Unsupervised Data Classification Using Jaya Evolutionary Algorithm

A Project Report Submitted
in Partial Fulfillment of the Requirements
for the Degree of
Bachelor of Technology
in
Computer Science & Engineering

by

Anil Balchandani (20144055)
Abhinandan Agarwal (20144091)
Jitesh (20144123)
Dheerendra Kumar (20144133)
Manoj Kumar (20144155)

to the

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
MOTILAL NEHRU NATIONAL INSTITUTE OF TECHNOLOGY
ALLAHABAD,UP,INDIA(211004)
April, 2018

UNDERTAKING

We declare that the work presented in this report titled "***Unsupervised Data Classification Using Jaya Evolutionary Algorithm***", submitted to the Computer Science and Engineering Department, Motilal Nehru National Institute of Technology Allahabad, for the award of the ***Bachelor of Technology*** degree in ***Computer Science & Engineering***, is our original work. I have not plagiarized or submitted the same work for the award of any other degree. In case this undertaking is found incorrect, I accept that my degree may be unconditionally withdrawn.

April, 2018
Allahabad

Anil Balchandani (20144055)
Abhinandan Agarwal (20144091)
Jitesh (20144123)
Dheerendra Kumar (20144133)
Manoj Kumar (20144155)

CERTIFICATE

Certified that the work contained in the report titled "***Unsupervised Data Classification Using Jaya Evolutionary Algorithm***", by *ANIL BALCHANDANI(20144055), ABHINANDAN AGARWAL(20144091), JITESH(20144123), DHEERENDRA KUMAR(20144133), MANOJ KUMAR(20144155)* has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

(Dr. Anoj Kumar)
Computer Science and Engineering Dept.
M.N.N.I.T Allahabad

April, 2018

Preface

Data Classification has an important roots in Computer Science and Engineering problems. We have attempt to solve an clustering problem using by optimizing multiple objectives such as set of cluster validity indices concurrently. The proposed clustering technique uses the most recent optimization algorithm Jaya as an underlying optimization stratagem [5].

Also, we would like to thank the director of our institute **Prof. Rajeev Tripathi** for his contribution and support towards the students of this institute and also we would like to thank our head of the department(HOD) **Prof. Neeraj Tyagi** who has shown a kind of interest in such activities for the development of students. In the end we are thankful to all those who have directly or indirectly helped and directed us at every possible step.

For this problem we have compared obtained results with that of standard K-means algorithm. We have applied evolutionary Jaya algorithm for optimization to obtain better results and improving overall efficiency [1].

Acknowledgements

We would like to express our deepest appreciation to all those who provided us the possibility to complete this report. A special gratitude we give to our project mentor, **Dr.Anoj Kumar** whose contribution in stimulating suggestions and encouragement, helped us to coordinate our project. Then we would like to thank all our friends, colleagues for their support and motivation. Last and the most important, we would like to thank Almighty God for his constant blessings and love.

Contents

Chapter 1

Abstract

One of fundamental challenges of clustering is how to evaluate results, without auxiliary information. A common approach for evaluation of clustering results is to use validity indexes. Clustering validity approaches can use three criterias: External criteria (evaluate the result with respect to a pre-specified structure), internal criteria (evaluate the result with respect a information intrinsic to the data alone). Consequently, different types of indexes are used to solve different types of problems and indexes selection depends on the kind of available information [2].

The proposed clustering technique uses the most recent optimization algorithm Jaya as an underlying optimization stratagem. This evolutionary technique always aims to attain global best solution rather than a local best solution in larger datasets. The explorations and exploitations imposed on the proposed work results to detect the number of clusters, appropriate partitioning present in data sets and mere optimal values towards CVIs frontiers.

Chapter 2

Introduction

The purpose of clustering is to determine the intrinsic grouping in a set of unlabeled data, where the objects in each group are indistinguishable under some criterion of similarity. Clustering is an unsupervised classification process fundamental to data mining. It has applications in several fields like bioinformatics, web data analysis, text mining and scientific data exploration [2]. However, to get good results, the clustering algorithm depends on input parameters. For instance, k-means and CURE algorithms require a number of clusters (k) to be created. In this sense, the question is: What is the optimal number of clusters?. Currently, cluster validity indexes research has drawn attention as a means to give a solution. Many different cluster validity methods have been proposed without any a priori class information.

Majority of optimization problems demands improvement issues with multiple objectives and are attracted towards evolutionary computation methodologies due their simplicity of transformative calculation. We provide clustering algorithms underplayed with Jaya evolutionary algorithm to solve large set of objectives, for affricating factual k determination, that are interesting, suitable detachment prompted in data sets, and optimizing a set of cluster validity indices (CVIs) simultaneously for encouraging most favourable convergence at final solutions.

2.1 Motivation

Data classification is one of the most important technique in data mining. It has applications in several fields like bioinformatics, web data analysis, text mining and scientific data exploration. Clustering refers to unsupervised learning and for that reason it has no a priori data set information. Moreover we are more concerned about optimizing multi-objective functions because in reality we have to deal with so many criterion to be fulfilled at the same time.

