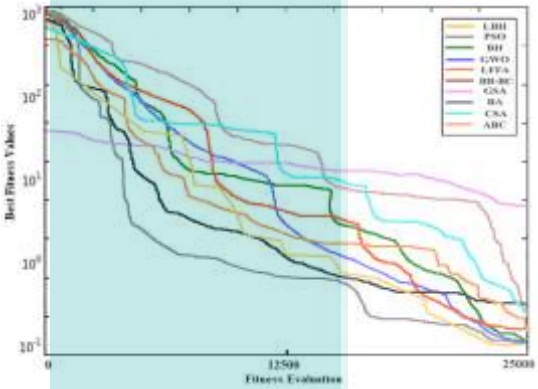


Figure 4.20c The convergence of (f_3)

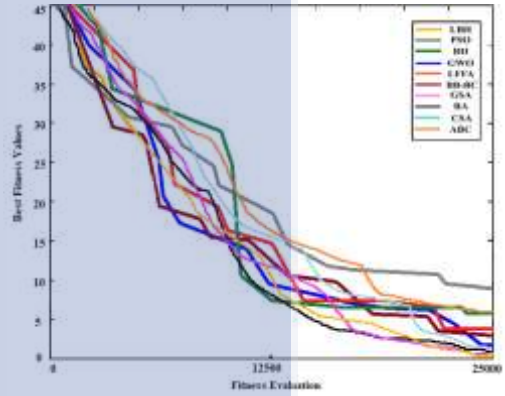


Figure 4.20d The convergence of (f_4)

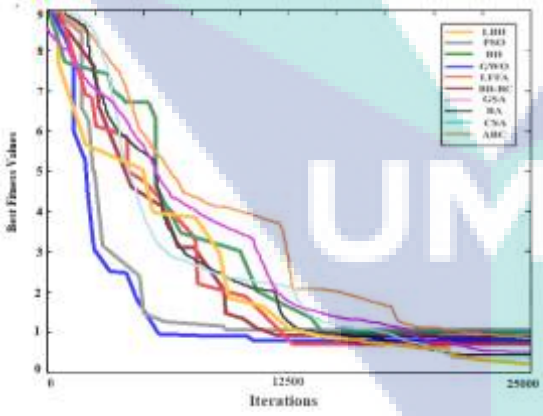


Figure 4.20e The convergence of (f_5)

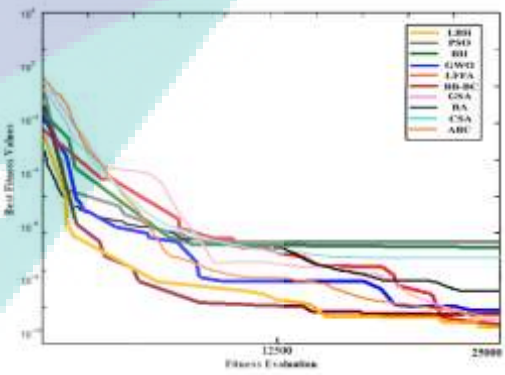


Figure 4.20b The convergence of (f_6)

Figure 4.20 Convergence analysis of LBH with other algorithms

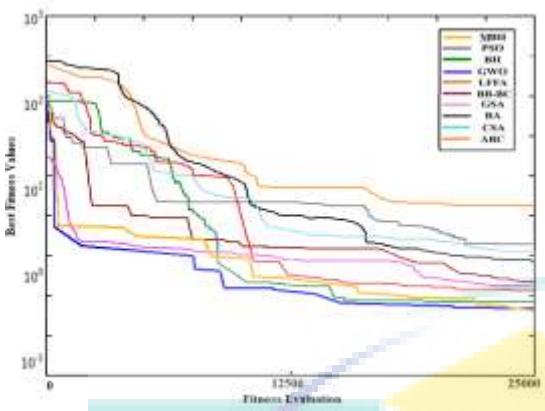


Figure 4.21a The convergence of (f_1)

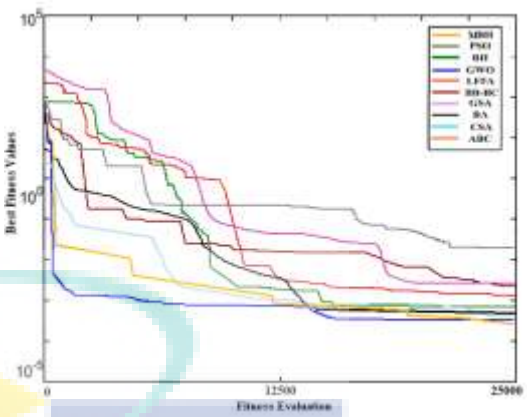


Figure 4. 21b The convergence of (f_2)

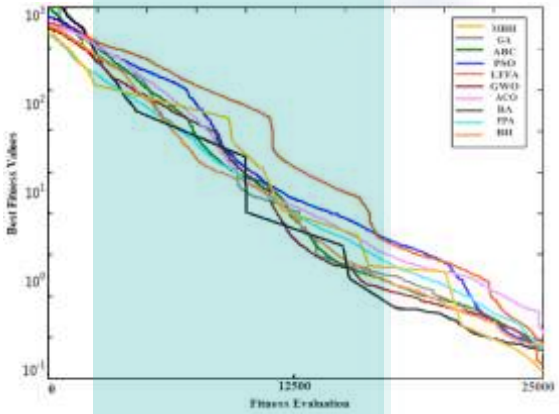


Figure 4. 21c The convergence of (f_3)

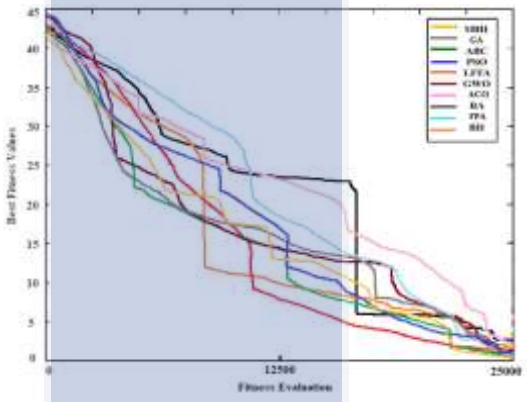


Figure 4. 21d The convergence of (f_4)

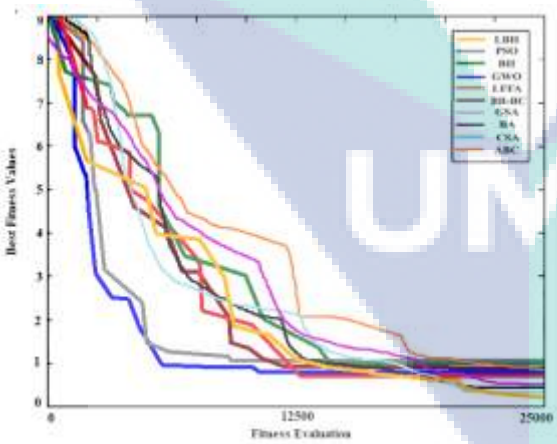


Figure 4. 21e The convergence of (f_5)

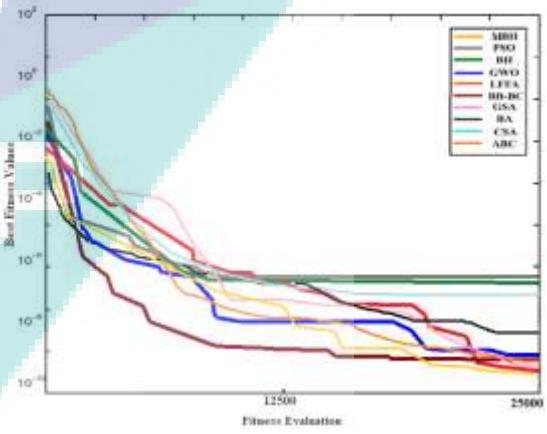


Figure 4. 21b The convergence of (f_6)

Figure 4.21. Convergence analysis of MBH with other algorithms

According to Figure 4.22, which depicts the convergence rate for the MLBH and the other benchmarking algorithms. It is worth to mention that the MLBH has better convergence than all other algorithms in the comparison for the six-test function, which also reveals that the MLBH is more version due to the two modifications LBH and MBH. In conclusion, MLBH has overcome the issue of the weak balancing between the exploration and exploitation in the original BH algorithm.

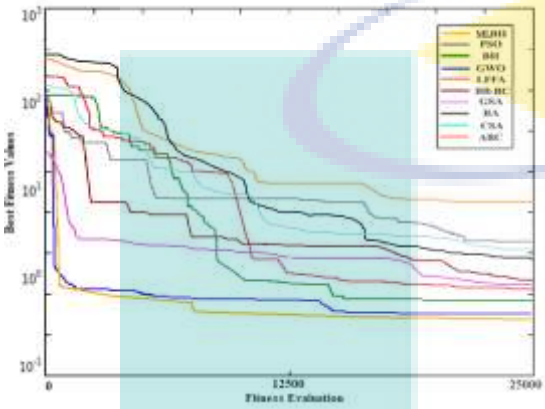


Figure 4.22a The convergence of (f_1)

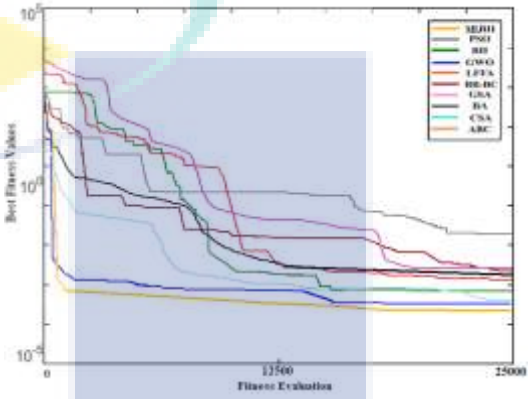


Figure 4. 22b The convergence of (f_2)

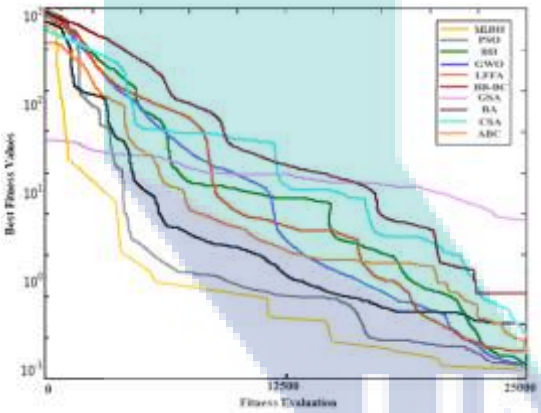


Figure 4. 22c The convergence of (f_3)

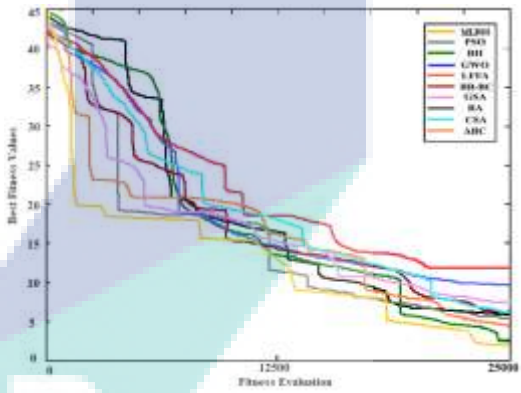


Figure 4. 22d The convergence of (f_4)

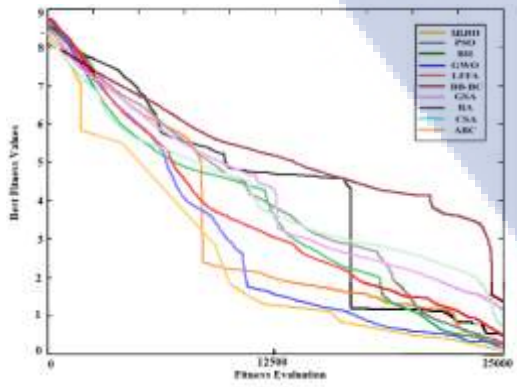


Figure 4. 22e The convergence of (f_5)

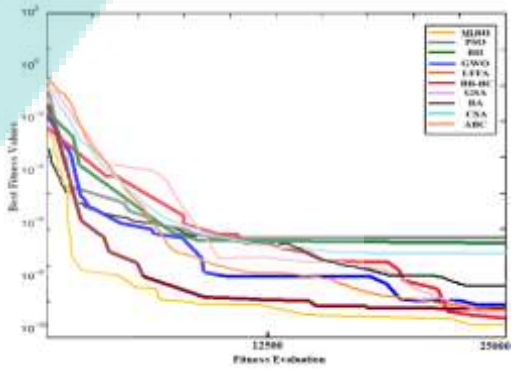


Figure 4. 22f The convergence of (f_6)

Figure 4.22. Convergence analysis of MLBH with other algorithms

According to the convergence rate presented in Figure 4.23 for the test function f_7 (Penalized), it can be seen that MLBH obtained faster convergence rate among the other algorithms. While, LBH has also obtained better convergence compared to the MBH and BH. Furthermore, the slowest convergence rate has been obtained by BH and LBH respectively.

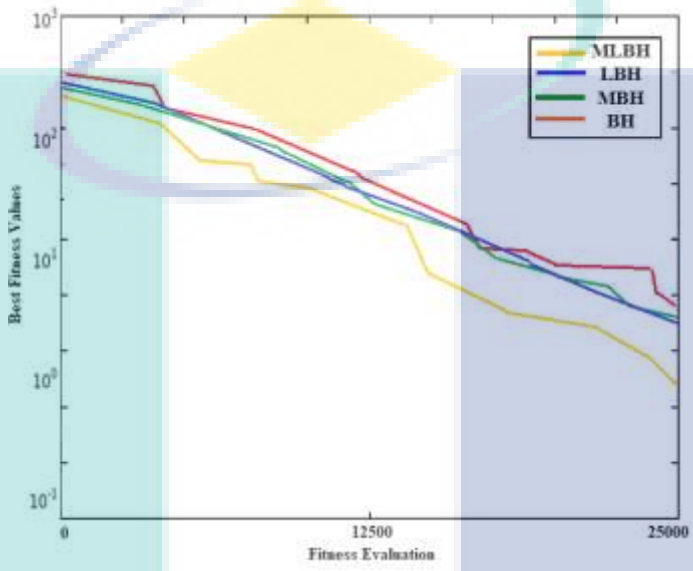


Figure 4.23 The convergence analysis of (f_7) for MLBH, LBH, MBH and BH.

