A

Major Project-II Report

On

A new approach for data clustering using Multi-verse optimizer algorithm

A dissertation submitted in the partial fulfillment for the award of Degree of

Master of Technology

In

Software Technology

by

Manvendra Pratap Singh (2K14/SWE/11)

Under the guidance of

Prof (Dr.) Kapil Sharma

Associate professor

Department of Computer Engineering, DTU

DEPARTMENT OF SOFTWARE ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

BAWANA ROAD, DELHI

DECLARATION

I hereby want to declare that the thesis entitled "**A new approach for data clustering using Multi-verse optimizer algorithm"** which is being submitted to the **Delhi Technological University**, in partial fulfillment of the requirements for the award of degree in **Master of Technology in Software Technology** is an authentic work carried out by me. The material contained in this thesis has not been submitted to any institution or university for the award of any degree.

_____

Manvendra Pratap Singh

Department of Computer Engineering

Delhi Technological University,

Delhi.

CERTIFICATE

Delhi Technological University
(Government of Delhi NCR)
Bhawana Road, New Delhi-42

This is to certify that the thesis entitled **"A new approach for data clustering using Multi-verse optimizer algorithm"** done by **Manvendra Pratap Singh** (2K14/SWE/) for the partial fulfillment of the requirements for the award of degree of **Master of Technology** Degree in **Software Technology** in the **Department of Computer Engineering**, Delhi Technological University, New Delhi is an authentic work carried out by him under my guidance.

Project Guide:
Dr. Kapil Sharma
Associate Professor
Department of Computer Engineering
Delhi Technological University, Delhi

ACKNOWLEDGEMENT

I take this opportunity to express my deep sense of gratitude and respect towards my guide **Dr. Kapil Sharma, Associate professor of Department of Computer Engineering.**

I am very much indebted to him for his generosity, expertise and guidance I have received from him while working on this project. Without his support and timely guidance the completion of the project would have seemed a far –fetched dream. In this respect I find myself lucky to have my guide. He have guided not only with the subject matter, but also taught the proper style and techniques of documentation and presentation. I would also like to take this opportunity to present my sincere regards to **Mr. Ashish Tripathi**, Research Scholar, DTU for extending his support and valuable Guidance.

Besides my guides, I would like to thank entire teaching and non-teaching staff in the Department of Computer Engineering, DTU for all their help during my tenure at DTU. Kudos to all my friends at DTU for thought provoking discussion and making stay very pleasant.

Manvendra Pratap Singh
M.Tech Software Engineering
2K14/SWE/11

Abstract

Nature has always been a source of inspiration. Over the last few decades, it has stimulated many successful algorithms and computational tools for dealing with complex and optimization problems. This work proposes a new heuristic algorithm that is inspired by the Universe theory of multi-verse i.e. more than one Universe phenomenon. Similar to other population-based algorithms, the Multi-verse optimizer (MVO) starts with an initial population of candidate solutions to an optimization problem and an objective function that is calculated for them. At each iteration of the MVO, the best candidate is selected to be the Best Universe, which then starts exchanging the objects from other Universe. Also the Universes with high inflation rate move their objects to the universe having low inflation rate in order to make abrupt changes. To evaluate the performance of the MVO algorithm, it is applied to solve the clustering problem, which is a NP-hard problem. The experimental results show that the proposed MVO clustering algorithm outperforms other traditional heuristic algorithms for five benchmark datasets.

Contents

List of Figures

Chapter One: INTRODUCTION

1.1 Clustering

Clustering is a technique by which set of heterogeneous objects are to be grouped into different sets. These sets are develop on the basis of similarity with in the objects. Most similar objects are to be grouped in same set. In today scenario clustering is used in many fields like data mining [1], pattern recognition, information retrieval, feature extraction, image segmentation [2] and many more. Clustering analysis is not an algorithm, but it is a task that can be done by using various algorithms like K-mean algorithm, fuzzy c-mean, density based algorithm and meta-heuristic based clustering. Basically there are two ways to do the clustering analysis, one is supervised clustering technique and another is unsupervised clustering technique. In unsupervised clustering we don't have any prior knowledge about data vectors or we say that teacher does not exists to train the test data set. In this technique we only know the number of cluster to be formed. Data vector to be grouped on the basis of some similarity metrics like distance based metrics or density based metrics. Most basic unsupervised clustering used is k-mean clustering [3, 4]. In k-mean clustering data vectors are grouped into predefined number of cluster and cluster are formed on the basis of distance similarity measure i.e. Mean squared distance between data vectors and cluster centroid. Objects with in same cluster have small Euclidean distance [5] from their cluster centroid as compare to other objects. On the other hand supervised clustering consist of training data set. These training data set drives the test data set to form the cluster centroid. This technique consist of external teacher which classify the test data set to its cluster. Main objective of all clustering algorithm is that minimizing the Intra-cluster distance or maximizing the Inter-cluster distance. Doing clustering on data set have advantages like one can easily analyze the data set when they are divided into their respective classes or cluster. By categorizing the data set to classes, a group based decision can be taken which is very easy as compare to object wise decision generation. We can learn group based characteristics of the data set by dividing them in classes.

1.2 Basic Framework for Data Clustering

Traditional clustering algorithm consist of their own similarity metrics based on their nature. Some algorithms group the data set on the basis of density regions like DBSCAN and OPTICS, on the basis of distance like k-mean, on the basis of connectivity like hierarchical clustering. Over the period of time they all iterate to refine the solution towards Best cluster centroids. Now heuristic based algorithm are used for data clustering. They consist of search agents which represents candidate solution to the clustering problem. These candidate solutions are initialize by randomly 'k' cluster centroid in d-dimensional search space. Each heuristic algorithm move the cluster centroid on the basis of their respective movement of candidate solution in order to achieve optimal results. This movement of cluster centroid carries on to fixed number of iteration to obtain optimal cluster centroid.

1.3 Swarm based optimization algorithms

The main motivation of the swarm based optimization algorithm is the natural phenomenon. Natural phenomenon are adopted for optimization purpose for meta-heuristic algorithms. Nature Inspired algorithms initially seeded with the random population in problem search space. These population are evolve, combine and move over the fixed number of iterations to find the best solution. It is the main framework of all the nature inspired optimization algorithm. These algorithm only differ by the movement or the evolution of its population for obtaining optimize results. For example there are algorithms like GA [6] (genetic algorithm) in which the concept of survival of the fittest is adopted by GA to find the best solution. Population are operated by two operators: mutation and crossover to evolve or move the population. PSO [7] (Particle swarm optimization) was inspired by individual thinking and social behavior of particles (bird) to move the swarm in search space. Every particle on the basis of its neighbor interaction and its local best position move to obtain global best position of swarm. All the nature based algorithm consist of two main concepts:

- ✓ **Exploration:** Exploration is the process of finding the promising areas for the optimization problem. These areas consist of potential solutions.

- ✓ **Exploitation:** Exploitation is used to convergence of the solution to the promising area find in the Exploration phase.

Proper balance has to maintain between exploration and exploitation phase so that solution does not trap in local optima and solution will be obtain in optimize time. So a proper transition is used to move the candidate solution toward global optima. Nowadays meta-heuristic algorithms are widely in many optimization problem. Many research are going on to proposing nature inspired optimization problems and modifying previous heuristic algorithm in order to make them efficient.

1.4 Motivation

Data analysis underlies many computing applications, either in a design phase or as part of their on-line operations. Data analysis procedures can be dichotomized as either exploratory or confirmatory, based on the availability of appropriate models for the data source, but a key element in both types of procedures (whether for hypothesis formation or decision-making) is the grouping, or classification of measurements based on either (i) goodness-of-fit to a postulated model, or (ii) natural groupings (clustering) revealed through analysis. Cluster analysis is the organization of a collection of patterns (usually represented as a vector of measurements, or a point in a multidimensional space) into clusters based on similarity. Data clustering helps in analysis of data and obtains various feature of clustered data. It will help in gaining information easily.

Meta-heuristic algorithm are algorithm are used in many combinatorial optimization problem like Travelling salesman problem, image segmentation, data mining, generating test suite, feature extraction and many more. One of the major application of meta-heuristic algorithm in data clustering. There are many proposed meta-heuristic based data clustering algorithm like PSO clustering[8], GA clustering[9], ABC clustering [10] etc. which are providing better results as compare to traditional clustering algorithm. The main advantage of swarm based optimization algorithm is that, they cannot trapped in local optima. Main motivation of using MVO (Multi-verse optimizer) algorithm [11] in data clustering is that it will provide better results from other meta-heursistic based

clustering problem. As Multi-verse optimizer algorithm provides better optimization than other optimization algorithms.

1.5 Organization of thesis:-

The rest of this paper is organized as follows.

Chapter 2 provides details about the past research done on data clustering algorithm based on meta-heuristic search.

Chapter 3 provide details about Multi-verse optimization algorithm and UCI data set used in MVO-Clustering.

Chapter 4 shows the procedure as to how we applied Multi-verse optimizaytion algorithm to data clustering. This chapter also contains flowchart and pseudo code of MVO-Clustering.

Chapter 5 provides information Results and tabular comparision between other meat-heuristic algorithm and proposed algorithm.

Chapter6 is about conclusion and future work and references, where all the research articles which contributed to this research are listed.

Chapter Two: Literature Survey

2.1 K-mean Clustering:-

K-mean clustering algorithm is proposed by MacQueen in 1967. It is one of the simplest unsupervised learning algorithms that solve the well-known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The main idea is to define k centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early group age is done. At this point we need to re-calculate 'k' new centroids of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop we may notice that the k centroids change their location step by step until no more changes are done. In other words centroids do not move any more. Finally, this algorithm aims at minimizing an objective function, in this case a squared error function. The objective function used in k-mean is given as

$$J(w,z)=\sum_{i=1}^{N}\sum_{j=1}^{K} w_{ij}\|x_i-z_j\|^2 \qquad \text{eq 1}$$

Where K is the number of clusters, N the number of patterns, x_i (i =1. . . N) the location of the ith pattern and z_j(j = 1, . . . , K) is the center of the jth cluster.

2.2 Pseudo code of k-mean clustering

1. Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.
2. Assign each object to the group that has the closest centroid based on Euclidean distance.
3. When all objects have been assigned, recalculate the positions of the K centroids.

4. Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.