Also choosing a suitable optimizing algorithm which could give us better results without much intervention of its own parameters.

2.1.1 Software Development Model Used

We have used Agile Software model for our project. First of all we have collected all data sets over which our algorithm would run. We have developed each component and test its feasibility. Then all the components are combined to obtain a working model. We have used Python as our language for coding purpose and different plotting tools such MAT plotter and online plotter have been used for this project.

Chapter 3

Overview

3.1 Clustering and Related Work

Data clustering is recognized as the most prominent unsupervised technique in machine learning. This technique apportions a given dataset into homogeneous groups in view of some likeness/disparity metric. Conventional clustering algorithms regularly make previous assumptions about grouping a cluster structure, and adoptable with a suitable objective function so that it can be optimized with classical or meta heuristic techniques. These estimations grade inadequately when clustering presumptions are not hold in data [2].

Many other algorithms have been proposed such as K-means and CURE but the problem with these algorithms is that of choosing initial centroids and the solutions obtained may be local optimum. But Jaya algorithm which finds the global optimal solution instead of local optimum.

3.2 K-Means Algorithm

k-means 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 apriori. The main idea is to define k centers, one for each cluster. These centers 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 [9]. The next step is to take each point belonging to a given data set and associate it to the nearest center. 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 as barycenter 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 center. A loop has been generated. As a result of this loop we may notice that the k centers change their location step by step until no more changes are done or in other words centers do not move any more. Finally, this algorithm aims at minimizing an objective function know as squared error function given by:

$$J(V) = \sum_{i=1}^{c} \sum_{j=1}^{c_i} (||x_i - v_j||)^2$$

where,

- $||x_i - v_j||$ is the Euclidean distance between x_i and v_j.
- c_i is the number of data points in i^{th} cluster.
- c is the number of cluster centers.

3.3 JAYA Algorithm

JAYA algorithm [8], is another powerful parameter less algorithm for finding optimal solutions. In this algorithm only common control parameters are required like population size and number of generations. The major advantage of the algorithm is at every iteration JAYA algorithm is rapidly moving towards an optimal solution because it removes the worst solutions in every iteration. So JAYA algorithm always tries to improvise the solution to avoid the worst solution [7]. JAYA algorithm has only one phase and it is relatively simpler to use for any specific problems. The working of JAYA is lot different from other algorithms.

First of all it needs to initialize the population size, iteration number which are common control parameters. Then it identifies the best and the worst solution for a given population. The important step of the algorithm is the modification of solutions which uses the below mentioned equation by evaluating the best and worst result for the particular type of objective function and particular type of problem defined for minimization or maximization [6].

$$X^{'}_{j,k,i} = X_{j,k,i} + r_{1,j,i}(X_{j,best,i} - |X_{j,k,i}|) - r_{2,j,i}(X_{j,worst,i} - |X_{j,k,i}|)(1)$$

- $X^{'}_{j,k,i}$ is updated solution of $X_{j,k,i}$ for j^{th} design variable in i^{th} iteration.
- $r_{1,j,i}$ and $r_{2,j,i}$ are random variables.
- $X_{j,best,i}$ is best candidate solution for j^{th} design variable in i^{th} iteration.

Based on the above equation all the solutions of population are updated, and according to set objective function of minimization or maximization the best solution is accepted for the particular iteration and this best solution is taken for further iterations.

Major strength of JAYA algorithm is its procedure to improvise the result by removing the worst result at every iteration.

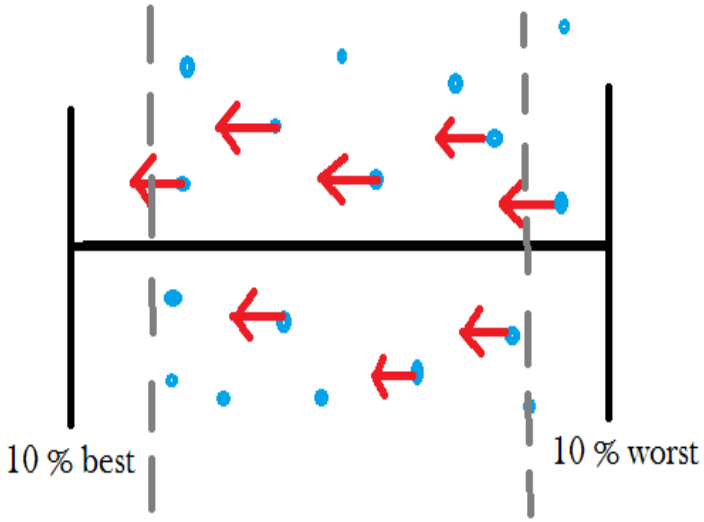


Figure 1: Modified JAYA Model

3.4 Problem Statement

In this project we have attempted to find proper distribution of given datasets into clusters. Clustering so obtained will be validated by Cluster Validity Indices(CVIs) as an multi-objective functions.