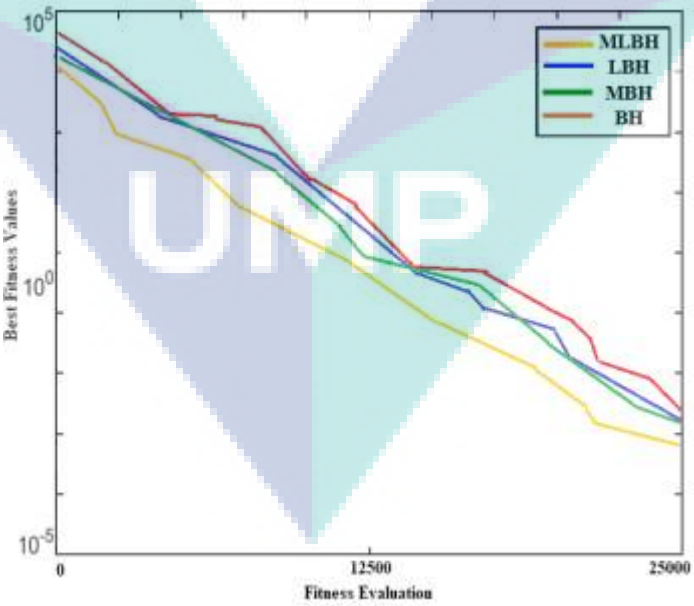


Figure 4.24 The convergence analysis of (f_8) for MLBH, LBH, MBH and BH.

According to Figure 4.24, which depicts the convergence rate for the test function f_8 (Zakharov), it is worth to mention that the MLBH has faster convergence rate during the search process, whereas, LBH and MBH have obtained different convergence rate for 30 iterations, then the convergence rate has been improved to some extent for both LBH and MBH respectively.

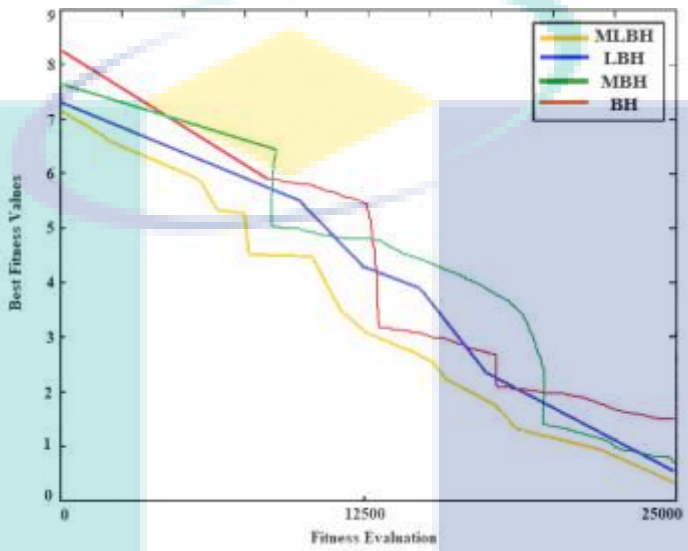


Figure 4.25 The convergence analysis of (f_9) for MLBH, LBH, MBH and BH.

According to Figure 4.25, MLBH and LBH performed the search process with faster convergence rate compared to other algorithms. BH and MBH performs the search with close convergence rate, then, MBH convergence rate has improved to some extent after certain amount of iterations.

4.4 Clustering Performance

In this section, the datasets used, and the number of clusters identified by different cluster validity indices after applied on well-known metaheuristic clustering algorithms. Instead of using just one technique, the newly proposed LBH, MBH and MLBH are tested and compared above-mentioned well-known metaheuristic clustering techniques.

4.4.1 Normal Datasets

To evaluate the performance of proposed algorithms (LBH, MBH and MLBH) for data clustering, six normal datasets have been used. The datasets, namely, Iris, Wine, Glass, Cancer, Contraceptive Method Choice (CMC) and Vowel. All data sets

are available from UCI machine learning laboratory. Table 4.5 to Table 4.10 shows the quality of the solutions found by clustering algorithms on above mentioned datasets. The sum of the intra-cluster distance is used to measure the quality of the resulting clusters as presented in equation 2.10. Clearly, the small value for the intra-cluster distance show the high-quality clusters and vice versa. The results are given in terms of the best, average and worst values of the intra-cluster distance after 30 independent runs for each of the six datasets. Moreover, the standard deviation of solutions (STD) for each algorithm is given to evaluate the reliability and stability of algorithms. A low standard deviation indicates that the respective algorithm is more reliable and stable to find optimal solution. Keeping in mind that the presented results of the algorithms in the comparison are collected from the respective researchers work.

4.4.1.1 Iris Datasets

Table 4.5 presents the obtained results for the Iris dataset. The evaluation of performance measurement has been conducted based on two criteria such as intra-cluster distances and error rate.

Table 4.5 The sum of intra-cluster distances and error rate obtained on Iris datasets.

Algorithm	Best	Average	Worst	Standard	Error rate
K-means	97.325	106.576	123.969	12.938	13.42
PSO	96.894	97.2320	97.897	0.347	12.58
GA	113.986	125.197	139.778	14.563	10.00
ACO	97.100	97.1710	97.808	0.367	10.32
ABC	95.616	95.8560	95.991	14.630	10.00
HS	98.648	98.4470	99.144	–	10.50
BAT	97.433	103.0360	108.870	3.410	10.78
GSA	96.687	96.73100	96.824	0.027	10.04
BB-BC	96.676	96.765	97.428	0.204	10.05
CS	97.983	102.513	106.760	2.182	09.80
TS	97.365	97.868	98.569	0.53	10.74
KH	96.655	96.655	96.655	1.9E − 06	10.00
IKH	96.655	96.655	96.655	9.8E − 06	9.78
QCCS	96.655	96.656	96.667	0.00266	09.43
ICAKHM	96.636	96.666	96.691	0.01055	11.23
EGWO	96.652	99.125	–	–	9.76
ABCL	–	96.655	–	1.351	10.45
BH	96.655	96.656	96.663	0.001	10.02
LBH	96.540	96.562	96.587	0.00014	9.40
MBH	96.533	96.522	96.539	0.00010	9.27
MLBH	**95.610**	**95.851**	**95.890**	**0.00005**	**8.98**

The simulation results given in Table 4.5, showed that the MLBH performs much better than BH and other methods for this dataset in term of the intra-cluster distances. Our proposed algorithm MLBH is able to achieve the best optimal value with a smaller standard deviation compared to other methods. Moreover for this dataset, LBH and MBH are able to converge to global optimum of 96.562, 96.522 for each run. While, the best solution of K-means, PSO, GA, ACO, HS, BAT, GSA, BB-BC, CS, TS, KH, IKH, QCCS, ICAKHM, EGWO, ABCL, and BH are 97.325, 96.894, 113.986, 97.100, 98.648, 97.433, 96.687, 96.676, 97.983, 97.365, 96.655, 96.655, 96.655, 96.636, 96.652, and 96.655 respectively. The standard deviation for LBH and MBH is zero, which is much less than other methods. On other hand, the error rate result shows that the MLBH obtained lowest error rate value compared to the proposed methods as well as the algorithms that been used as a benchmark. Moreover, the best results obtained by the all cluster algorithms in the comparison are depicted in Figure 4.26.

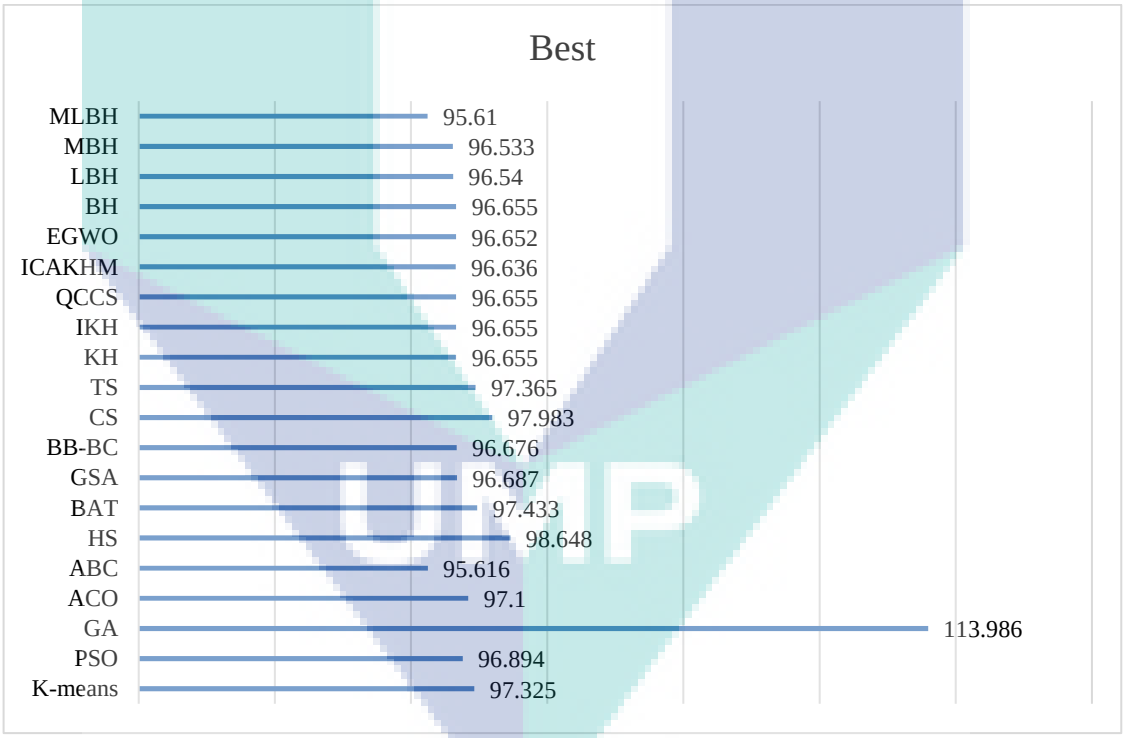


Figure 4.26 Best results of Iris datasets

4.4.1.2 Wine Datasets

The results for the wine dataset are presented in Table 4.5. The evaluation of performance measurement has been conducted based on two criteria such as intra-cluster distances and error rate. As illustrated in Table 4.6, in Wine dataset, the MLBH

algorithm has achieved superior results compared to MBH, LBH and BH algorithms and has better intra cluster distance value, also it has efficient standard deviation. The MBH is the second best and LBH has obtained almost the same result with slight difference compared to BH. Nevertheless, the proposed algorithms (LBH, MBH and MLBH) demonstrates better results compared to all other clustering algorithms in terms of intra cluster distance values as well as the standard deviation, as the standard deviation of LBH was 0.703, while for the MBH 0.697 and the MLBH 0.689. These results show that the proposed algorithms demonstrate very similar performance with other algorithms both in sum of intra-cluster distances and error rate. Moreover, the best results obtained by the all cluster algorithms in the comparison are depicted in Figure 4.27.

Table 4.6 The sum of intra-cluster distances and error rate obtained on Wine datasets.

Algorithm	**Best**	**Average**	**Worst**	**Standard**	**Error rate**
K-means	16,555.68	17,251.35	18,294.85	8.741	31.14
PSO	16,345.97	16,417.47	16,562.32	8.549	28.52
GA	16,530.53	16,530.53	16,530.53	3.410	28.76
ACO	16,530.53	16,530.53	16,530.53	–	28.43
ABC	16,306.00	16,306.00	16,306.00	2.213	29.55
HS	16,759.44	16,945.69	16,989.93	5.280	29.86
BAT	16,391.46	16,606.90	17,160.39	2.377	28.92
GSA	16,313.87	16,374.30	16,428.86	3.467	29.15
BB-BC	16,298.67	16,303.41	16,310.11	2.661	28.52
CS	16,363.12	16,420.81	16,525.72	4.554	29.10
TS	16,666.23	16,785.45	16,837.54	5.207	29.56
KH	16,292.19	16,579.66	18,293.60	4.249	29.78
IKH	16,292.21	16,294.30	16,292.84	0.706	28.90
QCCS	16,292.26	16,293.26	16,294.34	0.715	28.70
ICAKHM	16,293.90	16,295.60	16,296.94	1.002	28.73
EGWO	16,292.15	16,292.43	–	–	28.71
ABCL	–	16,295.30	–	1.097	29.80
BH	16,293.41	16,294.31	16,300.22	1.651	28.47
LBH	**16,291.99**	16,292.99	16,296.89	**0.703**	28.40
MBH	**16,289.34**	16,293.40	16,294.23	**0.697**	28.25
MLBH	**16,284.65**	16,286.27	**16,289.74**	**0.689**	26.31