Although it can be proved that the procedure will always terminate, the k-means algorithm does not necessarily find the most optimal configuration, corresponding to the global objective function minimum. The algorithm is also significantly sensitive to the initial randomly selected cluster centers. The k-means algorithm can be run multiple times to reduce this effect. K-means is a simple algorithm that has been adapted to many problem domains

2.3 GA Clustering

2.3.1 Genetic Algorithm

Genetic algorithm is inspired by biological phenomena in living beings. In GA data sets encoded as string and these string collection produces the population within search space. Random population in initial stage is seeded in search space and for each string and objective function or fitness function is associated. On probabilistic basis we find the some strings which goes over mutation and crossover [12] which generates new population. These two operators are adopted from biological theory. This process continues over a period of time (iterations) to get the Optimization results.

2.3.2 Clustering Using GA:-

The optimization ability of GA is adopted in clustering the ‘n’ number of data set into ‘k’ number of fixed clusters. The Euclidean distance has been taken as similarity metric for assigning object to cluster. The objective function or fitness of each string or chromosome is modelled as cluster distance that should be minimized. Each string represents by the sequence of real number and the length of every string is ‘NxK’ words where ‘K’ is number of clusters centroid and ‘n’ is N-dimensional space. Every string (chromosome) is randomly initialize by ‘k’ random points. Mathematically the metric for clustering is given as:

$$M(C_1, C_2, \ldots, C_k) = \sum_{t=1}^{k} ||xj - zi|| \qquad \text{eq 2}$$

2.3.3 Pseudo Code for GA

Begin

1. t=0
2. initialize population p(t)
3. compute fitness p(t)
4. t = t+1
5. if termination criterion achieved go to step 10
6. select p(t) from p(t-1)
7. crossover p(t)
8. mutate p(t)
9. goto step 3
10. output best and stop.

End

Now fitness computation is two stage process. At first stage for every chromosome and for each data point we find the nearest cluster centroid on the basis of Euclidean distance. Finally assigned that data point to nearest cluster centroid as

$$||x_i - z_j|| < ||x_i - z_p||,\ p = 1,2,\ldots k, \text{ and } p \neq j. \qquad \text{eq3}$$

After all the data points are assigned in second Stage calculation of mean point of all the cluster centers encoded in the chromosome are replaced with calculated mean which is calculated according to the following equation:

$$Z^* = (1/n_i) \sum x_j , \quad i=1,2,\ldots,k. \qquad \text{eq4}$$

Now selection of some chromosome will be done on the basis of their fitness value and roulette wheel mechanism adopted for the chromosomes which goes under the mating pool. In mating pool two operation are applied on the chromosome, they are mutation and crossover. In crossover to parent chromosome exchange their information to generate two child. In mutation each chromosome undergoes with some modification that results in new chromosome. These operation are responsible for generation of new and strong

population. Finally when condition criteria is met up we stop this genetic biological process.

2.4 PSO clustering:-

2.4.1 Particle Swarm Optimization algorithm:

Particles Swarm optimization is inspired by natural movement of the Bird flocks [13], schooling of fish, foraging behavior of ants etc. This is a population-based stochastic search algorithm in which particle represents a solution. The complete swarm consist of number of solutions to the optimization problems. Basically PSO find the optimal position of particle to optimize the given fitness function. Each particle updates its position to optimize the results on the basis of particles local best position attain so far and the best position of the particle in its neighborhood. Each particle represents as there attribute tuple.

- ✓ Particle's current position.
- ✓ Particle's Current velocity.
- ✓ Particle's personal best position.

Particle position is updated according to equation:

Over the fixed number of iteration particle update its velocity and position to achieve the best position.

2.4.2 Clustering Using PSO:-

There are two implementation of PSO clustering.

- ✓ Using only PSO for Clustering.
- ✓ PSO hybrid with K-mean Clustering.

In first clustering technique each particle represent as 'k' cluster centroid vectors. The number of candidate clustering is represented by Swarm. For each data vectors the objective function associated for each particle is intra-cluster distance which is calculated by using Euclidean distance formula over the fixed numbers of iteration, each particle and for every data vector we calculate the Euclidean distance of data vector from all

cluster centroids. The data vector is assigned to that cluster centroid which results is minimum Euclidean distance. The movement of centroid are based on the Particle Swarm optimization based equation of velocity and position this process continues to acquiring best cluster centroid. In second clustering technique k-mean is used with PSO to provide better results than PSO clustering [14]. The main advantage of K-means is that it converges faster than particle Swarm optimization but results are not that much optimized. Also k-means could not escape from local optima and particle Swarm optimization has the ability to achieve global optima. This pros and cons of both algorithm is the main inspiration to develop this PSO-k-means hybrid algorithm. This algorithm works similar as PSO based clustering but the only difference is that here initial population is not randomly generated. K-mean algorithm provides the initial cluster centroids to the PSO algorithm in order to get better results from simple PSO clustering.

2.5 Black holes clustering:-

2.5.1 Black Hole Optimization algorithm:

Nature is always the main inspiration of meta-heuristic algorithm. Black hole optimization algorithm [15] is based on Universe based phenomenon of black hole. Physicists theory on Black hole depict that it is a region which having extremely high gravitational pull. That results in all the nearby object are attracted and collapses in it. It has too much gravitational pull even the light cannot passes away from black hole. Black hole optimization is population-based stochastic search technique in which best solution represents the black hole and other candidate solution represent the stars. When initialization begins population goes through objective function. The best solution obtained using objective function is termed as Black hole. The black hole start pulling the stars. The nearby stars get attracted and collapsed by Black hole which results in generation of new stars and these newly generated stars are placed in search space. This process carried out over a period of time known as iteration in order to optimize the results.

2.5.2 Clustering using Black hole optimization algorithm:-

Data clustering by using black hole optimization technique used for developing the cluster centroids produces good results. In Black hole clustering [16] the search space is initially seeded by ‘k’ random cluster centroid vectors. The initial population consists of stars and no black hole. Each star is represent as ‘k’ cluster centroid vectors. For each data vector and all the cluster centroid are operated on the distance matric that is Euclidean distance. The data vector is assigned to the cluster centroid that having least distance from the cluster centroid. The best solution obtained so far represents the black hole. The movement of cluster centroids of all other candidate solution i.e. stars towards the black hole is governed by the gravitational laws of Physics. This movements of stars centroid responsible for abrupt changes in the search space. The abrupt changes leads to the optimization of the clustering problem.

2.6 Ant Colony clustering:-

2.6.1 Ant colony optimization:

Ant Colony Optimization [17] is nature inspired stochastic approach and ant colony optimization copies foraging behavior of ant. Search space in ant colony optimization are seeded with population of an agent. These ant agents used to provide optimal results for combinatorial optimization problem. Ant agents start their random walk in search of food. The main characteristic of an agent is that they laid pheromone trails on their covered path. Whenever the ants find food source they head back to their colonies with lying path of pheromone trail. This pheromone trail are followed by other search agents to find the food source. When two or more food source have been found, the shortest path is governed by the pheromone trail. Pheromone trail is start evaporating with respect to time. Shortest path consists of high concentration of pheromone trail and other paths have low concentration of phenomenon. This will result in finding best solution among others candidate solutions.

2.6.2 Clustering using ACO algorithm:

Ant based clustering [18] mimic the complete ant foraging behavior to obtain optimize cluster centroid. A candidate solution is associated with each ant agents. Ant agent is represent as randomly generated cluster centroid vector. The search space is populated by number of ant agents. These ant agents and are moved over the fixed number of iteration to get the optimal cluster centroid. The similarity Matrix used in ant colony clustering is mean square distance. On the basis of nearest cluster centroids, data vectors are assigned to different cluster. The movement of cluster centroid of every ant agent is dependent on the pheromone trail associated with them. The objective function that would be minimized is the intra cluster distance. Evaporation of pheromone trail with respect to time leads to the optimal cluster centroid.

2.7 GSA-heuristic approach for Data clustering:-

2.7.1 Gravitational search Algorithm:

GSA [19] is nature inspired algorithm based on Newtonian theory of gravity. This optimization algorithm used the law of gravity [20]. In the proposed algorithm, agents are considered as objects and their performance is measured by their masses. All these objects attract each other by the gravity force, and this force causes a global movement of all objects towards the objects with heavier masses. Hence, masses cooperate using a direct form of communication, through gravitational force. The heavy masses – which correspond to good solutions – move more slowly than lighter ones, this guarantees the exploitation step of the algorithm. In GSA, each mass (agent) has four specifications: position, inertial mass, active gravitational mass, and passive gravitational mass. The position of the mass corresponds to a solution of the problem, and its gravitational and inertial masses are determined using a fitness function. In other words, each mass presents a solution, and the algorithm is navigated by properly adjusting the gravitational and inertia masses. By lapse of time, we expect that masses be attracted by the heaviest

mass. This mass will present an optimum solution in the search space. The GSA could be considered as an isolated system of masses. It is like a small artificial world of masses obeying the Newtonian laws of gravitation and motion. More precisely, masses obey the following laws:

- ✓ Law of gravity: each particle attracts every other particle and the gravitational force between two particles is directly proportional to the product of their masses and inversely proportional to the distance between them, R. We use here R instead of R2, because according to our experiment results, R provides better results than R2 in all experimental cases.

- ✓ Law of motion: the current velocity of any mass is equal to the sum of the fraction of its previous velocity and the variation in the velocity. Variation in the velocity or acceleration of any mass is equal to the force acted on the system divided by mass of inertia.

2.7.2 GSA-Heuristic search based Clustering:

GSA-Heuristic search based clustering [21] composed of two main parts. The first part is the gravitational search algorithm. In this part some candidate solutions for clustering problem are created randomly, and then communicate via Newtonian gravity law to move in the problem space for searching better solutions. At the end of first part an initial solution is found. At the second part, which is an improvement part, the initial solution is further improved using a heuristic search algorithm. The structure of the proposed heuristic search for the second part of the hybrid proposed algorithm is as following:

At first, a threshold value will be considered as an initial step of movement for the algorithm. This value will be added to all attributes in the initial solution one by one. In the other word, the threshold will be added to the first attribute in the first centroid and then the fitness value of the objective function will be recalculated for the new produced centroid, and will be compared by the fitness value of the original or the previous centroid. If there is an improvement in the new centroid, the old centroid will be replaced by the new centroid. Otherwise, the old centroid will be reloaded and the search direction changes to the opposite side. Meaning that, the threshold value will be subtracted from the old value of respective attribute, and the above procedure will be done again. If there is no improvement in both sides of the current attribute in the current centroid using the current threshold, the threshold value of respective attribute will be divided by two for

the next iteration. The above mentioned procedure will be repeated for all attributes of current centroid and then for other centroids sequentially until the termination criteria are satisfied.