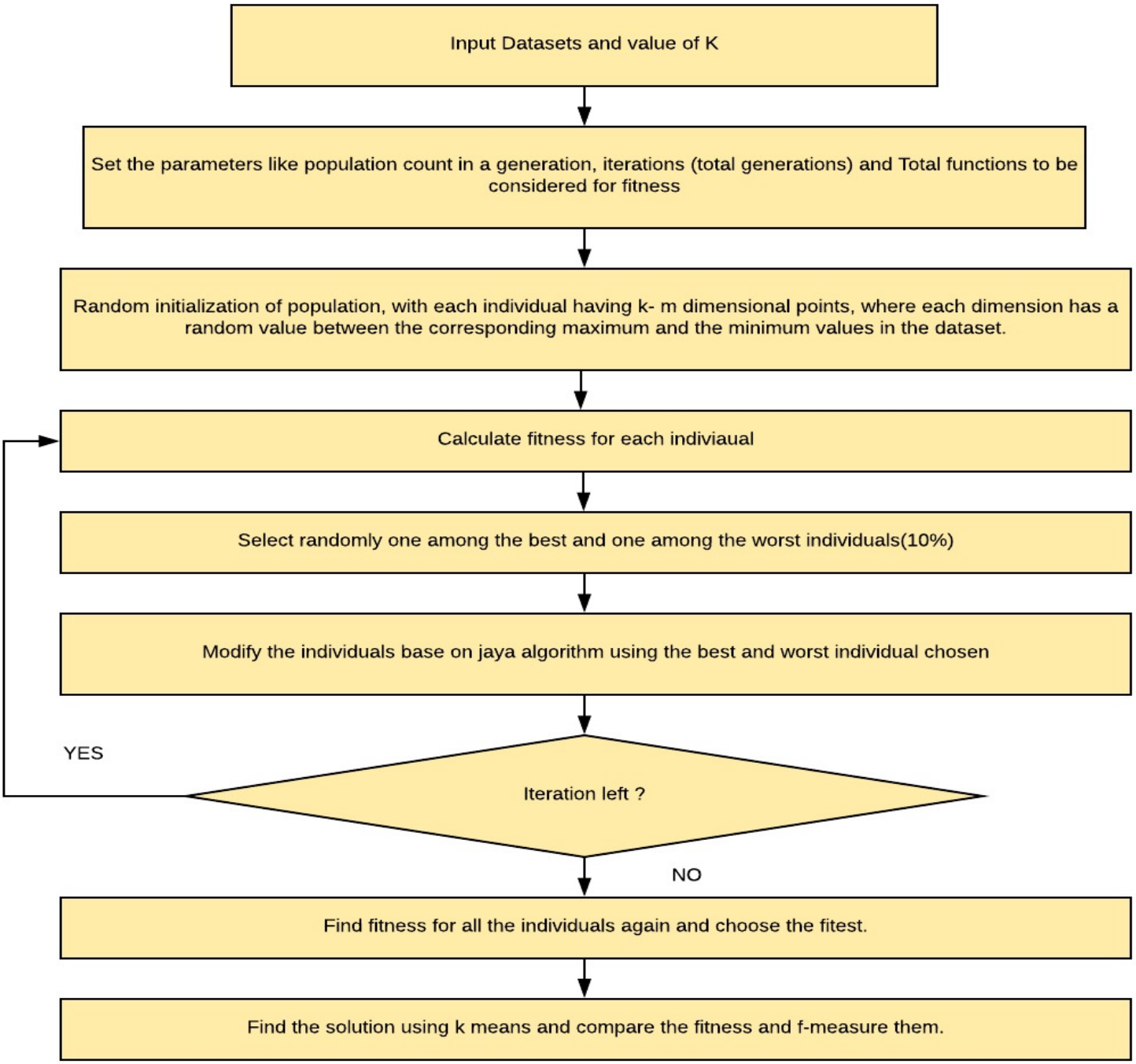
- Given different data sets and their dimensions.
- We have to distribute these set of points in different clusters according to K-means algorithm and with that of our proposed approach.
- Using various cluster validity indices as fitness functions and comparing results of both algorithms.

3.5 Framework

First of all we will build a set of initial population from given data set. Initial centroid will be chosen from these population. Now remaining points will be distributed in the clusters as per CVIs taken. After every iterations solution will be modified and overall fitness will improve.

The algorithm will tested on 12 data sets using various Cluster Validity Indices(CVIs). The results will be compared with K-Means algorithm. Various CVIs that have been used in this paper are as follows:

- Silhouette
- C-Index
- Ball and Hall Index
- H-Index
- CH-Index


Input Datasets and value of K
Set the parameters like population count in a generation, iterations (total generations) and Total functions to be considered for fitness
Random initialization of population, with each individual having k- m dimensional points, where each dimension has a random value between the corresponding maximum and the minimum values in the dataset.
Calculate fitness for each indiviaual
Select randomly one among the best and one among the worst individuals(10%)
Modify the individuals base on jaya algorithm using the best and worst individual chosen
YES
Iteration left ?
NO
Find fitness for all the individuals again and choose the fitest.
Find the solution using k means and compare the fitness and f-measure them.