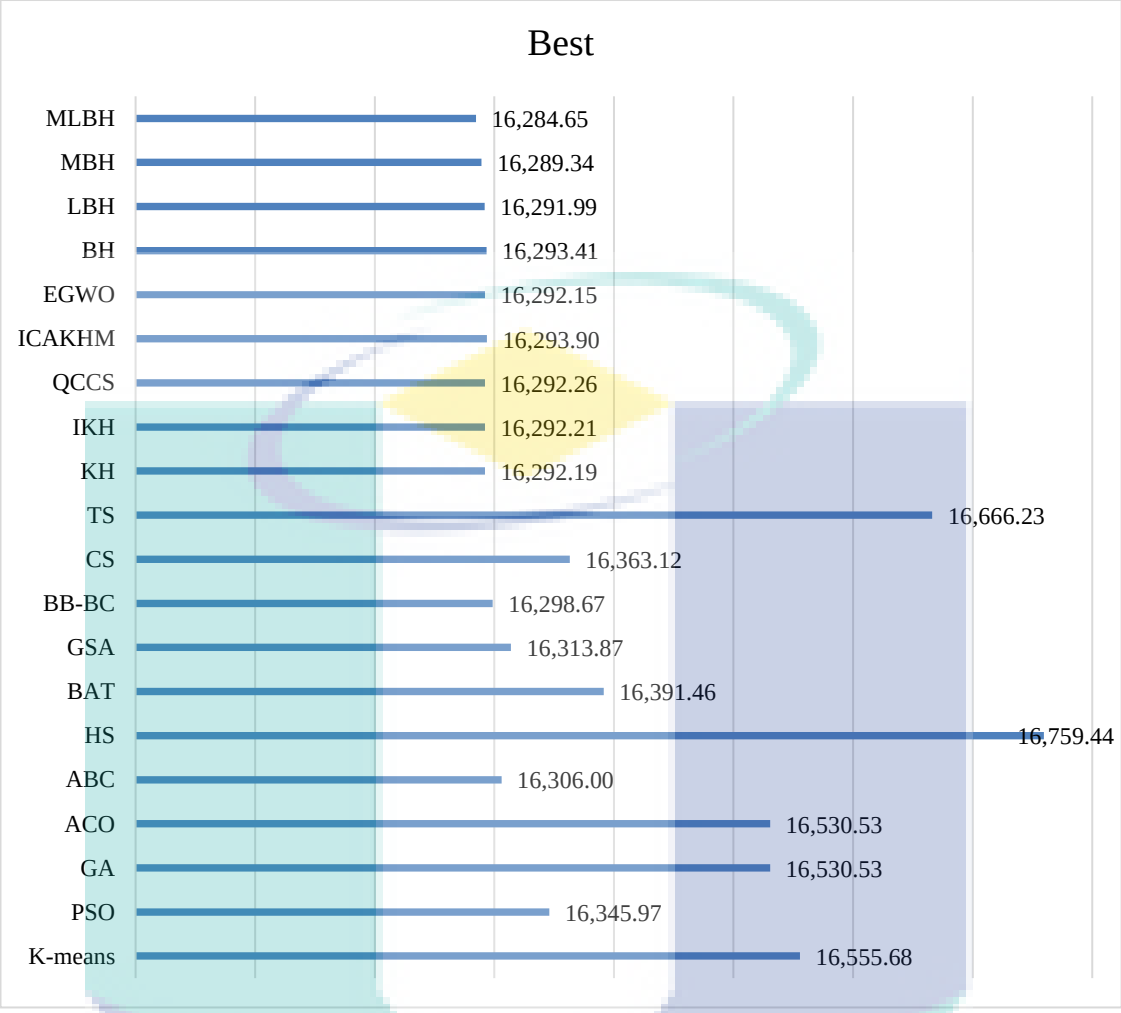


Figure 4.27 Best results of wine datasets

4.4.1.3 CMC Datasets

The results for the wine dataset for the proposed algorithms and the clustering algorithms in the comparison are presented in Table 4.6, the evaluation of performance measurement has been conducted based on two criteria such as intra-cluster distances and error rate.

Table 4.7 The sum of intra-cluster distances and error rate obtained on CMC datasets.

Algorithm	Best	Average	Worst	Standard	Error rate
K-means	5703.20	5705.37	5704.57	1.033	54.48
PSO	5700.98	5820.96	5923.24	46.959	54.49
GA	5756.59	5705,63	5812.64	50.369	57.68
ACO	5819.13	5701.92	5912.43	45.634	57.90
ABC	5695.67	5785.68	5899.00	10.200	56.78
HS	5698.56	5781.85	5814.86	5.280	56.00
BAT	5671.52	5802.14	5966.19	88.219	56.00
GSA	5542.27	5581.94	5658.76	41.136	55.67
BB-BC	5534.09	5574.75	5644.70	39.434	54.52
CS	5778.45	5962.09	6205.93	115.239	57.18
TS	5993.59	5885.06	5999.80	40.845	55.67
KH	5693.72	5737.23	6755.95	178.024	55.89
IKH	5693.72	5693.77	5693.73	**0.007**	55.90
QCCS	5532.22	5532.71	5535.29	0.134	57.11
ICAKHM	5699.21	5705.14	5721.17	1.268275	54.47
EGWO	–	–	–	–	–
ABCL	–	5533.77	–	0.85343	57.12
BH	5532.88	5533.63	5534.77	0.599	54.39
LBH	**5531.99**	5532.29	5532.58	**0.005**	54.35
MBH	**5530.80**	5530.00	5531.22	**0.003**	53.12
MLBH	**5527.45**	**5530.68**	**5531.47**	**0.001**	**52.98**

As illustrated in Table 4.7, in CMC dataset the MLBH algorithm has achieved the best performance in terms of the average, best, and worst inter-cluster distances compared to proposed algorithms as well as to the clustering algorithms in the comparison, in which the worst solution attained is 5500.47, This remained to be far superior to the best solutions obtained by the other algorithms. Moreover, MLBH has obtained the best results error rate value compared to the clustering algorithm with the lowest standard deviation. MBH achieved almost closed results to the LBH algorithm in terms of intra cluster distance, error rate and standard deviation. Nevertheless, the KH gives the worst optimization results on this dataset. The best results obtained by the proposed algorithm and all cluster algorithms in the comparison are depicted in Figure 4.28.

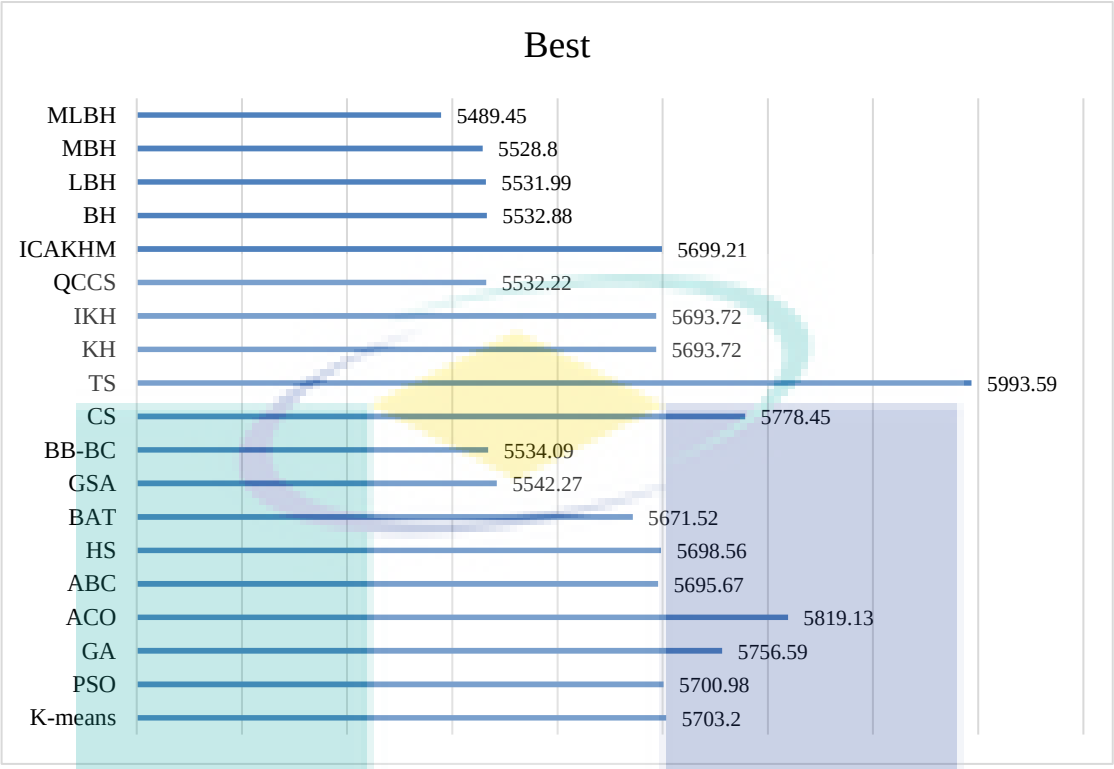


Figure 4.28 Best results obtained of CMC datasets

4.4.1.4 Cancer Datasets

Table 4.8 presents the results for the cancer dataset for the proposed algorithms such as (MBH, LBH and MLBH) and for the clustering algorithms in the comparison. As previously mentioned, that the evaluation of performance measurement has been carried out based on intra cluster distance and error rate. From Table 4.7, it is clear that in Cancer dataset, MLBH achieved superior performance over the proposed algorithms (MBH and LBH) and all other clustering algorithms in the comparison in terms of intra cluster distance, error rate and standard deviation, as MLBH has obtained standard deviation of 0.001 which is same result obtained by IKH and better than all the algorithms in the comparison. MBH is the second best and it's close to LBH. They are followed by ICAKHM, then, BB-BC, KH, IKH, QCCS and BH as the mentioned algorithms have obtained best optimization values of 2962.42 and 2964.38 respectively. The worst result obtained by LBH 2988.43. While, 2960.12 is the obtained value by MBH and the worst results obtained by MLBH was 2955.16. Moreover, the best results obtained by the proposed algorithms and all clustering algorithms in the comparison are depicted in Figure 4.29.

Table 4.8 The sum of intra-cluster distances and error rate obtained on Cancer datasets.

Algorithm	Best	Average	Worst	Standard	Error rate
K-means	2988.43	2988.99	2999.19	2.469	04.39
PSO	2973.50	3050.04	3318.88	110.801	05.25
GA	3249.46	2999.32	3427.43	229.734	03.87
ACO	3046.06	2970.49	3242.01	90.500	04.78
ABC	3576.87	3576.87	3576.87	0.020	03.93
HS	2988.85	2990.65	2998.28	45.640	**02.28**
BAT	3021.48	3107.12	3250.52	77.110	03.79
GSA	2965.76	2972.66	2993.24	8.918	03.74
BB-BC	2964.38	2964.38	2964.38	0.030	03.70
CS	3089.77	3200.79	3476.06	102.964	04.94
TS	3251.37	2982.84	3434.16	232.217	03.65
KH	2964.38	2971.97	3580.31	62.261	05.21
IKH	2964.387	2964.39	2964.38	**0.001**	03.69
ICAKHM	2962.42	3022.81	3150.15	0.396	04.27
EGWO	2964.11	2964.49	–	–	03.75
ABCL	–	–	–	–	–
QCCS	2964.38	2964.41	2964.49	0.027	03.51
BH	2964.38	2964.39	2964.45	0.009	03.70
LBH	2961.95	2963.90	2988.43	0.007	03.65
MBH	**2957.77**	2958.68	2960.12	0.005	03.61
MLBH	**2951.20**	**2953.37**	**2955.16**	**0.001**	03.19

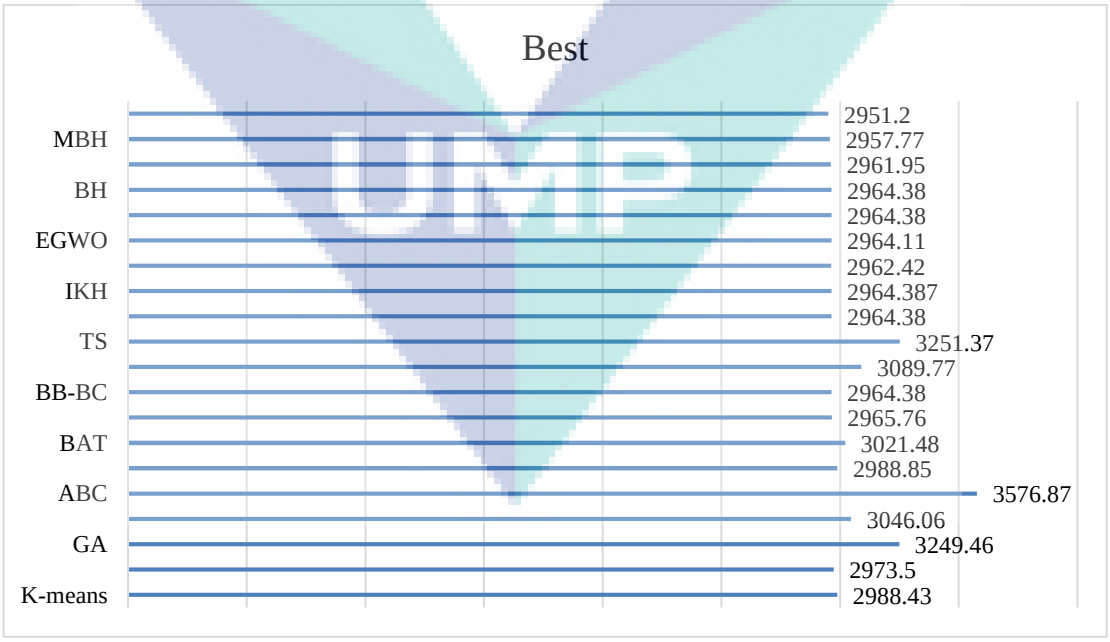


Figure 4.29 Best results obtained of cancer datasets

4.4.1.5 Glass Datasets

Table 4.9 presents the results for the Glass dataset. The presented results include the obtained results based on intra cluster distance and error rate as well as standard deviation for the proposed algorithms such as (LBH, MBH and MLBH) and the clustering algorithms in the comparison.

Table 4.9 The sum of intra-cluster distances and error rate obtained on Glass datasets.