2.8 ABC Clustering:-

2.8.1 Artificial Bee Colony optimization algorithm:

Artificial Bee Colony (ABC) algorithm was proposed by Karaboga for optimizing numerical problems in [22].In the ABC algorithm, the colony of artificial bees contains three groups of bees: employed bees, onlookers and scouts. A bee waiting on the dance area for making decision to choose a food source, is called an onlooker and a bee going to the food source visited by itself previously is named an employed bee. A bee carrying out random search is called a scout. In the ABC algorithm, first half of the colony consists of employed artificial bees and the second half constitutes the onlookers. The number of employed bees is equal to the number of food sources around the hive. The employed bee whose food source is exhausted by the employed and onlooker bees becomes a scout. In the ABC algorithm, each cycle of the search consists of three steps: sending the employed bees onto the food sources and then measuring their nectar amounts; selecting of the food sources by the onlookers after sharing the information of employed bees and determining the nectar amount of the foods; determining the scout bees and then sending them onto possible food sources. At the initialization stage, a set of food source positions are randomly selected by the bees and their nectar amounts are determined. Then, these bees come into the hive and share the nectar information of the sources with the bees waiting on the dance area within the hive. At the second stage, after sharing the information, every employed bee goes to the food source area visited by herself at the previous cycle since that food source exists in her memory, and then chooses a new food source by means of visual information in the neighbourhood of the present one. At the third stage, an onlooker prefers a food source area depending on the nectar information distributed by the employed bees on the dance area. When the nectar of a food source is abandoned by the bees, a new food source is randomly determined by a scout bee and

replaced with the abandoned one. This process is carried out to attain the global optima of the problem.

2.8.2 Clustering using ABC optimization algorithm:

Data clustering using ABC optimization algorithm completely mimic the behavior of bees for searching and finding the food source. In the algorithm, the first half of the colony consists of employed artificial bees and the second half constitutes the onlookers. The number of the employed bees or the onlooker bees is equal to the number of solutions (the cluster centers) in the population. At the first step, the ABC generates a randomly distributed initial population P(C = 0) of SN solutions (food source positions), where SN denotes the size of population. Each solution z_i where i = 1, 2. . . S_N is a D-dimensional vector. Here, D is the number of product of input size and cluster size for each data set. After initialization, the population of the positions (solutions) is subjected to repeated cycles, C = 1, 2, . . . , MCN, of the search processes of the employed bees, the onlooker bees and scout bees in order to obtain optimal intra cluster distance. Searching of better solution and finding the new position of cluster centroid is carried out similar to that ABC optimization problem. The cluster centroid of which the nectar is abandoned by the bees is replaced with a new cluster centroid by the scouts. In ABC, this is simulated by producing a position randomly and replacing it with the abandoned one. In ABC, providing that a position cannot be improved further through a predetermined number of cycles, then that cluster centroid is assumed to be abandoned. The value of predetermined number of cycles is an important control parameter of the ABC algorithm, which is called "limit" for abandonment. Assume that the abandoned cluster centroid is z_i and j ∈ {1, 2. . . D}, then the scout discovers a cluster centroid to be replaced with z_i. This operation can be defined as in (7)

$$z^j_i = z^j_{min} + \text{rand}\,(0, 1)\,(z^j_{max} - z^j_{min}) \qquad \text{eq 5}$$

After each candidate source position *v*i,j is produced and then evaluated by the artificial bee, its performance is compared with that of its old one. If the new food source has an equal or better nectar than the old source, it is replaced with the old one in the memory. Otherwise, the old one is retained in the memory. In other words, a greedy selection

mechanism is employed as the selection operation between the old and the candidate one. There are three control parameters in the ABC: the number of cluster centroid which is equal to the number of employed or onlooker bees (SN), the value of limit, the maximum cycle number (MCN).The distance similarity metrics adopted in ABC clustering [23] is sum of squared Euclidean distance which is given by:

$$J(w,z)=\sum_{i=1}^{N}\sum_{j=1}^{K} w_{ij}\|x_i-z_j\|^2 \qquad \text{eq 6}$$

Where K is the number of clusters, N the number of patterns, x_i (i =1. . . N) the location of the ith pattern and zj(j = 1, . . . , K) is the center of the jth cluster, to be found by Eq. (7):

$$Zj = (1/N_j)\sum_{i=1}^{N} w_{ij}.x_i \qquad \text{eq 7}$$

2.9 Evolutionary Particle Swarm Optimization Clustering:-

EPSO data clustering [24] is also the other variant of PSO clustering. The proposed algorithm is based on the evolution of swarm generations where the particles are initially uniformly distributed in the input data space and after a specified number of iterations; a new generation of the swarm evolves. The swarm tries to dynamically adjust itself after each generation to optimal positions. Evolutionary particle swarm optimization based clustering EPSO-Clustering is based on the idea of the generation based evolution of the swarm. The swarm evolves through different intermediate generations to reach a final generation. Particles are initialized in the first generation and after each generation the swarm evolves to a stronger swarm by consuming the weaker particles of that generation by the stronger ones. The stronger the particle is, the greater its chance of survival to the next generation. Stronger particles make mature and stable generations. The strongest generation reached with an optimal number of clusters and lowest intra-cluster distance represent an optimal solution of the problem. The following are the main concepts explaining the functionality of the technique:

• A particle (bird) represents the centroid data vector around which the food is scattered.

• Input data vectors are represented as *food* for the swarm.

• Iteration is a single search by all the particles for acquiring food.

• A generation of the swarm is a specified number of iterations of the swarm looking for food in the search input data space.

• The strength of the particle is represented by the amount of food acquired by that particular particle. The more food associated with a particle the greater the chance of survival to the next generation.

• The generation is said to be a stronger one if the generation contains no weaker particles and particle are positioned at their more optimal locations.

• A weaker particle is consumed by the nearest stronger particle by acquiring all the data vectors associated with that weaker particle.

• During consumption the acquiring particle's new attributes (position, distance vector) are obtained after averaging its attribute with the attributes of the particle to be consumed.

• Particle destruction is the death of a weaker particle, which takes place after the consumption phase. A particle is no longer a part of the swarm when it is consumed by a stronger particle.

The swarming process of EPSO-clustering starts the first generation of the swarm by initializing the particles to the data vectors from the input data in a uniform manner1 so that they cover the whole input space. During the first iteration the particles start searching for food, pickup their corresponding food and update their position. After a predefined number of iterations the next generation of the particles start. This generation will have better particles after the weaker particles are consumed by the nearest stronger particles in the previous generation. The swarm evolves until an optimal strong generation of particles is obtained on the basis of the number of generations, number of iterations or minimum number of data vectors in a cluster.

Each and every particle is representative of a candidate clustering centroid and moves with the following attributes:

• Particle-Id: this is the unique identifier for the particle. The value of Particle-Id is between 1 to total number of particles.

• Particle current position: this represents the position of the current particle in all dimensions.

• Distance vector: this stores the distances between the particle and all data vectors at any given iteration.

• Associated data vectors: those data vectors which are won by the particle at a given iteration.

• The corresponding data vectors which are currently consumed by this particle.

• pBest position: pBest position of the particle is the position of the nearest data vector to the particle achieved so far. This is obtained by keeping track of the position of the best previous data vector won by the particle i.e. the nearest data vector the particle has achieved in a specified generation.

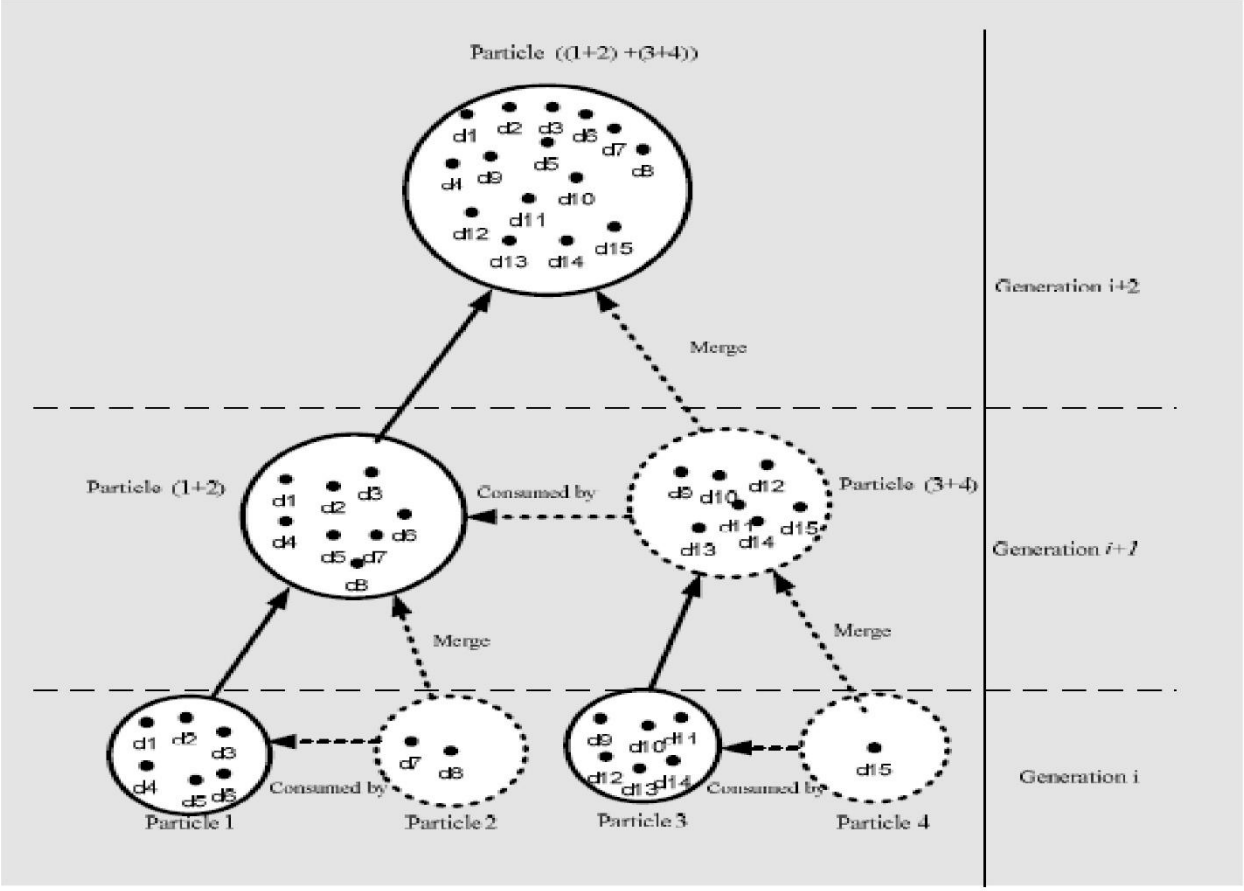


Figure 1 EPSO clustering

Figure 1 shows an example of how the particles evolve during different generations. During generation-*i* two particles particle 2 and particle 4 are consumed by particle 1 and particle 3 respectively. The resultant configuration has stronger particles thus formed from acquiring the food of the weaker particles.