Chapter 4

Proposed Work

We have applied our approach on twelve real time data sets of varying complexities. Various Cluster Validity Indices(CVIs) such as C-Index, H-Index, CH-Index, Silhouette, Ball and Hall Index are optimized simultaneously to endorse the validity of aimed algorithm. Determinately, the aimed algorithm is able to find proper apportioning in the dataset acco rding to given classes.

4.1 Initialization

To initialize the candidate solutions, the cluster centres are encoded as chromosomes. The population α or number of candidate solutions are initialized randomly with n rows and m columns. The set of solutions are represented as $\alpha_{i,j}(0) = {\alpha_j}^{min} + rand(1) * ({\alpha_j}^{max} - {\alpha_j}^{min})$ and each solution contains K number of selected cluster centers. -

4.2 Objective/Fitness Functions

A straightforward way to pose clustering as an optimization problem is to optimize some CVIs that reflect the goodness of the clustering solutions. The correctness or accuracy of

any optimization method depends on its objective or fitness function being used in the algorithm [2- 3]. In this manner, it is regular to instantaneously advance with numerous of such measures for optimizing distinctive attributes of data. To compute the distance between the centroid and candidate solutions Euclidean distance measure is used, along with it the other objective functions optimized simultaneously are the C-Index, H-Index, CH-Index, Silhouette, Ball and Hall Index CVIs [6].

4.3 Jaya Evolutionary Algorithm

Jaya is a simple, powerful optimization algorithm proposed by R Venkata Rao in 2015 for solving the constrained and unconstrained optimization problems [15]. This algorithm is predicated on the idea that the outcome obtained for a given problem should move towards the best solution and evade the worst solution. This evolutionary approach does not require any particular algorithm specific control parameter, rather mandates common control parameters. The working procedure of this evolutionary method is as follows:

Let $f(\alpha)$ is the objective function to be minimized or maximized. At any iteration,i assume that there are m number of design variables i.e.$(j = 1, 2, ..m)$, n number of candidate solutions (i.e. population size, $k = 1, 2, ..., n$). Let the best candidate best obtains the best value of $f(\alpha)$(i.e. $f(\alpha)_{best}$) in the entire candidate solutions and the worst candidate worst obtains the worst value of $f(\alpha)$(i.e. $f(\alpha)_{worst}$) in the entire candidate solutions. If $\alpha_{j,k,i}$ is the value of j^{th} variable for the k^{th} candidate during the i^{th} iteration, then this values is modified as per the following equation.

$$\alpha^{'}_{j,k,i} = \alpha_{j,k,i} + r_{1,j,i}|(\alpha_{j,best,i}) - |\alpha_{j,k,i}|| - r_{2,j,i}|(\alpha_{j,worst,i}) - |\alpha_{j,k,i}||$$

where $\alpha_{j,best,i}$ is the value of variable j for the best candidate and $\alpha_{j,worst,i}$ is the value of variable j for the worst candidate. $\alpha^{'}_{j,k,i}$is the updated value of $\alpha_{j,k,i}$ and $r_{1,j,i}$ and $r_{2,j,i}$ are the two random variables for the j^{th} variable during the i^{th}iteration in the range[0,1]. The term $r_{1,j,i}|(\alpha_{j,best,i}) - |\alpha_{j,k,i}||$ indicates the tendency of the solution to move closer

to the best solution and the term $r_{2,j,i}|(\alpha_{j,worst,i}) - |\alpha_{j,k,i}||$ indicates the tendency of the solution to avoid the worst solution. $\alpha'_{j,k,i}$ is accepted if it gives better function value. It indicates the tendency of the solution to avoid the worst solution. The steps in Jaya are as follows:

1. Initialize population size, number of design variables and termination condition.
2. Identify best and worst solution in the population.
3. Modify the solutions based on best and worst solutions using Jaya.
4. Is the solution corresponding to $\alpha'_{j,k,i}$ better than the corresponding to $\alpha_{j,k,i}$ then accept and replace the previous solution.
5. Else keep the previous solution.
6. If the termination criterion is satisfied then report as optimum solution.
7. Else go to Step 2.

4.4 Proposed Modified Jaya Algorithm

The working procedure of the aimed algorithm Modified Jaya is as follows:

1. Input data set and the value of k.
2. Set the parameters like population count in a generation, iterations (total generations) and total functions to be considered for fitness.
   (a) For each feature to be provided,calculate a random floating value between the max and minimum values of the particular feature.
3. Calculate fitness for each individual using the combination of error functions.