Algorithm	Best	Average	Worst	Standard	Error rate
K-means	215.73	218.70	227.35	2.45	38.44
PSO	270.57	275.71	283.52	4.55	30.58
GA	282.32	278.37	286.77	4.13	38.67
ACO	273.46	269.72	280.08	3.58	40.34
ABC	230.55	254.55	267.65	11.49	30.50
HS	243.15	246.25	251.55	4.71	41.16
BAT	232.00	241.91	247.08	5.05	40.56
GSA	224.98	233.54	248.36	6.13	41.39
BB-BC	223.89	231.23	243.20	4.65	41.37
CS	220.12	225.19	227.02	5.66	41.89
TS	283.79	279.87	286.47	4.19	40.90
KH	210.24	215.72	251.27	5.44	41.78
IKH	210.25	222.80	215.93	2.73	33.90
ICAKHM	**199.86**	202.41	209.77	0.26	32.61
EGWO	214.42	242.43	–	–	33.60
ABCL	–	220.09	–	4.63	32.56
QCCS	–	–	–	–	–
BH	210.51	211.49	213.95	1.18	36.51
LBH	209.99	210.97	211.56	0.09	30.50
MBH	208.76	208.79	208.90	0.09	30.19
MLBH	207.90	**207.55**	**208.18**	0.08	28.54

From Table 4.7, it is clear that in Glass dataset, the best optimization value of 199.86 which was obtained by ICAKHM. MBLH is the second best in term of the best obtained optimization value, which also has slightly close results to LBH and MBH respectively. However, MLBH, MBH and LBH have obtained the lowest standard deviation compared to all clustering algorithm in the comparison. Moreover, the best error rate value has been achieved by the MLBH compared to the proposed algorithms as well as to all other clustering algorithms in comparison. Nevertheless, GA and TS obtained the worst optimization value of 286.77 and 286.47 accordingly. By contrast, all the proposed algorithms achieved similarly very close results in the sense of sum of

intra-cluster distance. While, the error rate result obtained by LBH, MBH and MLBH are better than the most clustering algorithms. Moreover, the best results obtained by the proposed algorithms and all clustering algorithms in the comparison are depicted in Figure 4.30.

Figure 4.30 Best results obtained of glass datasets

4.4.1.6 Vowel Datasets

Table 4.10 presents the vowel dataset result. The results presented in this Table includes two evaluation criteria such as intra cluster distance and error rate, standard deviation has been also illustrated for the proposed algorithms such as LBH, MBH and MLBH and the clustering algorithms selected as benchmark. As illustrated in Table 4.9, the MLBH achieved superior optimization values in terms of best, worst, average and standard deviation of 148,900.88, 148,911.56, 148,919.59, 532.040 and 38.15. These obtained values were better than the proposed algorithms (MBH and LBH) and all other clustering algorithms in the comparison. MBH and LBH comes in the second best accordingly. While, TS obtained the worst value of 165,996.42, which is considered to be the worst result between all the clustering algorithms in the comparison. Furthermore, MLBH achieved the best error rate value compared to the proposed

algorithms as well as the other clustering algorithms, whereas, MBH and LBH achieved slightly close error rate values of 40.78 and 41.65 respectively. Moreover, the best results obtained by the proposed algorithms and all clustering algorithms in the comparison are depicted in Figure 4. 31.

Table 4.10 The sum of intra-cluster distances and error rate obtained on Vowel datasets.

Algorithm	Best	Average	Worst	Standard	Error rate
K-means	149,398.66	151,987.98	162,455.69	34,252.500	43.57
PSO	148,976.01	148,999.82	149,121.18	28,813.469	41.92
GA	159,153.49	149,513.73	165,991.65	31,055.445	42.87
ACO	159,458.14	149,395.60	165,939.82	34,853.816	41.90
ABC	–	–	–	–	–
HS	156,155.00	156.489.00	157,548.00	18,560.000	43.67
BAT	155,163.59	149,411.21	160,783.94	30,018.245	42.55
GSA	151,317.56	152,931.81	155,346.69	24,867.028	42.26
BB-BC	149,038.51	151,010.03	153,090.44	18,593.235	41.89
CS	149,417.31	150.186.12	150,841.40	15,763.697	42.41
TS	162,108.53	149,468.26	165,996.42	28,462.351	43.44
KH	148,967.24	150,035.98	158,503.04	17,078.424	42.89
IKH	148,967.24	158,600.52	150,172.42	17,324.516	41.56
ICAKHM	149,201.63	161,431.04	165,804.67	2746.041	41.98
EGWO	–	–	–	–	–
ABCL	–	149,600.5	–	1.1289e+0	41.90
QCCS	–	–	–	–	–
BH	148,985.61	149,848.18	153,058.98	13,069.537	41.65
LBH	148,965.64	149,466.52	149,484.69	1,297.647	41.21
MBH	148,941.18	148,943.67	148,949.48	799.890	40.78
MLBH	**148,900.88**	**148,911.56**	**148,919.59**	**532.040**	**38.15**

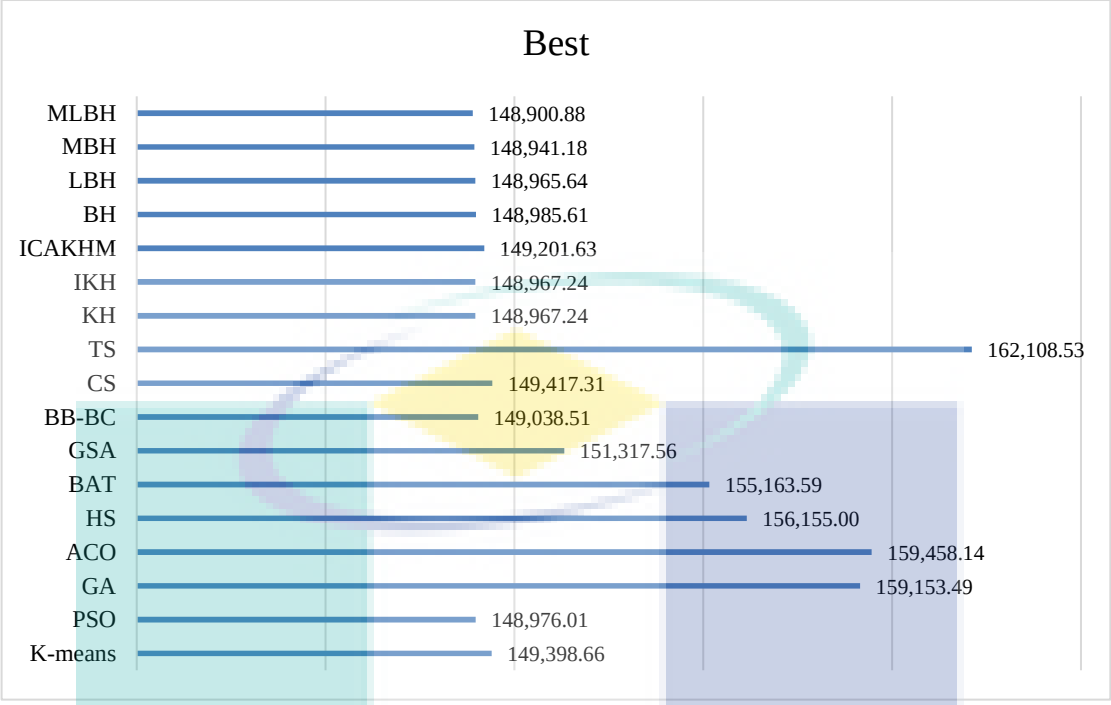


Figure 4.31 Best results obtained of Vowel datasets.

The statistical analysis performed using the Friedman is reported in Table 4.11. Form the statistical results, Friedman test indicate that the null hypothesis is rejected at 95% confidence level, which is clear evidence for the significant differences among the performances of the proposed clustering algorithms.

Table 4.11 Results of Friedman test based on the error rate

Test	Value	p-value	Results
Friedman test	11.9000	0.02481	Rejected

4.4.2 High Dimensional Datasets

This subsection illustrates the comparative analysis of clusters generated by black hole algorithm, Levy flight black hole (LBH), multiple population black hole (MBH) and multiple levy black hole algorithm (MLBH). Table 4.12, Table 4.13 and Table 4.14 shows the results of Colon Tumour, Breast cancer and CNS evaluated by sum of intra cluster distance and DB index as mentioned in 3.6.3 and 3.8 accordingly. All the proposed algorithms are executed for 250, 500 and 1000 iterations for 10, 20, 30, 40, 50 populations respectively.

Table 4.12 Comparison result between BH, LBH, MBH and MLBH on Colon Tumor Datasets.

Dataset	Iter.	Pop.	BH		LBH		MBH		MLBH	
			Intra	DB	Intra	DB	Intra	DB	Intra	DB
Colon Tumor	250	10	1845220.1176	163.735	1710613.2899	163.720	1718692.0786	163.701	1668266.7662	163.700
		20	1837532.7078	163.622	1710211.1433	163.598	1713167.0252	163.100	1642995.4386	163.003
		30	1818458.6271	162.899	1694453.7872	162.831	1691314.5003	162.293	1629180.5045	162.110
		40	1807681.8260	162.655	1685076.6086	161.846	1686681.7576	161.691	1595939.4406	161.612
		50	1800374.8607	162.437	1662208.8551	161.653	1677751.6127	161.453	1568777.3465	160.003
	500	10	1843935.1118	163.363	1708499.3873	163.370	1707819.1085	163.369	1651375.3796	162.365
		20	1830946.6141	162.139	1703560.2367	162.437	1701606.5781	163.233	1628639.9007	162.120
		30	1821699.8345	162.110	1675887.2657	162.109	1670451.9627	162.124	1601218.6316	161.400
		40	1803280.0171	161.398	1655296.3704	161.395	1656136.0851	161.589	1587835.9431	160.286
		50	1797511.8181	161.087	1653554.5530	160.385	1652385.1946	160.988	1581755.3851	158.980
	1000	10	1833411.9781	162.401	1698735.0989	162.403	1699715.8044	162.406	1630208.2575	162.380
		20	1827947.8411	161.966	1694939.7516	161.960	1698911.8589	161.956	1609867.2045	161.950
		30	1822045.9638	161.867	1682146.5878	160.864	1687451.6435	161.857	1597451.6601	159.850
		40	1789381.1057	161.373	1651336.6146	160.371	1656950.0397	160.367	1588752.0336	158.361
		50	1788757.1601	160.835	1641310.3810	160.035	1646848.3186	159.724	1586847.2700	158.115

According to experimental results presented in Table 4.11, the following observations are acknowledged:

- The first experiment is performed based on 250 iterations with population size of 10, 20, 30, 40 and 50. In the experiment where different numbers of iterations are used, the MLBH clearly outperformed the BH, LBH and MBH respectively for obtaining the best result in terms of quality solution. Table 4.11 shows that the quality solution of obtained results for MLBH improved to some extent when population size is increased. Moreover, MLBH obtained the best minimum DB index compared to the other proposed algorithm, it is observed that BH, LBH and MBH algorithms obtained same values of minimum DB index with slight difference.

- The second experiment is performed based on 500 iterations with population size of 10, 20, 30, 40 and 50 respectively. The increase number of iterations has slight effect on the quality solution for the obtained Intra results by all the proposed algorithms. MLBH achieved best Intra results for different number of populations, while there is a small difference for the minimum DB index values for the proposed algorithms as well as for the original BH algorithm. However, the best minimum DB index is obtained by the MLBH.

- The third experiment is performed based on 1000 iterations with population size of 10, 20, 30, 40 and 50 accordingly. The increase number of iteration has slight effect on the obtained Intra results for MLBH and MBH algorithm. While, there is some improvement on the obtained results of LBH. Moreover, the best minimum DB index is obtained by MLBH. Keeping in mind that there is a slight difference between the minimum DB index for MBH, LBH and BH respec

Table 4.13 Comparison result between BH, LBH, MBH and MLBH on CNS Datasets.

Dataset	Iter.	Pop. size	BH		LBH		MBH		MLBH	
			Intra	DB	Intra	DB	Intra	DB	Intra	DB
CNS	250	10	9253800.9852	44.494	9184098.9370	44.490	9182507.4068	44.487	9148614.0099	44.480
		20	9228713.0683	44.450	9105732.5859	44.447	9188850.4656	44.443	8855258.2266	44.433
		30	8583355.2429	44.416	8452024.6673	44.412	8444514.2479	44.413	8235156.8172	44.406
		40	8335006.1742	44.390	8266015.0489	44.387	8270169.9580	44.382	8131430.9051	44.371
		50	8773683.2227	44.218	8220835.5967	44.310	8269030.2494	43.380	8119339.2083	43.200
	500	10	9659065.3483	42.783	9087540.0317	42.780	9083247.6393	42.776	8948401.1025	42.772
		20	9175873.1268	42.676	8798261.4417	42.665	8894251.5065	42.662	8480556.7120	42.655
		30	8986447.4812	42.483	8625239.9557	42.386	8662390.9532	42.403	8263209.9966	41.987
		40	8844928.7958	42.215	8598544.165	41.989	8504997.494	41.970	7965622.0981	41.772
		50	8711558.8446	41.890	8484557.4342	41.854	8478236.8278	41.730	7853714.3792	41.459
	1000	10	9001930.3382	40.683	8997449.0511	40.210	8998395.9382	40.212	8863131.3097	40.206
		20	8736182.3228	40.426	8710268.0874	40.198	8732465.2648	40.181	8260474.0151	39.972
		30	8471992.8770	40.056	8136375.2002	39.950	8561653.6072	39.874	7796869.5953	38.877
		40	8321228.5659	39.890	8136015.9570	39.869	8106310.7939	39.751	7744184.3471	38.420
		50	8211578.2023	39.745	8025967.8655	38.775	7990962.2511	38.578	7706106.3618	38.331

According to experimental results presented in Table 4.12, the following observations are acknowledged:

- The first experiment is performed based on 250 iterations with population size of 10, 20, 30, 40 and 50. In the experiment where different numbers of iterations are used, the MLBH obtained best results compared to the proposed algorithms. MBH is the second best for best obtained results of intra cluster distance. LBH achieved better results compared to the original BH. Moreover, both MLBH and MBH have also obtained best minimum value of DB index with slight difference compared to other algorithms. While, there is also slight difference between the obtained results of LBH and BH in terms of minimum value of DB index.

- The second experiment is performed based on 500 iterations with population size of 10, 20, 30, 40 and 50 respectively. The increase number of population has slight effect on the quality solution for the obtained Intra cluster distance results by BH, LBH and MBH. While, the intra distance results of MLBH has improved when increasing the number of populations. Moreover, the increase number of populations has slight effect on the all proposed algorithms. However, MLBH still holding the best minimum value of DB index compared to other algorithms.