Chapter Three: Research Methodology

Research Methodology consist of basic algorithm of the proposed clustering technique. To do the cluster analysis Multi-Verse optimizer algorithm is used. The challenging Data set used to classify them into different sets or cluster.

3.1 Multi-verse optimizer algorithm:

3.1.1 Inspiration

The big bang theory [25] discusses that our universe starts with a massive explosion. According to this theory, the big bang is the origin of everything in this world, and there was nothing before that. Multi-verse theory is another recent and well-known theory between physicists. It is believed in this theory that there are more than one big bang and each big bang causes the birth of a universe. The term multi-verse stands opposite of universe, which refers to the existence of other universes in addition to the universe that we all are living in. Multiple universes interact and might even collide with each other in the multi-verse theory. The multi-verse theory also suggests that there might be different physical laws in each of the universes. We chose three main concepts of the multi-verse theory as the inspiration for the MVO algorithm: white holes, black holes, and wormholes. A white hole has never seen in our universe, but physicists think that the big bang can be considered as a white hole and may be the main component for the birth of a universe. It is also argued in the cyclic model of multi-verse theory [26] that big bangs white holes are created where the collisions between parallel universes occur. Black holes, which have been observed frequently, behave completely in contrast to white wholes. They attract everything including light beams with their extremely high gravitational force [27]. Wormholes are those holes that connect different parts of a universe together. The wormholes in the multi-verse theory act as time/space travel tunnels where objects are able to travel instantly between any corners of a universe (or even from one universe to another). Conceptual models of these three key components of the multi-verse theory are illustrated in Fig. 1. Every universe has an inflation rate (eternal inflation) that causes its expansion through space [28]. Inflation speed of a universe is very important in terms of forming stars, planets, asteroids, black holes, white

holes, wormholes, physical laws, and suitability for life. It is argued in one of the cyclic multi-verse models [29] that multiple universes interact via white, black, and wormholes to reach a stable situation. This is the exact inspiration

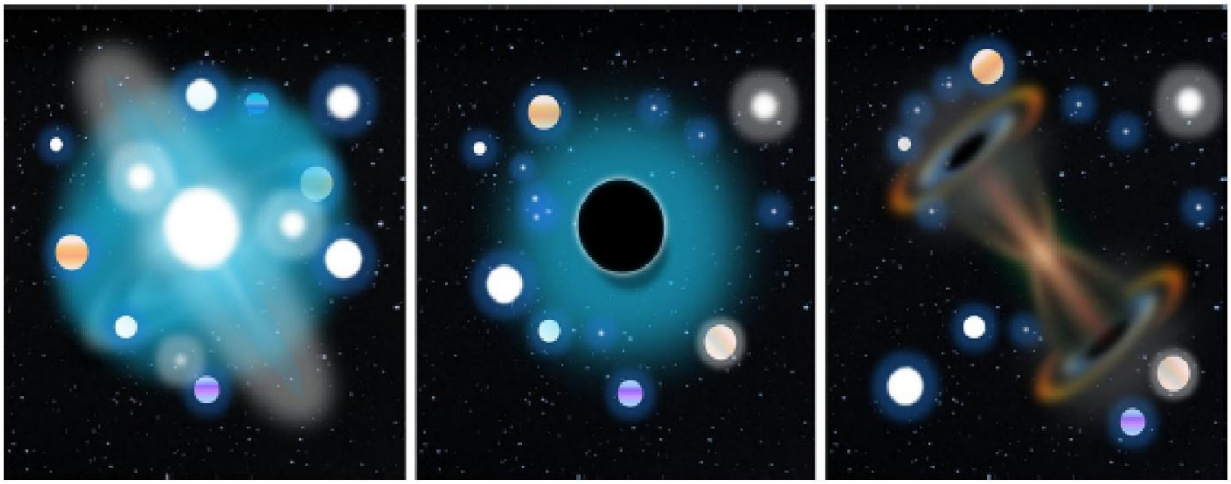


Figure 2 White hole, Black hole, Worm hole

the MVO algorithm, which is conceptually and mathematically modelled in the following subsection.

3.1.2 MVO algorithm

As discussed in the preceding section, a population-based algorithm divides the search process into two phases: exploration versus exploitation. We utilize the concepts of white hole and black hole in order to explore search spaces by MVO. In contrast, the wormholes assist MVO in exploiting the search spaces. We assume that each solution is analogous to a universe and each variable in the solution is an object in that universe. In addition, we assign each solution an inflation rate, which is proportional to the corresponding fitness function value of the solution. We also use the term time instead of the iteration in this paper since it is a common term in multi-verse theory and cosmology. During optimization, the following rules are applied to the universes of MVO:

1. The higher inflation rate, the higher probability of having white hole.
2. The higher inflation rate, the lower probability of having black holes.
3. Universes with higher inflation rate tend to send objects through white holes.

4. Universes with lower inflation rate tend to receive more objects through black holes.
5. The objects in all universes may face random movement towards the best universe via wormholes regardless of the inflation rate.

The conceptual model of the proposed algorithm is illustrated in Fig. 2.

This figure shows that the objects are allowed to move between different universes through white/black hole tunnels. When a white/black tunnel is established between two universes, the universe with higher inflation rate is considered to have white hole, whereas the universe with less inflation rate is assumed to own black holes.

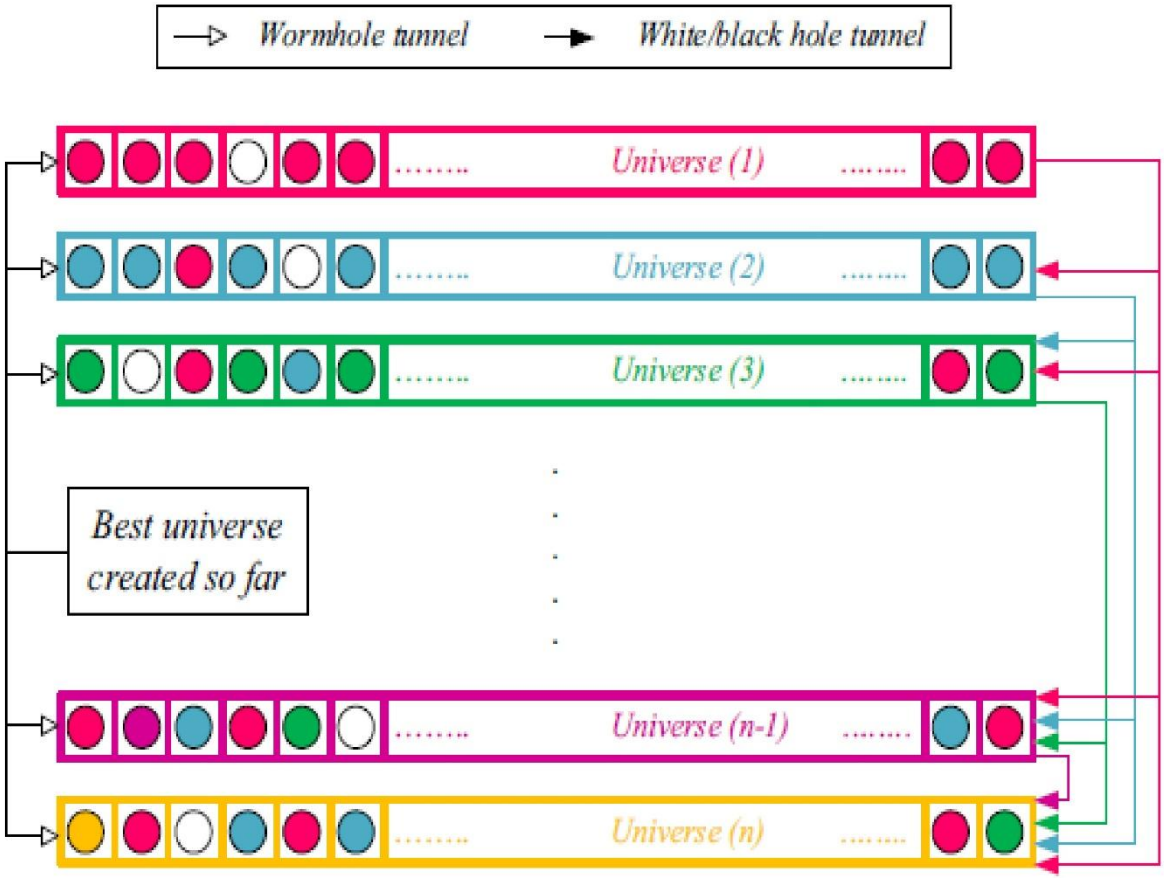


Figure 3 Conceptual model of the proposed MVO algorithm (I(U1) I(U2).....I(Un-1)[I(Un))

The objects are then transferred from the white holes of the source universe to black holes of the destination universe. This mechanism allows the universes to easily exchange objects. In order to improve the whole inflation rate of the universes, we assume that the universes with high inflation rates are highly probable to have white holes. In contrary, the universes with low inflation rates have a high probability of having black holes. Therefore, there is always high possibility to move objects from a universe with high inflation rate to a universe with low inflation rate. This can guarantee the improvement of the average inflation rates of the whole universes over the iterations. In order to mathematically model the white/black hole tunnels and exchange the objects of universes, we utilized a roulette wheel mechanism. At every iteration, we sort the universes based of their inflation rates and chose one of them by the roulette wheel to have a white hole. The following steps are done in order to do this.