4. Modify the individuals based on Jaya algorithm using the best and the worst individual chosen.

5. If iterations are left, goto step 4.

6. Find fitness for all the individuals again and choose the fittest.

7. Find the solution using k means and compare the fitness and calculate F-score for them.

4.5 Fitness Functions Used

4.5.1 Ball and Hall Index

In this we calculate two terms Sum of Square within the cluster(SSW) and Sum of Square between the clusters(SSB).

$$SSW = \sum_{j=1}^{m} \sum_{i \epsilon j} (x_{ij} - \bar{x_j})^2$$

$$SSB = \sum_{j=1}^{m} n_j (\bar{x_j} - \bar{\bar{x}})^2$$

therefore

$$BallandHallindex = \frac{SSW}{m}$$

$$CH - Index = \frac{SSB}{SSW} * \frac{n-m}{m-1}$$

$$H - Index = -\log \frac{SSW}{SSB}$$

For better clustering CH-Index and H-Index should be maximum and Ball and Hall Index should be minimum where,

- x_{ij} is the i^{th} data point in j^{th} cluster.
- $\bar{x_j}$ is the mean of j^{th} cluster.
- $\bar{\bar{x}}$ is mean of all data points.
- there are total m clusters.
- n is the total number of points in data set.

4.5.2 C- Index

C-Index is defined below:

$$C = \frac{S - S_{min}}{S_{max} - S_{min}}$$

where,

- S is sum of all distances between pairs of observations in the same cluster over all clusters.
- Let n be number of these pairs. S_{min} and S_{max} are the sum of n lowest and highest distances across all pairs of observations.
- Ideally its value should be 0.

4.5.3 Silhouette

It defined as below:

$$S(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}}$$

- $a(i)$ is measure of how well i is assigned to its cluster, a(i) is the average distance between i and all other data within the same cluster.
- $b(i)$ be the lowest average distance of i to all points in any cluster of which i is not a member.

- Its value lies between -1 to 1. For better clustering it should be near to 1.

4.6 Combination of all Objective functions

Since we have used many objective/fitness functions for checking the validity of clusters formed. We can combine all the objective functions into one.

$$F = w1 \times f1 + w2 \times f2 + w3 \times f3 + w4 \times f4 + ...$$

where,

- wi is the weight of i^{th} fitness function.
- fi is the fitness value of i^{th} fitness function.
- F should be maximum for better clustering.

Algorithm 1: Modified JAYA Algorithm for Clustering Problem

S1 *Require*
Input data set and the value of k.

S2 *Initialise*

Set the parameters like population count in a generation, iterations (total generations) and total functions to be considered for fitness.
$NDV \leftarrow Number\_of\_Design\_Variables$
$TER\_COD \leftarrow Termination\_Condition$

S3 For each feature to be provided, calculate a random floating value between the maximum and minimum values of the particular feature.

S4 Calculate fitness for each individual using the combination of error functions for both k-means and proposed algorithm.

S5 Modify Solution

$$X^{'} = X + rand(X_{top} - |X|) - rand(X_{bottom} - |X|)$$

....towards top 10% and away from the bottom 10%.

S6
if $X^{'} \geq X$ **then**
Update the previous solution
else
No updates in previous solution.
end if
S7
if Number of iteration are left **then**
goto **S4**
else
continue
end if

S8 Display the optimum results for both K-means and proposed algorithm.
S9 Calculate F-score for both algorithms.

Chapter 5

Experimental Setup and Results Analysis

5.1 Dataset Used

In this section, we report on experiments that use multi-objective clustering to identify partitions in diverged set of data sets. We have run our algorithm for each data set upto 1000 iterations and compared over results to that of K-means algorithm[3][4]. Also the percentage of improvement is observed. Fitness Values are shown in Table 5.1

The data sets that have been used are:

- 2-D dataset
- 3-D dataset
- Iris
- Wine
- Glass
- Ionosphere

- Sonar
- Rippleset
- Parkinsons
- Segmentation
- Weighting

5.2 Tools and System Used

All experiment will be run on a machine with 2.4 GHz with 8GB of RAM. We have implemented over algorithm in Python and used MAT plotter tool to plot graphs.

We have used different available CVIs functions to compare our results with that of standard algorithm (K-means).

5.3 F-Measure Comparison

The F measure (F1 score or F score) is a measure of a test's accuracy and is defined as the weighted harmonic mean of the precision and recall of the test. It considers both the precision p and the recall r of the test to compute the score: p is the number of correct positive results divided by the number of all positive results returned by the classifier, and r is the number of correct positive results divided by the number of all relevant samples (all samples that should have been identified as positive). The F1 score is the harmonic average of the precision and recall, where an F1 score reaches its best value at 1 (perfect precision and recall) and worst at 0.