- The third experiment is performed based on 1000 iterations with population size of 10, 20, 30, 40 and 50 accordingly. The increase number of iterations has clearly improved the obtained for MLBH in terms of intra cluster distance. MBH and LBH achieved slightly close results and better than the obtained results by original BH. Moreover, MLBH, MBH and LBH have obtained slightly same results of the minimum value of DB index. Keeping in mind that there is no remarkable improvement on the results when increasing the number of populations.

Table 4.14 Comparison result between BH, LBH, MBH and MLBH on Breast Cancer Datasets.

Dataset	Iter.	Pop size	BH		LBH		MBH		MLBH	
			Intra	DB	Intra	DB	Intra	DB	Intra	DB
Breast Cancer	250	10	372775.3467	38.881	365160.8251	38.879	356502.6847	38.877	364548.0363	38.870
		20	364031.4425	38.776	354428.4748	38.771	355387.0528	38.770	352752.1064	38.766
		30	363957.0725	38.685	353702.2895	38.585	354200.6553	37.982	342066.8815	37.980
		40	362918.2229	38.422	352978.0609	37.858	352884.7683	37.858	349200.7146	37.850
		50	361058.9541	38.376	350973.5597	37.992	349370.5976	37.847	348677.4692	37.759
	500	10	361106.4442	38.918	364101.1975	38.914	364890.9362	38.912	361848.2368	36.912
		20	360907.5786	38.890	360862.0102	38.885	364214.3398	38.882	360214.3398	36.879
		30	359435.8007	38.860	358125.5003	38.857	359125.7609	38.856	359897.4658	36.853
		40	358979.4239	37.833	357818.5289	37.729	356020.0506	38.828	348978.0717	36.825
		50	357037.8433	37.798	342373.0402	37.695	344063.5666	37.990	332854.8446	35.986
	1000	10	369963.8410	37.740	367842.3827	37.736	367941.5340	37.732	367197.5154	35.723
		20	365898.1301	37.587	367755.5473	37.583	366857.1114	37.580	349160.3574	35.577
		30	359985.0636	37.230	359812.1738	37.227	356709.6411	37.322	346303.1346	35.219
		40	358718.3228	36.978	358230.4288	36.984	356601.5203	37.100	336630.7741	34.778
		50	358622.5744	36.787	349186.0266	36.682	344557.3041	36.361	325222.1861	34.271

According to experimental results presented in Table 4.13, the following observations are acknowledged:

- The first experiment is performed based on 250 iterations with population size of 10, 20, 30, 40 and 50. The obtained results of MLBH have been improved when increasing the number of populations compared to other algorithms which shows no big difference when increasing the number of populations. On other hand, all the proposed algorithms demonstrate slightly close minimum value of DB index.

- The second experiment is performed based on 500 iterations with population size of 10, 20, 30, 40 and 50 respectively. On this dataset, the best intra cluster improvement results when increasing the number of populations was belong to MLBH, whereas, there is small improvement when increasing the number of population for BH, LBH and MBH respectively. BH, LBH and MBH achieved closely same results in terms of minimum DB index. While, MLBH obtained the bets minimum values of DB index compared to the other proposed algorithms.

- The third experiment is performed based on 1000 iterations with population size of 10, 20, 30, 40 and 50 accordingly. It is worth to mention that the increasing number of populations demonstrated a clear effect on the obtained intra cluster distance of MLBH. While, the increase in the obtained intra cluster distance results are small compared with the required increase in population size of BH, LBH and MBH algorithms. Additionally, the best minimum value of DB index is obtained by MLBH as compared to other proposed algorithms. However, BH, LBH and MBH obtained closely same minimum values of DB index.

In Table 4.14, it is obvious that all the suggested modifications were better than the original BH. Overall, MLBH was the best, due to two main reasons. First, the integration between the mutual information with the clustering algorithm identifies the most relevant features, which their weights were more than the threshold value (i.e., zero). While the second reason was the unique structure of the MLBH which leads to handle the datasets with a large number of features even when mutual information was applied. Portray the comparison between the original BH with the other algorithms.

Table 4.15 The performance analysis of MLBH and other algorithms

Dataset	Size	Clustering algorithms	DB index	Intra-distance
Colon Tumor	2000	BH	160.835	1788757.1601
		LBH	160.035	1641310.3810
		MBH	159.724	1626848.3186
		MLBH	158.115	1586847.2700
Breast Cancer	24484	BH	36.787	358622.5744
		LBH	36.682	349186.0266
		MBH	36.361	344557.3041
		MLBH	34.271	325222.1861
CNS	7134	BH	39.745	8211578.2023
		LBH	38.775	8025967.8655
		MBH	38.578	7920962.2511
		MLBH	38.331	7706106.3618

4.4.3 Validity Threats

Empirical and experimental studies are often susceptible to numerous validity threats due to the dependence of the external and internal validity of such studies on the nature of the research. As such, this study is prone to such threats. The threats to external validity are encountered when the experiments are difficult to be generalized to real-world problems. In this experiment, external validity threats were eliminated by selecting the commonest and most realistic benchmark available in the literature.

Regarding threats to internal validity, they are threats with factors which can have a significant influence on the outcome of the study even without being noticed. Some of the common sources of such threats are variations in the population size, the number of iterations, as well as the parametric settings for each metaheuristic algorithms. Being that the source code for all implementations is not available, it may be wrong to state that the LBH, MBH, MLBH and the benchmarked algorithms comprised the same number of fitness function evaluations.

Additionally, our statistical analysis has been based on two statistical tests Wilcoxon-Rank and Friedman test and these tests requires the total sample for all the algorithms, therefore, the algorithm with missing value are ignored (Zamli et al., 2016).

4.5 Summary

This chapter presented the performance criteria of the LBH, MBH and MLBH. The proposed methods are compared against the existing optimization algorithms and clustering algorithms. Based on the content of this chapter, the next chapter will summarize the study findings, draw the conclusions and contributions, as well as provide a roadmap for possible future studies in this direction. Overall, results from this chapter allow us to conclude that the proposed clustering algorithms are efficient clustering algorithms in most datasets.

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 Introduction

This chapter presents the research summary of the thesis, the main contribution along with future work. The previous chapter covered LBH, MBH, and MLBH with a number of experiments in order to establish their true performance in terms of searching for the optimal solutions on a set of unimodal and multimodal tests functions, convergence rate, and finding the minimum intra-clustering distance in normal and high dimensional datasets. Based on the content of the previous chapters, this chapter highlights the impact of the results obtained, as well as the direction for future works.

5.2 Objectives Revisited

The aim of this research work was to design, implement and evaluate MLBH for addressing the problem of data clustering. The objectives of this research offort for fulfilling the stated aim were as follow:

i. To design a new variant of black hole (BH) algorithm with levy flight (called LBH)

ii. To improve the BH and LBH by introducing the multi-population support (called MBH) and its ensemble algorithm (called MLBH).

iii. To evaluate LBH, MBH and MLBH with existing meta-heuristic algorithms use standard functions and datasets.

This section attempts to answer the formulated research objectives in this study. With reference to objective one, a new modification of BH was proposed. This was

accomplished through the integration of BH with Levy flight, in which the new algorithm is called LBH. LBH was proposed to overcome the issue of the movement equation. The long jumps have been undertaken via Levy distribution in order to ensure effectual use of the search space in comparison with BH. Previously investigated works have aimed to improve BH, whereby the current proposal calls for BH to perform random walks and global search. Levy flight, in particular, improves the global search capacity for the BH algorithm, preventing one to be stuck in local minima. Additionally, the proposed modification enhances the global search ability of BH algorithm as per the new equation of star movements underlined. As BH algorithm is incapable of attaining the optimum results in a specific number of iterations, an efficient Levy-flight selection is imperative to avoid being stuck in local optimum as it results in improved global and local search capability concomitantly. In addition, the LBH achieved significant results as compared to the existing algorithms in the literature.

In terms of the second objective, LBH and MBH were integrated together to produce a new variant of BH algorithm that is called MLBH. In that, the MLBH is based on the enhanced version of LBH by using a multiple population instead of single population. The main difference between the original BH and MLBH is that MLBH contains a new mechanism for exploring the search space more than the original BH algorithm. Therefore, the chances of falling in the local optima are less when using MLBH for the global optimization problems in general, and data clustering problems in particular due to its ability of visiting positions, which are not explored by the standard version of BH.

For the final objectives, MLBH and its proposed ensemble in this research were successfully employed to undertake all the experimentation, highlighting their performance for the test functions and the data sets. The experimentation against several well-known methods helped to reveal the performance of the proposed algorithms in a seamless manner. In the conducted evaluations, all of the proposed algorithms presented successful results compared to the available optimization algorithms. The MLBH experimental results were more encouraging as it obtained the best results as compared to all the benchmarking algorithms.

5.3 Contribution

The contributions drawn from this study are as follows:

1. A new optimization algorithm based on the BH algorithm has been developed. It consists of two modifications (LBH and MBH), which form together the new variant of BH, it is called MLBH. The LBH has enhanced the global search ability (exploration) of the BH algorithm with Levy Flight, while the MBH has improved the global search ability (exploration) of the BH by using multiple population instead of single population.

2. This study showed that the MLBH has significantly outperformed the existing optimization algorithms such as BB-BC, ABC, PSO, LFFA, GWO, GSA, CSA, BH, LBH, and MBH by using both unimodal and multimodal. And that the MLBH has proven its effectiveness in solving the clustering problems in normal and high dimensional datasets.

5.4 Future Work

In this study, new variant of Black Hole algorithm was proposed (MLBH) to handle the data clustering problems in normal and high dimensional datasets. Although the proposed metaheuristic has been successfully applied to several benchmark test functions with promising results, there are several suggestions for future investigations.

A few clustering properties that need to be studied for the proposed methods like cluster stability (The clustering must remain stable whenever the data changes by only a small amount), effect and identification of outliers (a data different from other data) if present in data and the effect of fuzziness in data and deciding membership function. An outlier is a data item that is dissimilar from other data items. Outlier detection also leaves space for future research.

MLBH can be improved in many different ways. Currently, MLBH does not support multi objectives optimization problems. In wireless sensor network, density of deployment, scale, and constraints in battery, storage device, bandwidth and computational resources create serious challenges to the developers of WSNs. MLBH algorithm can give a model to solve optimization problems in WSNs due to its

simplicity, best solution, fast convergence and minimum computational complexity. These can be form important topics for further research.

REFERENCES

Abualigah, L. M., Khader, A. T., Al-Betar, M. A., and Hanandeh, E. S. 2017a. A new hybridization strategy for krill herd algorithm and harmony search algorithm applied to improve the data clustering. *Management.* 9(11).

Abualigah, L. M., Khader, A. T., Hanandeh, E. S., and Gandomi, A. H. 2017b. A novel hybridization strategy for krill herd algorithm applied to clustering techniques. *Applied Soft Computing.* 60, (pp.423-435).

Agarwal, P., and Mehta, S. 2019. Subspace clustering of high dimensional data using differential evolution. *Nature-Inspired Algorithms for Big Data Frameworks* (pp. 47-74). IGI Global.

Aggarwal, C. C., and Reddy, C. K. 2013. *Data clustering: algorithms and applications*. CRC press.

Akbari, R., and Ziarati, K. 2011. A cooperative approach to bee swarm optimization. *Journal of Information. Scince. Engineering.* 27(3), (pp. 799-818).

Alizadeh, A. A., Eisen, M. B., Davis, R. E., Ma, C., Lossos, I. S., Rosenwald, A., Yu, X. 2000. Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature.* 403(6769), (pp.503-524).

Aljarah, I., Mafarja, M., Heidari, A. A., Faris, H., and Mirjalili, S. 2019. Clustering analysis using a novel locality-informed grey wolf-inspired clustering approach. *Knowledge and Information Systems.* (pp.1-33).

Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D., and Levine, A. J. 1999. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences.* 96(12), (pp. 6745-6750).

Alswaitti, M., Ishak, M. K., and Isa, N. A. M. 2018. Optimized gravitational-based data clustering algorithm. *Engineering Applications of Artificial Intelligence.* 73, (pp. 126-148).

Arora, S., and Chana, I. 2014. A survey of clustering techniques for big data analysis. *Proceedings of the 5th International Confluence The Next Generation Information Technology*, (pp.59-65). IEEE.

Aslani, H., Yaghoobi, M., and Akbarzadeh-T, M.-R. 2015. Chaotic inertia weight in black hole algorithm for function optimization. *Congress on Technology, Communication and Knowledge*, (pp.123-129). IEEE.

Bagirov, A. M., and Yearwood, J. 2006. A new nonsmooth optimization algorithm for minimum sum-of-squares clustering problems. *European Journal of Operational Research.* 170(2), (pp. 578-596).

Bansal, J. C. 2016. Black hole artificial bee colony algorithm. *Proceedings of the 6th International Conference of Swarm, Evolutionary, and Memetic Computing, 9873* (pp.214-245). Springer.

Banu, P. N., and Andrews, S. 2015. Gene clustering using metaheuristic optimization algorithms. *International Journal of Applied Metaheuristic Computing (IJAMC).* 6(4), (pp.14-38).

Banu, P. N., Azar, A. T., and Inbarani, H. H. 2017. Fuzzy firefly clustering for tumour and cancer analysis. *International Journal of Modelling, Identification and Control.* 27(2), (pp.92-103).

Bartz-Beielstein, T., Chiarandini, M., Paquete, L., and Preuss, M. 2010. *Experimental methods for the analysis of optimization algorithms*. (pp.311-336). Springer.

Baskar, S., and Suganthan, P. N. 2004. A novel concurrent particle swarm optimization. *Proceedings of the Congress on Evolutionary Computation*.1, (pp.792-796). IEEE.

Bellazzi, R., and Zupan, B. 2007. Towards knowledge-based gene expression data mining. *Journal of biomedical informatics.* 40(6), (pp.787-802).

Bhaduri, A., and Bhaduri, A. 2009. Color image segmentation using clonal selection-based shuffled frog leaping algorithm. *Proceedings of the International Conference on Advances in Recent Technologies in Communication and Computing*, (pp. 517-520). IEEE.