Assume that

$$U = \begin{bmatrix} x_1^1 & x_1^2 & \dots & x_1^d \\ x_2^1 & x_2^2 & \dots & x_2^d \\ \vdots & \vdots & \vdots & \vdots \\ x_n^1 & x_n^2 & \dots & x_n^d \end{bmatrix}$$

where d is the number of parameters (variables) and n is the number of universes (candidate solutions):

$$x^j_i = \begin{cases} x^j_k & r1 < NI(Ui) \\ x^j_i & r1 >= NI(Ui) \end{cases} \qquad \text{eq(8)}$$

where x^j_i indicates the jth parameter of ith universe, Ui shows the ith universe, NI(Ui) is normalized inflation rate of the ith universe, r1 is a random number in [0, 1], and x^j_k

indicates the jth parameter of kth universe selected by a roulette wheel selection mechanism.

The pseudocodes for this part are as follows:

```
SU=Sorted universes
NI=Normalize inflation rate (fitness) of the universes
    for each universe indexed by i
        Black_hole_index=i;
            for each object indexed by j
                r1=random([0,1]);
                if r1<NI(Ui)
                    White_hole_index= RouletteWheelSelection(-NI);
                    U(Black_hole_index,j)= SU(White_hole_index,j);
                end if
            end for
    end for
```

As may be seen in these pseudocodes and Eq. (3.1), the selection and determination of white holes are done by the roulette wheel, which is based on the normalized inflation rate. The less inflation rate, the higher probability of sending objects though white/black hole tunnels. Please note that -NI should be changed to NI for the maximization problems. The exploration can be guaranteed using this mechanism since the universes are required to exchange objects and face abrupt changes in order to explore the search space. With the above mechanism, the universes keep exchanging objects without perturbations. In order to maintain the diversity of universes and perform exploitation, MVO consider that each universe has wormholes to transport its objects through space randomly. In Fig. 2, white points represent transferred objects through the wormholes. It may be observed that the wormholes randomly change the objects of the universes without consideration of their inflation rates. In order to provide local changes for each universe and have high probability of improving the inflation rate using wormholes, we

assume that wormhole tunnels are always established between a universe and the best universe formed so far. The formulation of this mechanism is as follows:

$$X^j_i = \begin{cases} \begin{cases} X_j + TDR*((ub_j - lb_j)*r4 + lb_j) \\ \\ X_j - TDR*((ub_j - lb_j)*r4 + lb_j) \end{cases} & r2 < WEP \\ X^j_i & r2 >= WEP \end{cases} \quad \text{eq(9)}$$

where Xj indicates the jth parameter of best universe formed so far, TDR is a coefficient, WEP is another coefficient, lbj shows the lower bound of jth variable, ubj is the upper bound of jth variable, x^j_i indicates the jth parameter of ith universe, and r2, r3, r4 are random numbers in [0, 1].

The pseudocodes are as follows (assuming that ub and lb indicate upper bound and lower bound of the variables):

```
for each universe indexed by i
    for each object indexed by j
        r2=random([0,1]);
        if r2<Wormhole_existance_probability
            r3= random([0,1]);
            r4= random([0,1]);
            if r3<0.5
            U(i,j)=Best_universe(j) + Travelling_distance_rate * (( ub(j) -
            lb(j)) *r4 + lb(j));
```

else

U(i,j)=Best_universe(j) - Travelling_distance_rate * ((ub(j) - lb(j))*r4 + lb(j));

end if

end if

end for

end for

It may be inferred from the pseudo codes and mathematical formulation that there are two main coefficients herein: wormhole existence probability (WEP) and travelling distance rate (TDR). The former coefficient is for defining the probability of wormhole's existence in universes. It is required to increase linearly over the iterations in order to emphasize exploitation as the progress of optimization process. Travelling distance rate is also a factor to define the distance rate (variation) that an object can be teleported by a wormhole around the best universe obtained so far. In contrast to WEP, TDR is increased over the iterations to have more precise exploitation/local search around the best obtained universe. Wormhole existence and travelling distance rates are illustrated in Fig. 3. The adaptive formula for both coefficients are as follows:

where min is the minimum value, max is the maximum value, l indicates the current iteration, and L shows the maximum iterations.

$$WEP = min + l^* ((max - min) / L) \quad \text{eq 10}$$

$$TDR = 1-(pow(l , 1/p))/(pow(L , 1/p)) \quad \text{eq 11}$$

where p defines the exploitation accuracy over the iterations. The higher p, the sooner and more accurate exploitation/local search. Note that WEP and TDR can be considered as constants as well, but we recommend adaptive values according to the MVO methodolgy. In the MVO algorithm, the optimization process starts with creating a set of random universes. At each iteration, objects in the universes with high inflation rates tend

to move to the universes with low inflation rates via white/ black holes. Meanwhile, every single universe faces random teleportations in its objects through wormholes towards the best universe. This process is iterated until the satisfaction of an end criterion (a pre-defined maximum number of iterations, for instance).

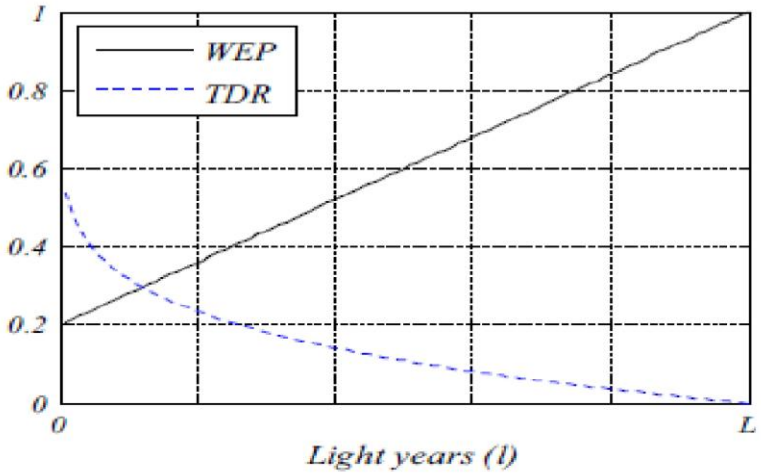


Figure 4 Worm hole exixtence probablity versus Travelling distance rate

The computational complexity of the proposed algorithms depends on number of iterations, number of universes, roulette wheel mechanism, and universe sorting mechanism. Sorting universe is done in every iteration, and we employ the Quicksort algorithm, which has the complexity of O (n log n) and O(n^2) in the best and worst case, respectively. The roulette wheel selection is run for every variable in every universe over the iterations and is of O (n) or O(log n) based on the implementation. Therefore, the overall computational complexity is as follows:

$$O(MVO)=O(l(O(quicksort)+n*d*(O(roulette\ \ wheel)))) \quad \text{eq(12)}$$

$$O(MVO)=O(l(n^2 + n*d*\log(n))) \quad \text{eq(13)}$$

where n is the number of universes, l is the maximum number of iterations, and d is the number of objects.

To see how the proposed algorithm theoretically has the potential to solve optimization problems, some observations are as follows:

- White holes are more possible to be created in the universes with high inflation rates, so they can send objects to other universes and assist them to improve their inflation rates.
- Black holes are more likely to be appeared in the universes with low inflation rates so they have higher probability to receive objects from other universes. This again increases the chance of improving inflation rates for the universes with low inflation rates.
- White/black hole tunnels tend to transport objects from universes with high inflation rates to those with low inflation rates, so the overall/average inflation rate of all universes is improved over the course of iterations.
- Wormholes tend to appear randomly in any universe regardless of inflation rate, so the diversity of universes can be maintained over the course of iterations.
- While/black hole tunnels require universes to abruptly change, so this can guarantee exploration of the search space.
- Abrupt changes also assist resolving local optima stagnations.
- Wormholes randomly re-span some of the variables of universes around the best solution obtained so far over the course of iterations, so this can guarantee exploitation around the most promising region of the search space.
- Adaptive WEP values smoothly increase the existence probability of wormholes in universes. Therefore, exploitation is emphasized during optimization process.
- Adaptive TDR values decrease the travelling distance of variables around the best universe, a mechanism for increasing the accuracy of local search over the iterations.
- The convergence of the proposed algorithm is guaranteed by emphasizing exploitation/local search proportional to the number of iterations.

3.2 Data Set used in Clustering:

Widely used dataset in clustering problem are usually multivariate. These are available in the repository of the machine learning databases [30]. These dataset have more than two dimension. Some of widely used data sets are:

3.2.1 IRIS Dataset:

Iris dataset [31] consist of a flower which having three types of breed. It is perhaps the best known database to be found in the pattern recognition literature. The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. Each dataset has four attributes. One class is linearly separable from the other 2; the latter are NOT linearly separable from each other.

Predicted attribute: class of iris plant.

.**Attribute Information:**

1. Sepal length in cm.
2. Sepal width in cm.
3. Petal length in cm.
4. Petal width in cm
5. Class:
    - Iris Setosa
    - Iris Versicolour
    - Iris Virginica

3.2.2 Wine dataset:

These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines. The initial data set had around 30 variables, but for some reason I only have the 13 dimensional version. It consist a list of what the 30 or so variables were, but a.) I lost it, and b.), I would not know which 13 variables are included in the set. It consist of 178 instances and 3 types of wine. Each classes have 59, 71, 48 instances of each class.

Attribute Information:

The attributes are

1. Alcohol
2. Malic acid
3. Ash
4. Alcalinity of ash
5. Magnesium
6. Total phenols
7. Flavanoids
8. Nonflavanoid phenols
9. Proanthocyanins
10. Color intensity
11. Hue
12. OD280/OD315 of diluted wines
13. Proline

In a classification context, this is a well posed problem with "well behaved" class structures. A good data set for first testing of a new classifier, but not very challenging.

3.2.3 Glass Dataset:

The study of classification of types of glass was motivated by criminological investigation. At the scene of the crime, the glass left can be used as evidence...if it is correctly identified. This glass dataset consist of 9 attributes and 6 types of glasses which results in 6 cluster.the number of instaces consist in all cluster is 70,76,13,9 and 29.