$$F = 2 \times \frac{precision \times recall}{precision + recall}$$

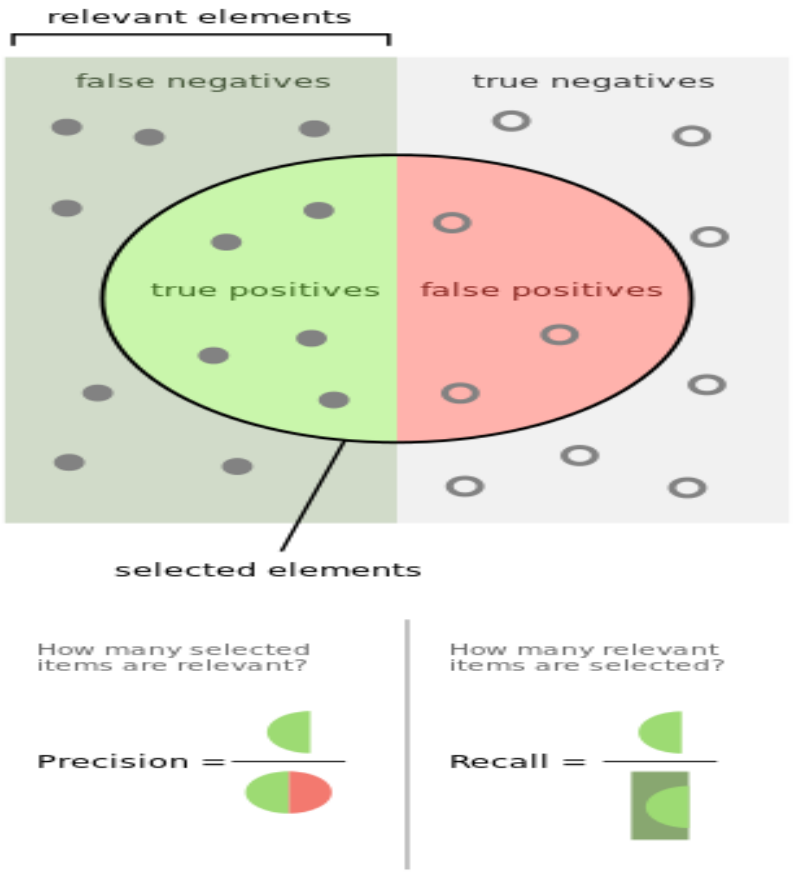


Figure 2: Precision and Recall

Table 1: F score for both algorithms

F-Measure		
Data set	F Score for K-Meas	F Score for Jaya Algorithm
Iris	0.9492	0.9398
Ionosphere	0.6546	0.7625
Parkinsons	0.7466	0.8017
Rippleset	0.7898	0.8142
Wine	0.6593	0.7421
Sonar	0.4746	0.5042
Glass	0.4538	0.4682
Segmentation	0.4930	0.4952

We have calculated F measure for all data sets for both K-means and Jaya algorithm and results are shown below in table 1.

5.4 Results

Given below tables shows the fitness value obtained by two algorithms and their comparison is being shown. Fitness value is being obtained by combining so many CVIs(Silhouette, C-index, H-Index etc.) as a weighted sum.

Graphical representation of data points for 2-D and 3-D data sets have been shown after tables.

Table 2: Comparison of Fitness Value Iteration wise

For 2-D dataset (100*2,3)			
No of iterations	Fitness Value for K-means	Fitness Value for Modified Jaya	Percent of Improvement
10	15.25	10.22	-32.98
20	18.56	14.56	-27.06
30	19.92	14.66	-26.40
40	20.23	18.12	-10.43
50	22.21	21.56	-2.92
60	22.14	23.12	4.42
70	23.45	25.69	9.55
80	28.65	30.24	5.54
90	29.44	33.45	13.62
100	31.39	35.56	14.28
150	31.39	36.45	16.11
200	31.39	39.45	25.67
500	32.00	42.28	32.12
1000	32.10	44.23	37.78