Blackwell, T., and Branke, J. 2004. Multi-swarm optimization in dynamic environments. *Workshops on Applications of Evolutionary Computation*, (pp. 489-500). Springer.

Blackwell, T., and Branke, J. 2006. Multiswarms, exclusion, and anti-convergence in dynamic environments. *IEEE Transactions on Evolutionary Computation.* 10(4), (pp. 459-472).

Bouchekara, H. R. 2013. Optimal design of electromagnetic devices using a black-hole-based optimization technique. *IEEE Transactions on Magnetics.* 49(12), (pp. 5709-5714).

Boushaki, S. I., Kamel, N., and Bendjeghaba, O. 2018. A new quantum chaotic cuckoo search algorithm for data clustering. *Expert Systems with Applications.* 96, (pp. 358-372).

Bouyer, A. 2016. An optimized k-harmonic means algorithm combined with modified particle swarm optimization and Cuckoo Search algorithm. *Foundations of Computing and Decision Sciences.* 41(2), (pp. 99-121).

Bouyer, A., Ghafarzadeh, H., and Tarkhaneh, O. 2015. An efficient hybrid algorithm using cuckoo search and differential evolution for data clustering. *Indian Journal of Science and Technology*. 8(24) (pp. 489-498)..

Bouyer, A., and Hatamlou, A. 2018. An efficient hybrid clustering method based on improved cuckoo optimization and modified particle swarm optimization algorithms. *Applied Soft Computing*. 67, (pp**.** 172-182).

Branke, J. 1999. Memory enhanced evolutionary algorithms for changing optimization problems. *Conference of the Congress on Evolutionary Computation***.** *3* (pp.1875-1882). IEEE.

Chakravarthy, V., and Rao, P. M. 2015. On the convergence characteristics of flower pollination algorithm for circular array synthesis. *Proceedings of the 2nd International Conference on Electronics and Communication Systems*, (pp. 485-489). IEEE.

Chandramouli, K., and Izquierdo, E. 2006. Image classification using chaotic particle swarm optimization. *Proceedings of the 2nd International Conference on Image Processing*, (pp. 3001-3004). IEEE.

Chandrasekar, P., and Krishnamoorthi, M. 2014. BHOHS: A two stage novel algorithm for data clustering. *International Conference on Intelligent Computing Applications*, (pp. 138-142). IEEE.

Chawla, M., and Duhan, M. 2014. Applications of recent metaheuristics optimisation algorithms in biomedical engineering: a review. *International Journal of Biomedical Engineering and Technology*. 16(3), (pp. 268-278).

Chawla, M., and Duhan, M. 2018. Levy flights in metaheuristics optimization algorithms–a review. *Applied Artificial Intelligence*. 32(9-10), (pp. 802-821).

Chechkin, A. V., Metzler, R., Klafter, J., and Gonchar, V. Y. 2008. Introduction to the theory of Lévy flights. *Anomalous transport: Foundations and Applications*. 49(2), (pp. 431-451).

Cheng, M.-Y., and Lien, L.-C. 2012. Hybrid artificial intelligence–based PBA for benchmark functions and facility layout design optimization. *Journal of Computing in Civil Engineering*. 26(5), (pp. 612-624).

Cheng, Y., Jiang, M., and Yuan, D. 2009. Novel clustering algorithms based on improved artificial fish swarm algorithm. *Proceedings of the 6th International Conference on Fuzzy Systems and Knowledge Discovery*, *3* (pp. 141-145). IEEE.

Chu, S.-C., Roddick, J. F., Su, C.-J., and Pan, J.-S. 2004. Constrained ant colony optimization for data clustering. *Proceedings of International Conference on Artificial Intelligence*, (pp. 534-543). Springer.

Chu, S.-C., Tsai, P.-W., and Pan, J.-S. 2006. Cat swarm optimization. *Proceedings of the International Conference on Artificial Intelligence*, (pp. 854-858). Springer.

Clough, E., and Barrett, T. 2016. The gene expression omnibus database. *Statistical Genomics* (pp. 93-110): Springer.

Cohen, B., Sakoda, J., and Bousfield, W. 1954. The statistical analysis of the incidence of clustering in the recall of randomly arranged associates: *Connecticut Univ Storrs.*

Cruz, C., González, J. R., and Pelta, D. A. 2011. Optimization in dynamic environments: a survey on problems, methods and measures. *Soft Computing.* 15(7), (pp. 1427-1448).

Cura, T. 2012. A particle swarm optimization approach to clustering. *Expert Systems with Applications.* 39(1), (pp. 1582-1588).

Daniel, W. W. 1990. Friedman two-way analysis of variance by ranks. *Applied nonparametric Statistics.* (pp. 262-274).

Dash, R., and Misra, B. B. 2018. Performance analysis of clustering techniques over microarray data: A case study. *Physica A: Statistical Mechanics and its Applications.* 493**,** (pp. 162-176).

Davies, D. L., and Bouldin, D. W. 1979. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* (2), (pp. 224-227).

DeRisi, J., Penland, L., Bittner, M., Meltzer, P., Ray, M., Chen, Y., Trent, J. 1996. Use of a cDNA microarray to analyse gene expression. *Nat. genet.* 14, (pp. 457-460).

Dey, A., Bhattacharyya, S., Dey, S., Platos, J., and Snasel, V., 2019. Quantum-inspired bat optimization algorithm for automatic clustering of grayscale images. In Recent Trends in Signal and Image Processing, Springer, (pp. 89-101).

Diaz, J. E., Handl, J., and Xu, D.-L. 2018. Integrating meta-heuristics, simulation and exact techniques for production planning of a failure-prone manufacturing system. *European Journal of Operational Research.* 266(3), (pp. 976-989).

Doraghinejad, M., and Nezamabadi-pour, H. 2014. Black hole: a new operator for gravitational search algorithm. *International Journal of Computational Intelligence Systems.* 7(5), (pp. 809-826).

Dutta, P., Saha, S., and Chauhan, A. B. 2018. Predicting degree of relevance of pathway markers from gene expression data: A PSO based approach. *Proceedings of the International Conference on Neural Information Processing*, (pp. 3-14). Springer.

Eisen, M. B., Spellman, P. T., Brown, P. O., and Botstein, D. 1998. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences.* 95(25), (pp. 14863-14868).

El-Abd, M., Hassan, H., Anis, M., Kamel, M. S., and Elmasry, M. 2010. Discrete cooperative particle swarm optimization for FPGA placement. *Applied Soft Computing.* 10(1), (pp. 284-295).

El-Abd, M., and Kamel, M. S. 2010. A cooperative particle swarm optimizer with migration of heterogeneous probabilistic models. *Swarm Intelligence.* 4(1), (pp. 57-89).

Emami, H., Dami, S., and Shirazi, H. 2015. K-Harmonic means data clustering with imperialist competitive algorithm. *University Politehnica of Bucharest-Scientific Bulletin, Series C: Electrical Engineering and Computer Science.* 77(7). (pp. 9-54).

Emary, E., Zawbaa, H. M., and Sharawi, M. 2019. Impact of Lèvy flight on modern meta-heuristic optimizers. *Applied Soft Computing.* 75, (pp. 775-789).

Erol, O. K., and Eksin, I. 2006. A new optimization method: big bang–big crunch. *Advances in Engineering Software.* 37(2), (pp. 106-111).

Eskandarzadehalamdary, M., Masoumi, B., and Sojodishijani, O. 2014. A new hybrid algorithm based on black hole optimization and bisecting k-means for cluster analysis. *Proceedings of the 22nd Iranian Conference on Electrical Engineering* (pp. 1075-1079). IEEE.

Esmin, A. A., Coelho, R. A., and Matwin, S. 2015. A review on particle swarm optimization algorithm and its variants to clustering high-dimensional data. *Artificial Intelligence Review.* **44**(1), (pp. 23-45).

Farah, A., and Belazi, A. 2018. A novel chaotic Jaya algorithm for unconstrained numerical optimization. *Nonlinear Dynamics.* **93**(3), (pp. 1451-1480).

Fehrmann, R. S., Karjalainen, J. M., Krajewska, M., Westra, H.-J., Maloney, D., Simeonov, A., Schultes, E. A. 2015. Gene expression analysis identifies global gene dosage sensitivity in cancer. *Nature Genetics.* **47**(2), (pp.103-115).

Fister, I., Fister Jr, I., Yang, X.-S., and Brest, J. 2013. A comprehensive review of firefly algorithms. *Swarm and Evolutionary Computation.* 13, (pp. 34-46).

Gao, W., Ge, M., Chen, D., and Wang, X. 2016a. Back analysis for rock model surrounding underground roadways in coal mine based on black hole algorithm. *Engineering with Computers.* 32(4), (pp. 675-689).

Gao, W., Wang, X., Dai, S., and Chen, D. 2016b. Study on stability of high embankment slope based on black hole algorithm. *Environmental Earth Sciences.* 75(20), (pp. 1381-1373).

García, J., Crawford, B., Soto, R., and Astorga, G. 2019. A clustering algorithm applied to the binarization of swarm intelligence continuous metaheuristics. *Swarm and Evolutionary Computation.* 44, (pp. 646-664).

García, J., Crawford, B., Soto, R., and García, P. 2017. A multi dynamic binary black hole algorithm applied to set covering problem. *Proceedings of the International Conference on Harmony Search Algorithm*, (pp. 42-51). IEEE.

Geem, Z. W., Kim, J. H., and Loganathan, G. V. 2001. A new heuristic optimization algorithm: harmony search. *Simulation.* 76(2), (pp. 60-68).

Ghafarzadeh, H., and Bouyer, A. 2016. An Efficient Hybrid Clustering Method Using an Artificial Bee Colony Algorithm and Mantegna Lévy Distribution. *International Journal on Artificial Intelligence Tools.* 25(02), (pp.155-194).

Giacconi, R. 2001. Black hole research past and future. *Black Holes in Binaries and Galactic Nuclei: Diagnostics, Demography and Formation*. (pp. 3-15). Springer.

Goel, S., Sharma, A., and Bedi, P. 2011. Cuckoo Search Clustering Algorithm: A novel strategy of biomimicry. *Proceedings of the Congress of Information and Communication Technologies*, (pp. 916-921). IEEE.

Goldberg, D. E., and Deb, K. 1991. A comparative analysis of selection schemes used in genetic algorithms. *In Foundations of Genetic Algorithms* (pp. 69-93). Elsevier.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Caligiuri, M. A. 1999. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science.* 286(5439), (pp. 531-537).

Guo, Y.-n., Liu, D., and Cheng, J. 2011. Multi-population cooperative cultural algorithms. *Proceedings of the International Conference on Intelligent Computing*, 51, (pp. 199-206). Springer.

Gupta, G. P., and Jha, S. 2018. Integrated clustering and routing protocol for wireless sensor networks using Cuckoo and Harmony Search based metaheuristic techniques. *Engineering Applications of Artificial Intelligence.* 68, (pp. 101-109).

Haklı, H., and Uğuz, H. 2014. A novel particle swarm optimization algorithm with Levy flight. *Applied Soft Computing.* 23, (pp. 333-345).

Handl, J., Knowles, J., and Dorigo, M. 2006. Ant-based clustering and topographic mapping. *Artificial life.* 12(1), (pp. 35-62).

Hassanzadeh, T., and Meybodi, M. R. 2012. A new hybrid approach for data clustering using firefly algorithm and K-means. *Proceedings of the International Symposium on Artificial Intelligence and Signal Processing*, (pp. 007-011). IEEE.

Hatamlou, A. 2013. Black hole: A new heuristic optimization approach for data clustering. *Information Sciences.* 222**,** (pp. 175-184).

Hatamlou, A. 2014. Heart: a novel optimization algorithm for cluster analysis. *Progress in Artificial Intelligence.* 2(2-3), (pp. 167-173).

Hatamlou, A., Abdullah, S., and Hatamlou, M. 2011a. Data clustering using big bang–big crunch algorithm. *Proceedings of the International Conference on Innovative Computing Technology* (pp. 383-388): Springer.

Hatamlou, A., Abdullah, S., and Nezamabadi-Pour, H. 2011b. Application of gravitational search algorithm on data clustering. *Proceedings of the International Conference on Rough Sets and Knowledge Technology*, (pp. 337-346). Springer.

Hatamlou, A., Abdullah, S., and Othman, Z. 2011c. Gravitational search algorithm with heuristic search for clustering problems. *Proceedings of the 3rd Conference on Data Mining and Optimization*, (pp. 190-193). IEEE.

Hatamlou, A., and Hatamlou, M. 2013. PSOHS: An efficient two-stage approach for data clustering. *Memetic Computing.* **5**(2), (pp. 155-161).

Heidari, A., and Abbaspour, R. 2014. A gravitational black hole algorithm for autonomous UCAV mission planning in 3D realistic environments. *International Journal of Computer Applications.* 95(9) (pp. 160-179).

Hongwei, Z., Xiaoke, C., and Shurong, Z. 2010. Fuzzy multi-population cooperative GA and MOT optimization. *Proceedings of the 3rd International Conference on Advanced Computer Theory and Engineering, V1* (pp.669 -680). IEEE.

Hussain, K., Salleh, M. N. M., Cheng, S., and Shi, Y. 2018. Metaheuristic research: a comprehensive survey. *Artificial Intelligence Review.* (pp. 1-43).

Ilango, S. S., Vimal, S., Kaliappan, M., and Subbulakshmi, P. 2018. Optimization using artificial bee colony based clustering approach for big data. *Cluster Computing.* (pp. 1-9).

Jaddi, N. S., Alvankarian, J., and Abdullah, S. 2017. Kidney-inspired algorithm for optimization problems. *Communications Proceedings of the Nonlinear Science and Numerical Simulation.* 42, (pp. 358-369).

Jadidoleslamy, H. 2014. A novel clustering algorithm for homogenous and large-scale wireless sensor networks: based on sensor nodes deployment location coordinates. ***I****nternational Journal of Computer Science and Network Security*. 14(2), (pp. 89-97).