Attribute Information:

1. RI: refractive index
2. Na: Sodium (unit measurement: weight percent in corresponding oxide, as are attributes 4-10)
3. Mg: Magnesium
4. Al: Aluminum
5. Si: Silicon
6. K: Potassium
7. Ca: Calcium
8. Ba: Barium
9. Fe: Iron
10. Type of glass: (Clusters)
    - Building windows float processed
    - Building windows non float processed
    - Vehicle windows float processed
    - Containers
    - Tableware
    - Headlamps

3.2.4 Cancer Dataset

This is one of three domains provided by the Oncology Institute that has repeatedly appeared in the machine learning literature. This Breast cancer data set includes 444 instances of one class and 239 instances of another class. The instances are described by 9 attributes, some of which are linear and some are nominal.

Attribute Information:

1. Class: no-recurrence-events, recurrence-events
2. Age
3. Menopause
4. Tumor-size
5. Inv-nodes

6. Node-caps
7. Deg-malig
8. Breast:
9. Breast-quad:
10. Irradiat:.

3.2.5 CMC Dataset:

This dataset is a subset of the 1987 National Indonesia Contraceptive Prevalence Survey. The samples are married women who were either not pregnant or do not know if they were at the time of interview. The problem is to predict the current contraceptive method choice (no use, long-term methods, or short-term methods) of a woman based on her demographic and socio-economic characteristics.

Attribute Information:

1. Wife's age (numerical)
2. Wife's education (categorical) 1=low, 2, 3, 4=high
3. Husband's education (categorical) 1=low, 2, 3, 4=high
4. Number of children ever born (numerical)
5. Wife's religion (binary) 0=Non-Islam, 1=Islam
6. Wife's now working? (Binary) 0=Yes, 1=No
7. Husband's occupation (categorical) 1, 2, 3, 4
8. Standard-of-living index (categorical) 1=low, 2, 3, 4=high
9. Media exposure (binary) 0=Good, 1=Not good
10. Contraceptive method used (class attribute) 1=No-use, 2=Long-term, 3=Short-term

Chapter Four: Proposed work

In this section, I have proposed MVO (Multi-verse optimizer) clustering. MVO clustering based on Multi verse optimization algorithm, in order to solve the clustering problem. This algorithm has taken similar idea that are incorporated in the previous algorithms that are described in literature survey section.

4.1 Clustering using MVO:-

MVO clustering consist of more than one universe. Each universe is analogues to the cluster solution. The searching capability of MVO has been used in this work for the purpose of appropriately determining a fixed number *K* of cluster centers in population metrics; thereby suitably clustering the set of *n* unlabeled points. The clustering metric that has been adopted is the sum of the Euclidean distances of the points from their respective cluster centers. Mathematically, the clustering metric M for the K clusters C_1, C_2,…, C_K is given by:

$$M(C_1, C_2, \ldots, C_k) = \sum_{t=1}^{k} \sum ||xj - zi||, \qquad j \in \text{Number of data points in } C_i \quad \text{eq(14)}$$

The task of the GA is to search for the appropriate cluster centers z_1, $z_2 \ldots z_k$ such that the clustering metric M is minimized. The population metrics is Nx(Kxd) size here N is the number of universes or and search agents, 'K' is the no. of cluster and 'd' is dimensions or attributes of each data set. Population metrics is given as:

$$U = \begin{bmatrix} X11 & \cdots & X1d \\ \vdots & \ddots & \vdots \\ Xn1 & \cdots & Xnd \end{bmatrix}$$

x^i_j indicates the ith attribute of jth cluster centroid and 'n' are the number of candidate solution. Out of 'n' candidate solution one will be the best solution that find by Multi-verse optimizer.

4.2 MVO-clustering algorithm:

The basic steps of MVO, which are also followed in the MVO-clustering algorithm, which are describe in MVO algorithm are now described in detail.

4.2.1 Universe representation

Each Universe is a sequence of real numbers representing the K cluster centres. For an d-dimensional space, the universe is represented by d*K objects, where the first‘d’ objects represent the d dimensions of the "first cluster center, the next d object represent those of the second cluster center, and so on. As an illustration let us consider the following example.

Example 1. Let N=2 and K=3, i.e., the space is two dimensional and the number of clusters being considered is three. Then a universe 51.6 72.3 18.3 15.7 29.1 32.2 is represents the three cluster centers (51.6, 72.3), (18.3, 15.7) and (29.1, 32.2). Note that each real number in the universe is an indivisible object.

4.2.2 Population initialization

The K cluster centers encoded in each chromosome are initialized to K randomly chosen points from the data set. This process is repeated for each of the N universes in the population, where N is the size of the population or search agents.

4.2.3 Objective function and computation

The fitness computation process consists of two phases. In the first phase, the clusters are formed according to the centers encoded in the chromosome under consideration. This is done by assigning each point x_i, i=1, 2,… n, to one of the clusters Cj with center z_j such that

$$||x_i-z_j||<||x_i-z_p||,\ p = 1,2,\ldots k, \text{ and } p!=j. \qquad \text{eq(16)}$$

All ties are resolved arbitrarily. After the clustering is done, the cluster centers encoded in the chromosome are replaced by the mean points of the respective clusters. In other words, for cluster Ci, the new center z^*_i is computed as

$$Z^* = (1/n_i) \sum x_j , \quad i = 1,2,\ldots,k \qquad \text{eq(17)}$$

These z^*_i s now replace the previous z* in the universe. As an illustration, let us consider the following example.

Example 2. The First cluster center in the universe considered in Example 1 is (51.6, 72.3). With (51.6, 72.3) as center, let the resulting cluster contain two more data points, that is (50.0, 70.0) and (52.0, 74.0) besides itself i.e., (51.6, 72.3). Hence the newly computed cluster center becomes ((50.0+52.0+51.6)/3, (70.0+74.0+72.3)/3) = (51.2, 72.1). The new cluster center (51.2, 72.1) now replaces the previous value of (51.6, 72.3). The objective function that is Intra-cluster has to minimize in order to obtain good results.

4.2.4 Exploration

MVO-Clustering utilize the concepts of white hole and black hole in order to explore search spaces of MVO algorithm. In order to obtain exploration the cluster centroid are exchanged between universes through white/black hole tunnels. When a white/black tunnel is established between two universes, the universe with higher inflation rate is considered to have white hole, whereas the universe with less inflation rate is assumed to own black holes. The cluster centroids are then transferred from the white holes of the source universe to black holes of the destination universe In order to mathematically model the white/black hole tunnels and exchange the objects of universes, we utilized a roulette wheel mechanism. At every iteration, we sort the universes based of their inflation rates and chose one of them by the roulette wheel to have a white hole. In this way exploration will be achieved.

$$x^j_i = \begin{cases} x^j_k & r1 < \text{Intra-cluster distance}(Ui) \\ x^j_i & r1 >= \text{Intra-cluster distance}(Ui) \end{cases} \qquad \text{eq(18)}$$

where x^{j}_{i} is the jth attribute of ith cluster centroid, xjk is the jth centrid of kth cluster centroid which is selected by roulette wheel mechanism and r1 is randomly generated number.

4.2.5 Exploitaion

In order to maintain the diversity of universes and perform exploitation, we consider that each universe has wormholes to transport its objects through space randomly. It may be observed that the wormholes randomly change the objects of the universes without consideration of their inflation rates. The formulation of this mechanism is as follows:

$$X^{j}_{i} = \begin{cases} \begin{cases} X_j + TDR*((ub_j - lb_j)*r4 + lb_j) & \\ & r2 < WEP \\ X_j - TDR*((ub_j - lb_j)*r4 + lb_j) & \end{cases} & \\ X^{j}_{i} & r2 >= WEP \end{cases} \quad \text{eq(19)}$$

where Xj indicates the jth attribute of best cluster solution formed so far, TDR is a coefficient, WEP is another coefficient, lbj shows the lower bound of jth attribute, ubj is the upper bound of jth attribute, xji indicates the jth attribute of ith universe, and r2, r3, r4 are random numbers in [0, 1].

4.2.6 Termination criterion

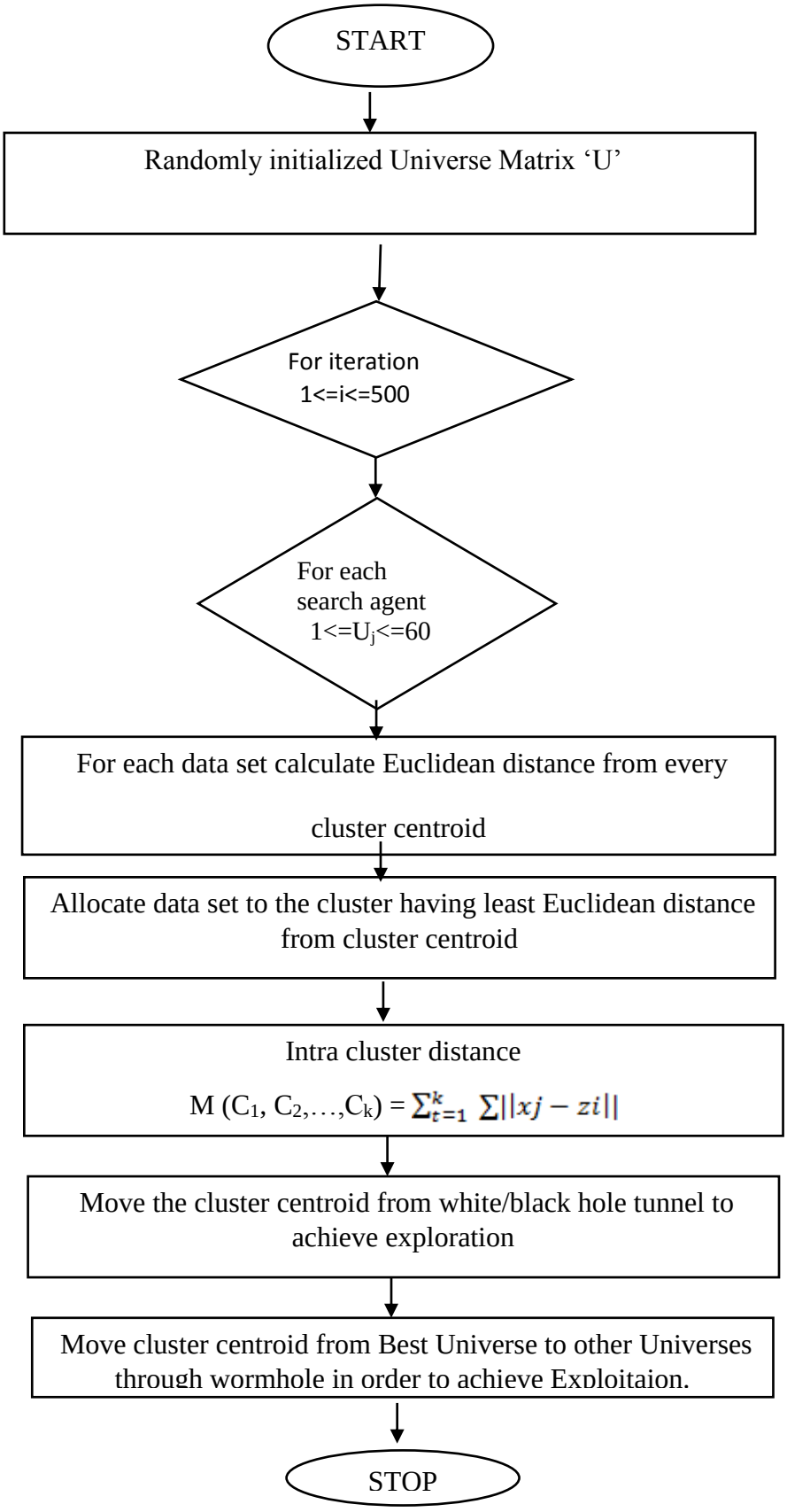
In the MVO-clustering the processes of fitness computation, exploration and exploitation are executed for a maximum number of iterations. The best universe seen up to the last step provides the solution to the clustering problem. When the maximum iteration reached or Intra cluster distance does not varies we stop the MVO-Clustering process.