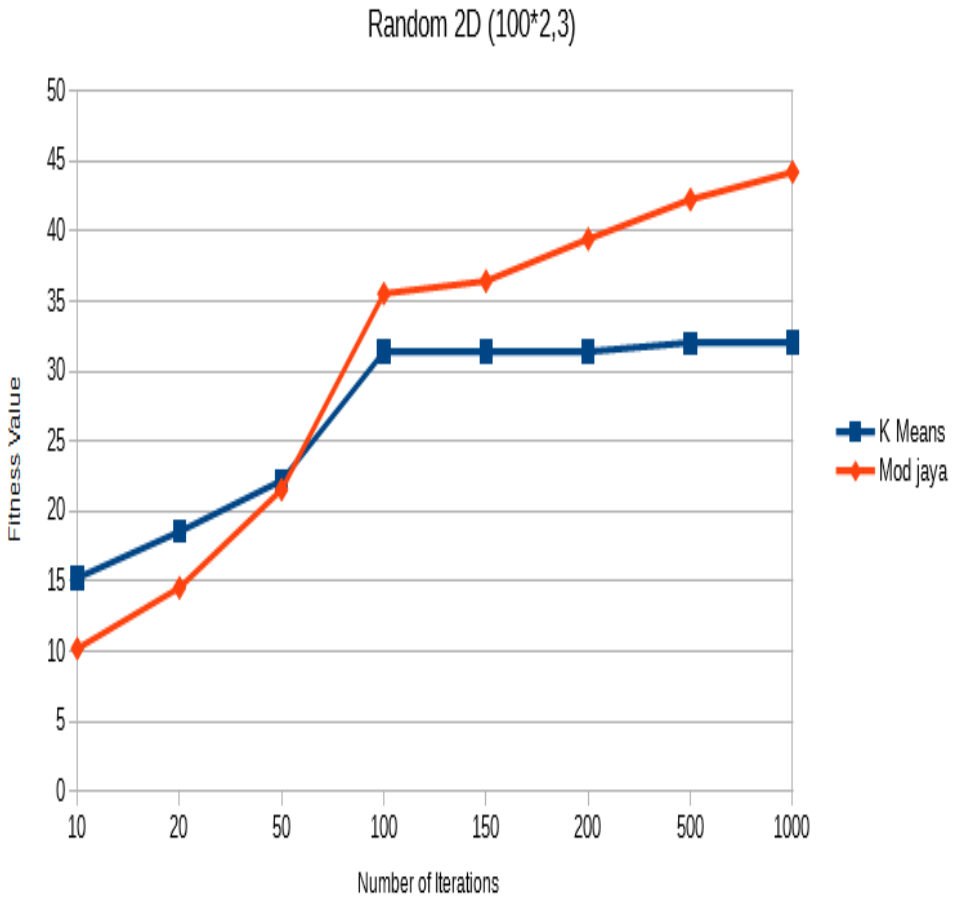


Figure 3: Graph for fitness value vs No. of Iterations

Table 3: Comparison of Fitness Value Iteration wise

For 3-D dataset (50*3,4)			
No of iterations	Fitness Value for K-means	Fitness Value for Modified Jaya	Percent of Improvement
10	10.52	8.56	-18.63
20	12.65	11.23	-12.64
30	17.56	17.25	-1.76
40	20.24	21.98	8.59
50	24.63	25.64	4.10
60	29.84	31.25	4.72
70	31.35	34.565	10.25
80	32.45	35.69	10.98
90	33.25	38.98	17.23
100	36.25	39.98	10.28
150	36.25	40.22	10.95
200	36.25	40.22	10.95
500	36.25	40.22	10.95
1000	36.25	40.22	10.95