Jaiprakash, K. P., and Nanda, S. J. 2019. Elephant herding algorithm for clustering. *In Recent Developments in Machine Learning and Data Analytics* (pp. 317-325): Springer.

Janardhanan, M. N., Li, Z., Bocewicz, G., Banaszak, Z., and Nielsen, P. 2019. Metaheuristic algorithms for balancing robotic assembly lines with sequence-dependent robot setup times. *Applied Mathematical Modelling*. 65, (pp. 256-270).

Jensi, R., and Jiji, G. W. 2015. MBA-LF: A new data clustering method using modified bat algorithm and levy flight. *Journal on Soft Computing*. 6(1) (pp. 150-173).

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

Ji, J., Pang, W., Zheng, Y., Wang, Z., and Ma, Z. 2015. A novel artificial bee colony based clustering algorithm for categorical data. *PloS One*. 10(5), (pp. 121-125).

Jiang, D., Tang, C., and Zhang, A. 2004. Cluster analysis for gene expression data: A survey. *IEEE Transactions on Knowledge & Data Engineering*. (11), (pp. 1370-1386).

Joshua, C. J., Duraisamy, R., and Varadarajan, V. 2019. A reputation based weighted clustering protocol in VANET: a multi-objective firefly approach. *Mobile networks and applications*. (pp. 1-11).

Jourdan, L., Basseur, M., and Talbi, E.-G. 2009. Hybridizing exact methods and metaheuristics: A taxonomy. *European Journal of Operational Research*. 199(3), (pp. 620-629).

Karaboga, D., and Basturk, B. 2007. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of Global Optimization*. 39(3), (pp. 459-471).

Kartous, W., Layeb, A., & Chikhi, S. 2014. A new quantum cuckoo search algorithm for multiple sequence alignment. Journal of Intelligent Systems, 23(3), (pp. 261-275).

Karaboga, D., and Ozturk, C. 2011. A novel clustering approach: Artificial Bee Colony (ABC) algorithm. *Applied Soft Computing*. 11(1), (pp. 652-657).

Kaushik, K., and Arora, V. 2015. A hybrid data clustering using firefly algorithm based improved genetic algorithm. *Procedia Computer Science.* 58, (pp. 249-256). Elsevier

Kennedy, J., and Eberhart, R. 1995. Particle swarm optimization (PSO). *Proceedings of the International Conference on Neural Networks.* (pp. 1942-1948). IEEE.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. 1983. Optimization by simulated annealing. *Science.* **220**(4598), (pp. 671-680).

Kowalski, P. A., Łukasik, S., Charytanowicz, M., and Kulczycki, P. 2019. Nature Inspired Clustering–Use Cases of Krill Herd Algorithm and Flower Pollination Algorithm. *Interactions Between Computational Intelligence and Mathematics.* (pp. 83-98): Springer.

Kumar, G. N., Rao, B. V., Chowdary, D. D., and Sobhan, P. V. 2018. Multi-objective optimal power flow using metaheuristic optimization algorithms with unified power flow controller to enhance the power system performance. *Advancements in Applied Metaheuristic Computing* (pp. 1-33). IGI Global.

Kumar, S., Datta, D., and Singh, S. K. 2015. Black hole algorithm and its applications. *Computational Intelligence Applications in Modeling and Control.* (pp. 147-170). Springer.

Kumar, V., Chhabra, J. K., and Kumar, D. 2017. Grey wolf algorithm-based clustering technique. *Journal of Intelligent Systems.* 26(1), (pp. 153-168).

Lakshmi, K., Visalakshi, N. K., and Shanthi, S. 2018. Data clustering using K-Means based on Crow Search Algorithm. *Sādhanā.* 43(11), (pp. 185-190).

Li, Q., and Pei, Z. 2015. The black hole clustering algorithm based on membrane computing. *Proceedings of the International Symposium on Computers & Informatics.* Atlantis Press.

Liu, Y., Yi, Z., Wu, H., Ye, M., and Chen, K. 2008. A tabu search approach for the minimum sum-of-squares clustering problem. *Information Sciences.* 178(12), (pp. 2680-2704).

Łukasik, S., and Żak, S. 2009. Firefly algorithm for continuous constrained optimization tasks. *Proceedings of the International Conference on Computational Collective Intelligence*, (pp. 97-106). Springer.

Mageshkumar, C., Karthik, S., and Arunachalam, V. 2019. Hybrid metaheuristic algorithm for improving the efficiency of data clustering. *Cluster Computing.*(pp. 1-8).

Mahdavi, M., and Abolhassani, H. 2009. Harmony K-means algorithm for document clustering. *Data Mining and Knowledge Discovery.* 18(3), (pp. 370-391).

Marichelvam, M. K., Prabaharan, T., and Yang, X. S. 2013. A discrete firefly algorithm for the multi-objective hybrid flowshop scheduling problems. *IEEE Transactions on Evolutionary Computation.* 18(2), (pp. 301-305).

Maulik, U., and Bandyopadhyay, S. 2000. Genetic algorithm-based clustering technique. *Pattern recognition.* 33(9), (pp. 1455-1465).

McDowell, I. C., Manandhar, D., Vockley, C. M., Schmid, A. K., Reddy, T. E., and Engelhardt, B. E. 2018. Clustering gene expression time series data using an infinite Gaussian process mixture model. *PLoS Cmputational Biology.* 14(1), (pp. 5896-5903).

Mirjalili, S. 2016. Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Computing and Applications.* 27(4), (pp. 1053-1073).

Mirjalili, S., Mirjalili, S. M., and Hatamlou, A. 2016. Multi-verse optimizer: a nature-inspired algorithm for global optimization. *Neural Computing and Applications.* 27(2), (pp. 495-513).

Mirjalili, S., Mirjalili, S. M., and Lewis, A. 2014. Grey wolf optimizer. *Advances in Engineering Software.* 69, (pp. 46-61).

Mohammed, S. K., Ibrahim, Z., Daniyal, H., and Aziz, N. A. A. 2016. A new hybrid gravitational search–black hole algorithm. *Proceedings of the The National Conference for Postgraduate Research*, (pp. 834-842).

Moscato, P. 1989. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. *Caltech Concurrent Computation Program, C3P Report.* (pp. 826-834).

Munoz, R., Olivares, R., Taramasco, C., Villarroel, R., Soto, R., Barcelos, T. S., Alonso-Sánchez, M. F. 2018. Using black hole algorithm to improve EEG-based emotion recognition. *Computational Intelligence and Neuroscience.*

Nayak, S. K., Rout, P. K., and Jagadev, A. K. 2019. Multi-objective clustering: a kernel based approach using differential evolution. *Connection Science.* (pp. 1-28).

Nerurkar, P., Pavate, A., Shah, M., and Jacob, S. 2019. Performance of internal cluster validations measures for evolutionary clustering *computing, Communication and Signal Processing* (pp. 305-312). Springer.

Neshat, M., Yazdi, S. F., Yazdani, D., and Sargolzaei, M. 2012. A new cooperative algorithm based on PSO and k-means for data clustering. *Journal of Computer Science.* 8(2), (pp. 188.201).

Ngenkaew, W., Ono, S., and Nakayama, S. 2008. Pheromone-based concept in Ant Clustering. *Proceedings of the 3rd International Conference on Intelligent System and Knowledge Engineering.* (pp. 308-312). IEEE.

Niu, B., Zhu, Y., and He, X. 2005. Multi-population cooperative particle swarm optimization. *European Conference on Artificial Life*, (pp. 874-883).

Niu, B., Zhu, Y., He, X., and Wu, H. 2007. MCPSO: A multi-swarm cooperative particle swarm optimizer. *Applied Mathematics and Computation.* 185(2), (pp. 1050-1062). Springer.

Olivares, R., Soto, R., Crawford, B., Barría, M., and Niklander, S. 2016. Evaluation of choice functions to self-adaptive on constraint programming via the black hole algorithm. *Proceedings of the XLII Latin American Computing Conference*, (pp. 1-8). IEEE.

Pal, N. R., and Bezdek, J. C. 1995. On cluster validity for the fuzzy c-means model. *IEEE Transactions on Fuzzy Systems.* 3(3), (pp. 370-379).

Pashaei, E., and Aydin, N. 2017. Binary black hole algorithm for feature selection and classification on biological data. *Applied Soft Computing.* 56**,** (pp. 94-106).

Pickover, C. A. 1998. Black holes: a traveler's guide. (pp. 244-261).

Piotrowski, A. P., Napiorkowski, J. J., and Rowinski, P. M. 2014. How novel is the "novel" black hole optimization approach? *Information Sciences.* 267**,** (pp. 191-200).

Pomeroy, S. L., Tamayo, P., Gaasenbeek, M., Sturla, L. M., Angelo, M., McLaughlin, M. E., Lau, C. 2002. Prediction of central nervous system embryonal tumour outcome based on gene expression. *Nature.* 415(6870), (pp. 436-443).

Potter, M. A., and De Jong, K. A. 1994. A cooperative coevolutionary approach to function optimization. *Proceedings of the International Conference on Parallel Problem Solving from Nature*, (pp. 249-257). Springer.

Prakash, J., and Singh, P. K. 2019. Gravitational search algorithm and K-means for simultaneous feature selection and data clustering: a multi-objective approach. *Soft Computing.* 23(6), (pp. 2083-2100).

Premalatha, K., and Balamurugan, R. 2015. A nature inspired swarm based stellar-mass black hole for engineering optimization. *Proceedings of the International Conference on Electrical, Computer and Communication Technologies*, (pp. 1-8). IEEE.

Rajabioun, R. 2011. Cuckoo optimization algorithm. *Applied Soft Computing.* 11(8), (pp. 5508-5518).

Ramadas, M., and Abraham, A. 2019. Metaheuristics and data clustering. *Metaheuristics for Data Clustering and Image Segmentation* (pp. 7-55): Springer.

Rana, S., Jasola, S., and Kumar, R. 2011. A review on particle swarm optimization algorithms and their applications to data clustering. *Artificial Intelligence Review*. 35(3), (pp. 211-222).

Rashedi, E., Nezamabadi-Pour, H., and Saryazdi, S. 2009. GSA: a gravitational search algorithm. *Information Sciences*. 179(13), (pp. 2232-2248).

Rehman, S., Ali, S., and Khan, S. A. 2016. Wind farm layout design using cuckoo search algorithms. *Applied artificial intelligence*. 30(10), (pp. 899-922).

Reynolds, A. M., and Rhodes, C. J. 2009. The Lévy flight paradigm: random search patterns and mechanisms. *Ecology*. 90(4), (pp. 877-887).

Richer, T. J., and Blackwell, T. M. 2006. The Lévy particle swarm. *Proceedings of the International Conference on Evolutionary Computation*, (pp.808-815). IEEE.

Rubio, Á. G., Crawford, B., Soto, R., Jaramillo, A., Villablanca, S. M., Salas, J., and Olguín, E. 2016. An Binary Black Hole Algorithm to Solve Set Covering Problem. *Proceedings of the International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, (pp. 873-883). Springer.

Saida, I. B., Nadjet, K., and Omar, B. 2014. A new algorithm for data clustering based on cuckoo search optimization. *Genetic and Evolutionary Computing* (pp. 55-64): Springer.

Santosa, B., and Ningrum, M. K. 2009. Cat swarm optimization for clustering. *Proceedings of the International Conference of Soft Computing and Pattern Recognition*, (pp. 54-59). IEEE.

Sarstedt, M., and Mooi, E. 2019. Cluster analysis. *A concise guide to market research* (pp. 301-354). Springer.

Schutz, B. 2003. *Gravity from the ground up: An introductory guide to gravity and general relativity*. Cambridge University Press.

Senthilnath, J., Das, V., Omkar, S., and Mani, V. 2013. Clustering using levy flight cuckoo search. *Proceedings of the 7th International Conference on Bio-Inspired Computing: Theories and Applications*, (pp. 65-75). Springer.

Senthilnath, J., Kulkarni, S., Suresh, S., Yang, X., and Benediktsson, J. 2019. FPA clust: evaluation of the flower pollination algorithm for data clustering. *Evolutionary Intelligence. (pp.* 1-11).

Senthilnath, J., Omkar, S., and Mani, V. 2011. Clustering using firefly algorithm: performance study. *Swarm and Evolutionary Computation*. 1(3), (pp. 164-171).

Sharma, F. B., and Kapoor, S. R. 2017. Induction motor parameter estimation using disrupted black hole artificial bee colony algorithm. *International Journal of Metaheuristics.* 6(1-2), (pp. 85-106).

Sharma, P., Sharma, H., Kumar, S., and Sharma, K. 2019. black-hole gbest differential evolution algorithm for solving robot path planning problem. *Harmony Search and Nature Inspired Optimization Algorithms* (pp. 1009-1022). Springer.

Shelokar, P., Jayaraman, V. K., and Kulkarni, B. D. 2004. An ant colony approach for clustering. *Analytica Chimica Acta.* 509(2), (pp. 187-195). Elsevier.

Singh, V., and Sood, M. M. 2013. Krill Herd clustering algorithm using dbscan technique. *Internatunal Journal of Computing.* 4(03), (pp. 197-200).

Sivaraman, N., Mohan, S., and Arunkumar, R. 2019. A grey wolf optimization algorithm for clustering and networking. *Journal of Computational and Theoretical Nanoscience.* 16(4), (pp. 1360-1364).

Škrjanc, I., Ozawa, S., Ban, T., and Dovžan, D. 2018. Large-scale cyber attacks monitoring using evolving cauchy possibilistic clustering. *Applied Soft Computing.* 62, (pp. 592-601).

Soto, R., Crawford, B., Figueroa, I., Niklander, S., and Olguín, E. 2016. A black hole algorithm for solving the set covering problem. *Proceedings of the International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, (pp. 855-861). Springer.

Soto, R., Crawford, B., Olivares, R., Barraza, J., Figueroa, I., Johnson, F., Olguín, E. 2017a. Solving the non-unicost set covering problem by using cuckoo search and black hole optimization. *Natural Computing.* 16(2), (pp. 213-229).