4.3 Pseudo code and Flowchart of MVO clustering:

1. Initialize each Universe to contain Nc randomly selected cluster centroids.
2. For t = 1 to tmax do
    a) For each Universe Ui do
    b) For each data vector **z**p
        I. calculate the Euclidean distance d(**z**p, **m**ij)to all cluster centroids Cij
        II. assign **z**p to cluster Cij such that d(**z**p, **m**ij) = $\min_{c=1;\_\_;\ Nc}$ {d(**z**p;$\mathbf{m}_{ic}$)}
        III. calculate the fitness using equation (8)

    (c) Move the cluster centroid on the basis for equation 3.1

    (d) Find the Best Universe and move the cluster's centroid on the basis of equation 3.2
3. Stop.



Flowchart of MVO CLustering

Chapter Five: Results

Five benchmark datasets from UCI depository with a variety of complexity are used to evaluate the performance of the proposed approach. The datasets are Iris, Wine, Glass, Wisconsin Breast Cancer and Contraceptive Method Choice (CMC), which are available in the repository of the machine learning databases [30]. Below Table 1 summaries the main characteristics of the used datasets.

The performance of the MVO-clustering algorithm is compared against well-known and the most recent algorithms reported in the literature, including K-means , particle swarm optimization, and gravitational search algorithm . The performance of the algorithms is evaluated and compared using the Sum of intra-cluster distances as an internal quality measure: The distance between each data object and the center of the Corresponding cluster is computed and summed up. Clearly, the smaller the sum of intra-cluster distances, the higher the quality of the clustering. The sum of intra-cluster distances is also the evaluation fitness in this work.

As seen from the results the MVO Clustering algorithm achieved the best results among all the algorithms. For the Iris dataset, the best, worst, and average solutions obtained by MVO Clustering are 96.65589, 96.65681, and 96.6605, respectively, which are better than the other algorithms. For the Wine dataset, the MVO algorithm achieved the optimum value of 16293.41995, which is significantly better than the other test algorithms. As seen from the results for the Glass dataset, the MVO clustering algorithm is far superior to the other algorithms. The worst solution obtained by the MVO clustering algorithm on the Glass dataset is 213.95689, which is much better than the best solutions found by the other algorithms. For the Cancer dataset, the MVO clustering algorithm outperformed the K-means, PSO and GSA algorithms. For the CMC dataset, the proposed MVO clustering algorithm reached an average of 5533.63122, while other algorithms were unable to reach this solution even once within 50 runs. From the above results, we can say that in five of the test datasets the proposed MVO clustering algorithm is superior to the other test algorithms. It can find high quality solutions. In other words, the MVO clustering algorithm converges to global optimum in all the runs while the

other algorithms may get trapped in local optimum solutions. Only in the Cancer dataset did one of the algorithms (GSA) reach a better solution than the MVO clustering. Even in this dataset, the MVO clustering algorithm reached high quality clusters compared to the other three test algorithms.

The sum of intra-cluster distances obtained by algorithms on different datasets.

Dataset	Criteria	K-means	PSO	GSA	MVO-Clustering
Iris	Best	97.32592	96.87935	96.68794	96.6605
	Average	105.72902	98.14236	96.731051	96.6568
	Worst	128.40420	99.76952	96.824632	96.8605
Wine	Best	16,555.67	16,304.48	16,313.87	16,293.41
	Average	16,963.044	16,316.27	16,374.30	16,299.01
	Worst	23,755.049	16,342.78	16,428.86	16,300.22
Glass	Best	215.67753	223.90546	224.98410	203.86515
	Average	227.97785	230.49328	233.54329	211.49860
	Worst	260.83849	246.08915	248.36721	213.95689
Cancer	Best	2986.96134	2974.48092	2964.76394	2964.88878
	Average	3032.24781	2981.78653	2964.66312	2965.39539
	Worst	5216.08949	3053.49132	2993.24458	2964.45074
CMC	Best	5542.18214	5539.17452	5542.27631	5532.88323
	Average	5543.42344	5547.89320	5581.94502	5533.63122
	Worst	5545.33338	5561.65492	5658.76293	5534.77738

Table 1

Graph based comparison of K-mean and MVO Clustering is discussed in this section. MVO clustering completely outperform the K-mean clustering algorithm. The graph consist of number of iteration vs optimization in each iteration. This comparison have been done on five benchmark fuctions like Iris, Wine and Glass. The graph shown below:

1. **Iris dataset:** Graph shown below is the Inta-cluster distance vs number of iteration is given in which MVO clustering has optimal value of 96.6005 and k-mean clustering provides 97.3259 over the 100 number of iterations when applied on iris dataset.

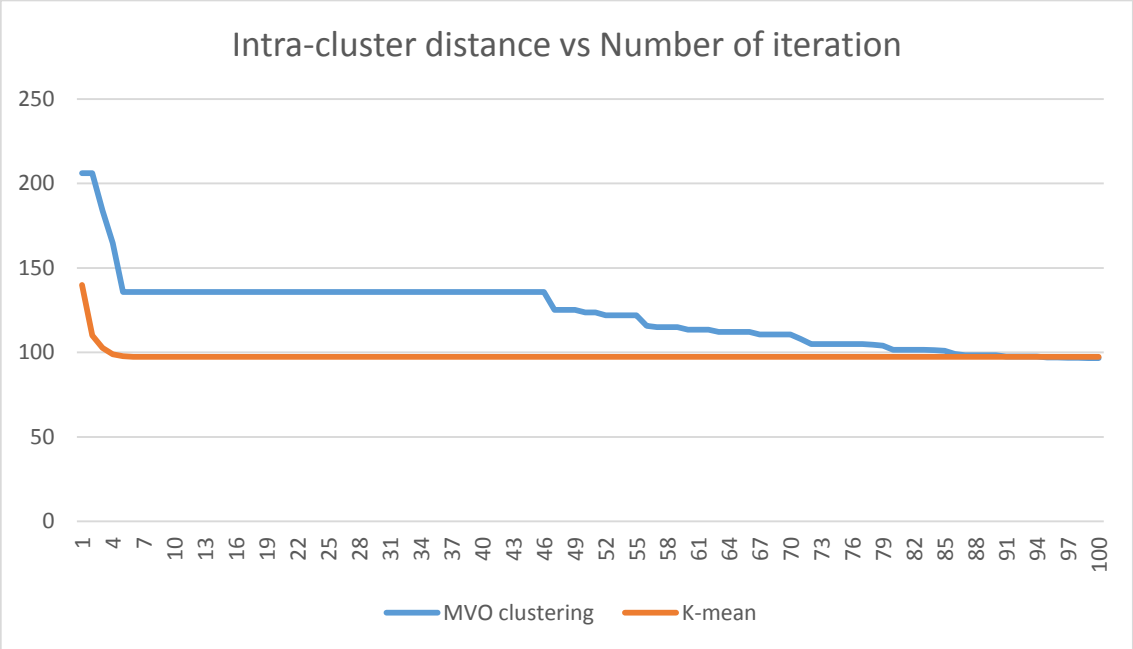


Figure 5 Iris dataset graph: Intra-cluster distance vsNumber of iteration

2. Glass dataset: Graph shown below is the Intra-cluster distance vs number of iteration is given in which MVO clustering has optimal value of 203.861 and k-mean clustering provides 215.8526 over the 100 number of iterations when applied on glass dataset.