Soto, R., Crawford, B., Olivares, R., Niklander, S., Johnson, F., Paredes, F., and Olguín, E. 2017b. Online control of enumeration strategies via bat algorithm and black hole optimization. *Natural Computing.* 16(2), (pp. 241-257).

Spellman, P. T., Sherlock, G., Zhang, M. Q., Iyer, V. R., Anders, K., Eisen, M. B., Futcher, B. 1998. Comprehensive identification of cell cycle–regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. *Molecular Biology of the Cell.* 9(12), (pp. 3273-3297).

Suad Khairi, M., Zuwairie, I., Hamdan, D., Azlina, N., and Aziz, A. 2016. White hole-black hole algorithm. *Proceedings of the National Conference for Postgraduate Research.* (pp. 824-833).

Sun, J., Chen, W., Fang, W., Wun, X., and Xu, W. 2012. Gene expression data analysis with the clustering method based on an improved quantum-behaved Particle Swarm Optimization. *Engineering Applications of Artificial Intelligence.* 25(2), (pp. 376-391).

Sun, J., Xu, W., and Ye, B. 2006. Quantum-behaved particle swarm optimization clustering algorithm. *Advanced Data Mining and Applications.* (pp**.** 340-347). Springer.

Sun, S., and Peng, Q. 2014. A hybrid PSO-GSA strategy for high-dimensional optimization and microarray data clustering. *Proceedings of the International Conference on Information and Automation* (pp. 41-46). IEEE.

Talbi, E.-G. 2009. *Metaheuristics: from design to implementation*. *Vol. 74*. John Wiley & Sons.

Tang, R., Fong, S., Yang, X.-S., and Deb, S. 2012. Integrating nature-inspired optimization algorithms to K-means clustering. *Proceedings of the 7th International Conference on Digital Information Management*. (pp. 116-123). IEEE.

Tsai, C.-W., Hsieh, C.-H., and Chiang, M.-C. 2015. Parallel black hole clustering based on MapReduce. *Proceedings of the International Conference on Systems, Man, and Cybernetics* (pp. 2543-2548). IEEE.

Van't Veer, L. J., Dai, H., Van De Vijver, M. J., He, Y. D., Hart, A. A., Mao, M., Witteveen, A. T. 2002. Gene expression profiling predicts clinical outcome of breast cancer. *Nature.* 415(6871), (pp. 521-530).

Van den Bergh, F., and Engelbrecht, A. P. 2004. A cooperative approach to particle swarm optimization. *IEEE Transactions on Evolutionary Computation.* 8(3), (pp. 225-239).

Van der Merwe, D., and Engelbrecht, A. P. 2003b. Data clustering using particle swarm optimization. *Proceedings of the Congress on Evolutionary Computation,* (pp. 215-220). IEEE.

Wan, Y., Pei, T., Zhou, C., Jiang, Y., Qu, C., and Qiao, Y. 2012. ACOMCD: A multiple cluster detection algorithm based on the spatial scan statistic and ant colony optimization. *Computational Statistics & Data Analysis*. 56(2), (pp. 283-296).

Wang, G.-G., Chang, B., and Zhang, Z. 2015. A multi-swarm bat algorithm for global optimization. *Proceedings of the Congress on Evolutionary Computation***,** (pp. 480-485). IEEE.

Wen, X., Fuhrman, S., Michaels, G. S., Carr, D. B., Smith, S., Barker, J. L., and Somogyi, R. 1998. Large-scale temporal gene expression mapping of central nervous system development. *Proceedings of the National Academy of Sciences.* **95**(1), (pp. 334-339).

Wilcoxon, F. 1945. Individual comparisons by ranking methods. *Biometrics bulletin.* 1(6), (pp. 80-83).

Xiao, X., Dow, E. R., Eberhart, R., Miled, Z. B., and Oppelt, R. J. 2003. Gene clustering using self-organizing maps and particle swarm optimization. *Proceedings of the International Symposium of Parallel and Distributed Processing* (pp.10-23). IEEE.

Yaghoobi, S., and Mojallali, H. 2016. Modified black hole algorithm with genetic operators. *International Journal of Computational Intelligence Systems*. 9(4), (pp. 652-665).

Yan, X., Zhu, Y., Zou, W., and Wang, L. 2012. A new approach for data clustering using hybrid artificial bee colony algorithm. *Neurocomputing*. 97, (pp. 241-250).

Yang, F., Sun, T., and Zhang, C. 2009. An efficient hybrid data clustering method based on K-harmonic means and particle swarm optimization. *Expert Systems with Applications*. 36(6), (pp. 9847-9852).

Yang, S., and Li, C. 2010. A clustering particle swarm optimizer for locating and tracking multiple optima in dynamic environments. *IEEE Transactions on Evolutionary Computation*. 14(6), (pp. 959-974).

Yang, S., Ong, Y.-S., and Jin, Y. 2007. *Evolutionary computation in dynamic and uncertain environments*. *Vol. 51* (pp. 167-173) Springer Science & Business Media.

Yang, X.-S. 2009. Firefly algorithms for multimodal optimization. *Proceedings of the International Symposium on Stochastic Algorithms*, (pp. 169-178). Springer.

Yang, X.-S. 2010a. Firefly algorithm, Levy flights and global optimization. *Research and development in Intelligent Systems*. (pp. 209-218): Springer.

Yang, X.-S. 2010b. Firefly algorithm, stochastic test functions and design optimisation. *arXiv preprint arXiv:(pp. 1003-1409.)*

Yang, X.-S. 2010d. A new metaheuristic bat-inspired algorithm. *Nature inspired cooperative strategies for optimization*. (pp. 65-74). Springer.

Yang, X.-S. 2012. Flower pollination algorithm for global optimization. *Proceedings of the International Conference on Unconventional Computing and Natural Computation*, (pp. 240-249). Springer.

Yang, X.-S. 2013. Bat algorithm and cuckoo search: a tutorial. *In Artificial Intelligence, Evolutionary Computing and Metaheuristics*. (pp. 421-434): Springer.

Yang, X.-S., and Deb, S. 2009. Cuckoo search via Lévy flights. *Proceedings of the Congress on Nature & Biologically Inspired Computing*. (pp. 210-214). IEEE.

Yang, X.-S., and Deb, S. 2010. Eagle strategy using Lévy walk and firefly algorithms for stochastic optimization. *Nature Inspired Cooperative Strategies for Optimization*. (pp. 101-111): Springer.

Yang, X.-S., and Deb, S. 2013. Multiobjective cuckoo search for design optimization. *Computers & Operations Research.* 40(6), (pp. 1616-1624).

Yang, X.-S., Deb, S., Fong, S., He, X., and Zhao, Y.-X. 2016. From swarm intelligence to metaheuristics: nature-inspired optimization algorithms. *Computer.* 49(9), (pp. 52-59).

Yang, X.-S., Deb, S., and He, X. 2013a. Eagle strategy with flower algorithm. *Proceedings of the International Conference on Advances in Computing, Communications and Informatics*, (pp. 1213-1217). IEEE.

Yang, X.-S., Karamanoglu, M., and He, X. 2013b. Multi-objective flower algorithm for optimization. *Procedia Computer Science.* 18, (pp. 861-868).

Yildiz, A. R. 2012. A comparative study of population-based optimization algorithms for turning operations. *Information Sciences.* 210 (pp. 81-88).

Yildiz, A. R. 2013a. Comparison of evolutionary-based optimization algorithms for structural design optimization. *Engineering Applications of Artificial Intelligence.* 26(1), (pp. 327-333).

Yildiz, A. R. 2013b. Optimization of cutting parameters in multi-pass turning using artificial bee colony-based approach. *Information Sciences.* 220, (pp. 399-407).

Yin, M., Hu, Y., Yang, F., Li, X., and Gu, W. 2011. A novel hybrid K-harmonic means and gravitational search algorithm approach for clustering. *Expert Systems with Applications.* 38(8), (pp. 9319-9324).

Zabihi, F., and Nasiri, B. 2018. A novel history-driven artificial bee colony algorithm for data clustering. *Applied Soft Computing.* 71, (pp. 226-241).

Zambrano-Bigiarini, M., Clerc, M., and Rojas, R. 2013. Standard particle swarm optimisation 2011 at cec-2013: A baseline for future pso improvements. *Proceedings of the Congress on Evolutionary Computation*, (pp. 2337-2344). IEEE.

Zamli, K. Z., Alkazemi, B. Y., and Kendall, G. 2016. A tabu search hyper-heuristic strategy for t-way test suite generation. *Applied soft computing.* 44, (pp. 57-74).

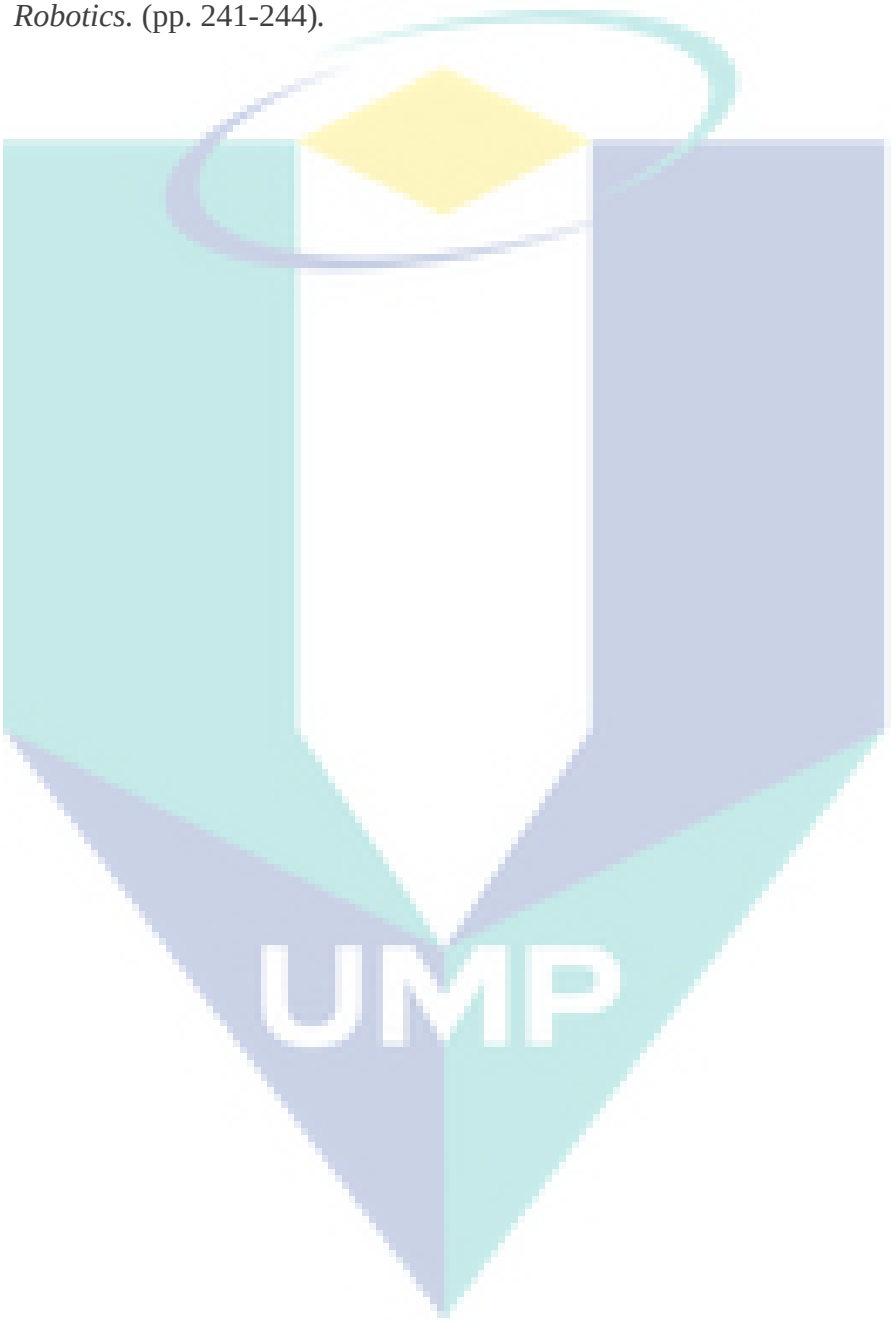
Zhang, C., Ouyang, D., and Ning, J. 2010. An artificial bee colony approach for clustering. *Expert Systems with Applications.* 37(7), (pp. 4761-4767).

Zhang, L., and Cao, Q. 2011. A novel ant-based clustering algorithm using the kernel method. *Information sciences.* 181(20), (pp. 4658-4672).

Zhang, L., Shan, L., and Wang, J. 2017. Optimal feature selection using distance-based discrete firefly algorithm with mutual information criterion. *Neural Computing and Applications.* 28(9), (pp. 2795-2808).

Zhou, Y., Wu, H., Luo, Q., and Abdel-Baset, M. 2019. Automatic data clustering using nature-inspired symbiotic organism search algorithm. *Knowledge-Based Systems.* 163**, (pp.** 546-557).

Zuwairie, I., Suad Khairi, M., Norazian, S., Azlina, N., Aziz, A., Hidayati, N., Zulkifli, M. 2018. A review on fundamental advancements of black hole algorithm. *Proceedings of the International Conference on Artificial Intellegence and Robotics.* (pp. 241-244).



APPENDIX A
LIST OF PUBLICATIONS

Haneen A. A, AbdulRahman A. AlSewari, Noraziah Ahmed, Sinan Q. Salih, 2019 "An Enhanced Version of Black Hole Algorithm Via Levy Flight for Optimization and Data Clustering Problems", IEEE Access. ISI Q1, IF: 4.098.

Haneen A. A, Noraziah Ahmed, Ritu Gupta 2018, "Review on Data Partitioning Strategies in Big Data Environment" in the Journal of physics 1018(1),012019.

Haneen A. A, Noraziah Ahmed, Ritu Gupta, Fakherldin, M.A.I. 2017, "Review on data partitioning strategies in big data environment" Journal of Advanced Science Letters, American Scientific Publishers, 23(11), pp.11101-11104.