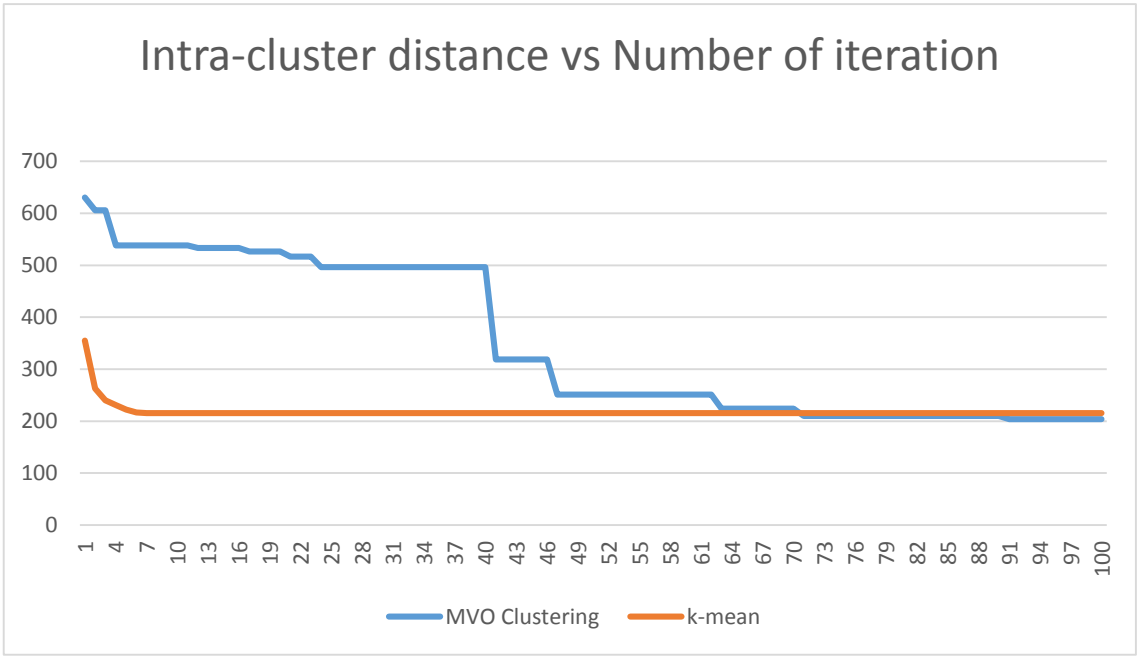


Figure 6 Glass dataset graph: Intra-cluster distance vs Number of iteration

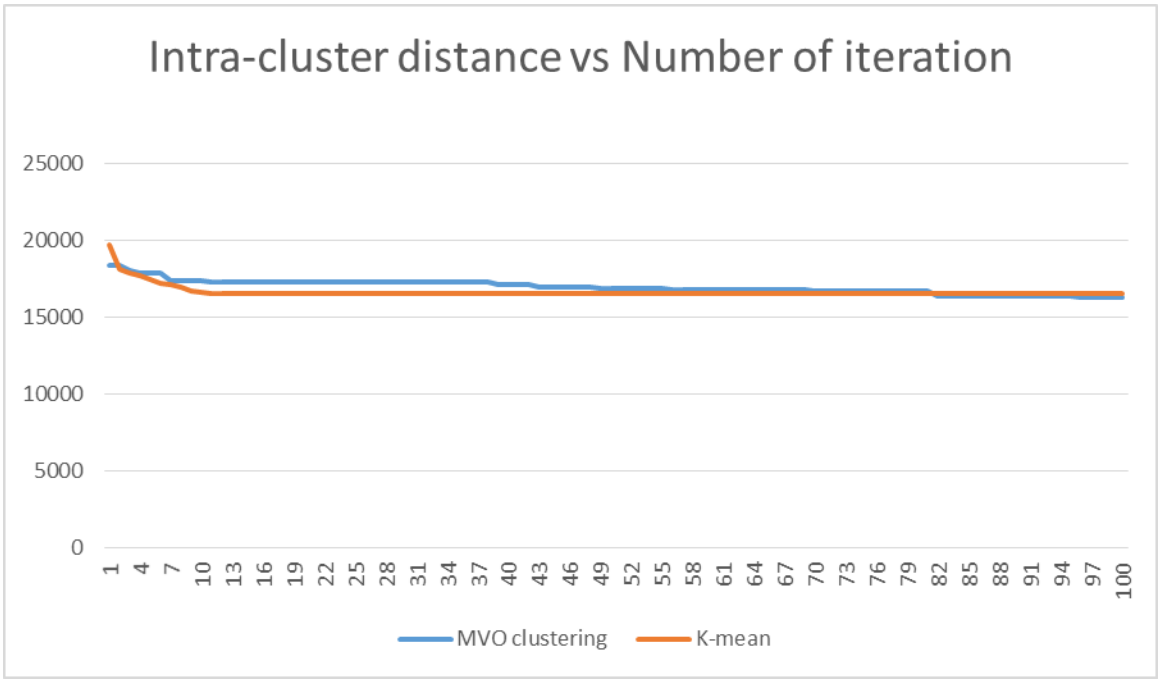


Figure 7 Wine dataset graph: Intra-cluster distance vs Number of iteration

3. Wine Dataset: Graph shown above is the Inta-cluster distance vs number of iteration is given in which MVO clustering has optimal value of 16299.01 and k-mean clustering provides 16,555.67 over the 100 number of iterations when applied on wine data.

Chapter Six: Conclusion and Future Work

Data clustering is a comprehensive and important technique in data mining and is attracting more researchers as the amount of data and need for information management increases. Clustering techniques aim to group similar data into identical clusters in an optimal manner. To achieve optimality in the process and in the results various optimization techniques have been used to improve one or another aspect of clustering. One of such optimization based technique is Multi-verse optimization algorithm. Our aim in this work is to tackle these problems by proposing a novel generation based called MVO-clustering algorithm. The algorithm was tested on benchmark data and results are compared with the benchmark k-means clustering algorithm as well as PSO-clustering algorithms, Gravitational search clustering. The experimental MVO-clustering results are better than k-means clustering, PSO-clustering and GSA clustering. The idea presented in this work possess new research directions by introducing the idea of exchanging universe's object or movement of cluster centroid through white/black hole and worm hole tunnels which can lead to find the optimal cluster centroid of the cluster problem. The exploration and exploitation phases

Some of the issues which need to be addressed in order to enhance the performance and generalization of the algorithm are as follows.

- Selection of the number of agents
- Initialization of the Universe
- Attribute evolution during exchange of objects.
- The criteria for maturity (strength) of the Universe.

In the future we would like to address some of the above issues in detail specially a complete implementation model of MVO for some specified data mining domain will be explored. Future studies will extend the fitness function to also explicitly optimize the inter- and intra-cluster distances. More elaborate tests on higher dimensional problems and large number of patterns will be done. The MVO clustering algorithms will also be extended to dynamically determine the optimal number of clusters.

References

[1] IE Evangelou, DG Hadjimitsis, AA Lazakidou, C Clayton,“Data Mining and Knowledge Discovery in Complex Image Data using Artificial Neural Networks”, Workshop on Complex Reasoning a Geographical Data, Cyprus, 2001.

[2] T Lillesand, R Keifer, “Remote Sensing and Image Interpretation”, John Wiley & Sons, 1994.

[3] E Forgy, “Cluster Analysis of Multivariate Data: Efficiency versus Interpretability of Classification”, Biometrics, Vol. 21, pp 768–769, 1965.

[4] JA Hartigan, “Clustering Algorithms”, John Wiley & Sons, New York, 1975.

[5] Danielsson, Per-Erik. "Euclidean distance mapping." Computer Graphics and image processing 14.3 (1980): 227-248.

[6] John H (1992) Holland, adaptation in natural and artificial systems. MIT Press, Cambridge

[7] Kennedy J, Eberhart R (1995) Particle swarm optimization. In:Proceedings of IEEE international conference on neural networks, pp 1942–1948

[8] Van der Merwe, D. W., and Andries Petrus Engelbrecht. "Data clustering using particle swarm optimization." Evolutionary Computation, 2003. CEC'03. The 2003 Congress on. Vol. 1. IEEE, 2003.

[9] Maulik, Ujjwal, and Sanghamitra Bandyopadhyay. "Genetic algorithm-based clustering technique." Pattern recognition 33.9 (2000): 1455-1465.

[10] Karaboga, Dervis, and Celal Ozturk. "A novel clustering approach: Artificial Bee Colony (ABC) algorithm." Applied soft computing 11.1 (2011): 652-657.

[11] Mirjalili, Seyedali, Seyed Mohammad Mirjalili, and Abdolreza Hatamlou. "Multi-verse optimizer: a nature-inspired algorithm for global optimization."Neural Computing and Applications 27.2 (2016): 495-513.

[12] L. Davis (Ed.), Handbook of Genetic Algorithms, Van Nostrand Reinhold, New York, 1991.

[13] J Kennedy, RC Eberhart, Y Shi, “Swarm Intelligence”, Morgan Kaufmann, 2002.

[14] M Omran, A Salman, AP Engelbrecht, “Image Classification using Particle Swarm Optimization”, Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning, Singapore, 2002.

[15] Piotrowski, Adam P., Jaroslaw J. Napiorkowski, and Pawel M. Rowinski. "How novel is the “novel” black hole optimization approach?." Information Sciences 267 (2014): 191-200.

[16] Hatamlou, Abdolreza. "Black hole: A new heuristic optimization approach for data clustering." Information sciences 222 (2013): 175-184.

[17] Dorigo, Marco, Mauro Birattari, and Thomas Stutzle. "Ant colony optimization." IEEE computational intelligence magazine 1.4 (2006): 28-39.

[18] Shelokar, P. S., Valadi K. Jayaraman, and Bhaskar D. Kulkarni. "An ant colony approach for clustering." Analytica Chimica Acta 509.2 (2004): 187-195.

[19] Rashedi, Esmat, Hossein Nezamabadi-Pour, and Saeid Saryazdi. "GSA: a gravitational search algorithm." Information sciences 179.13 (2009): 2232-2248.

[20] D. Holliday, R. Resnick, J. Walker, Fundamentals of physics, John Wiley and Sons, 1993.

[21] Hatamlou, Abdolreza, Salwani Abdullah, and Hossein Nezamabadi-Pour. "A combined approach for clustering based on K-means and gravitational search algorithms." Swarm and Evolutionary Computation 6 (2012): 47-52.

[22] D. Karaboga, An idea based on honey bee swarm for numerical optimization, Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.

[23] Zhang, Changsheng, Dantong Ouyang, and Jiaxu Ning. "An artificial bee colony approach for clustering." Expert Systems with Applications 37.7 (2010): 4761-4767.

[24] Alam, Shafiq, Gillian Dobbie, and Patricia Riddle. "An evolutionary particle swarm optimization algorithm for data clustering." Swarm Intelligence Symposium, 2008. SIS 2008. IEEE. IEEE, 2008.

[25] Khoury J, Ovrut BA, Seiberg N, Steinhardt PJ, Turok N (2002) From big crunch to big bang. Phys Rev D 65:086007

[26] Steinhardt PJ, Turok N (2002) A cyclic model of the universe. Science 296:1436–1439

[27] Davies PC (1978) Thermodynamics of black holes. Rep Prog Phys 41:1313

[28] Guth AH (2007) Eternal inflation and its implications. J Phys A Math Theor 40:6811

[29] Steinhardt PJ, Turok N (2005) The cyclic model simplified. New Astron Rev 49:43 57

[30] C.J. Merz, C.L. Blake, UCI Repository of Machine Learning Databases. <http://www.ics.uci.edu/-mlearn/MLRepository.html>.

[31] R.A. Fisher, The use of multiple measurements in taxonomic problems, Ann. Eugenics 3 (1936) 179}188.