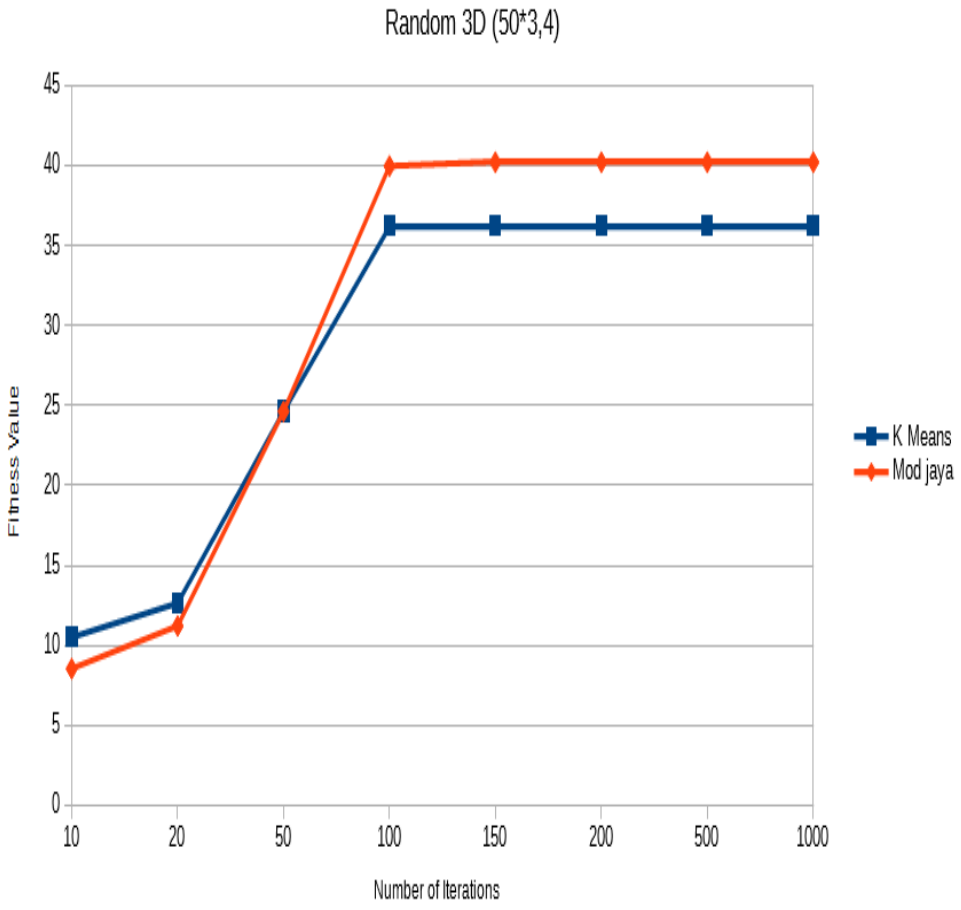


Figure 4: Graph for fitness value vs No. of Iterations

Table 4: Comparison of Fitness Value Iteration wise

For Iris (150*4,3)			
No of iterations	Fitness Value for K-means	Fitness Value for Modified Jaya	Percent of Improvement
10	-52.20	-55.25	-5.8
20	-47.5	-49.25	-3.68
30	-42.67	-44.33	-3.89
40	-33.21	-31.29	5.78
50	-31.89	-30.84	3.29
60	-30.02	-29.12	2.99
70	-26.89	-24.76	7.92
80	-22.77	-20.11	11.68
90	-22.6	-18.45	18.36
100	-19.9	-16.69	16.13
150	-19.07	-15.98	16.20
200	-18.69	-14.55	22.15
500	-18.62	-13.21	29.05
1000	-18.62	-13.19	29.07

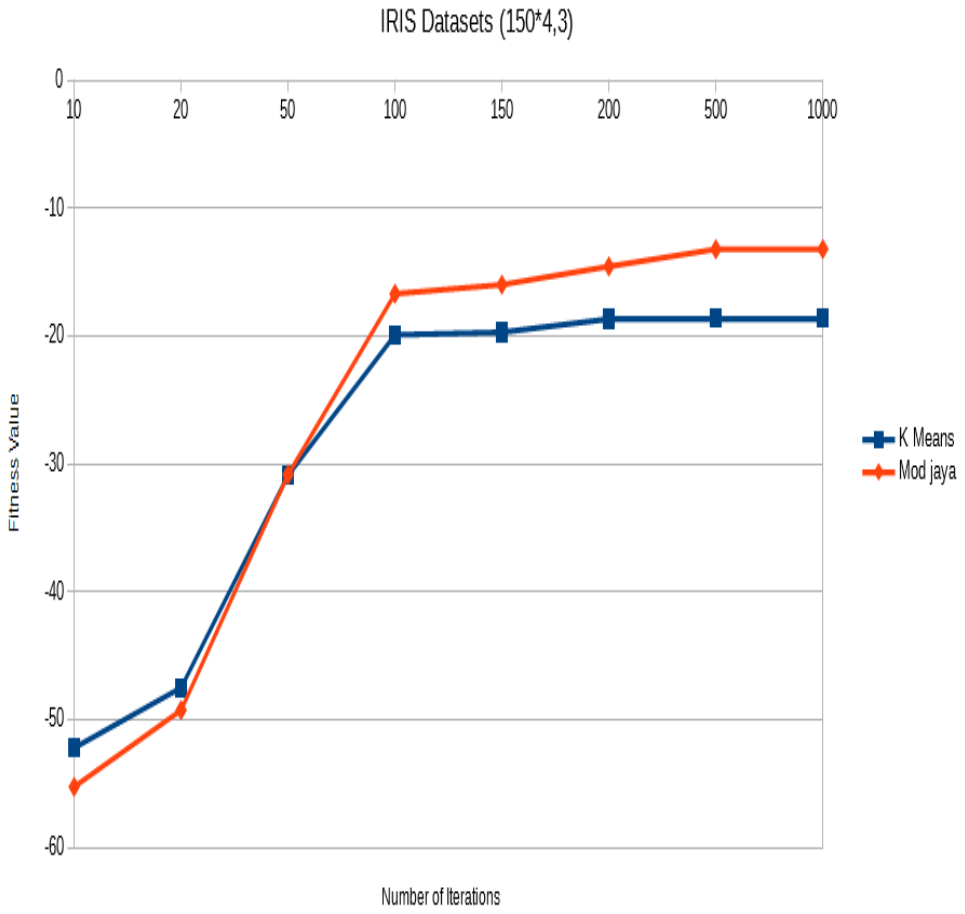


Figure 5: Graph for fitness value vs No. of Iterations

Table 5: Comparison of Fitness Value Iteration wise

For Wine (178*13,3)			
No of iterations	Fitness Value for K-means	Fitness Value for Modified Jaya	Percent of Improvement
10	242.24	200.35	-17.29
20	250.22	222.23	-11.18
30	257.43	255.35	-0.08
40	265.09	269.87	1.80
50	265.98	275.14	3.44
60	272.00	280.32	3.05
70	272.87	285.45	4.61
80	273.45	288.89	5.64
90	276.25	291.25	5.69
100	284.20	289.63	1.91
150	285.11	295.14	3.50
200	287.09	299.98	4.49
500	287.29	300.25	4.52
1000	287.38	300.29	4.53

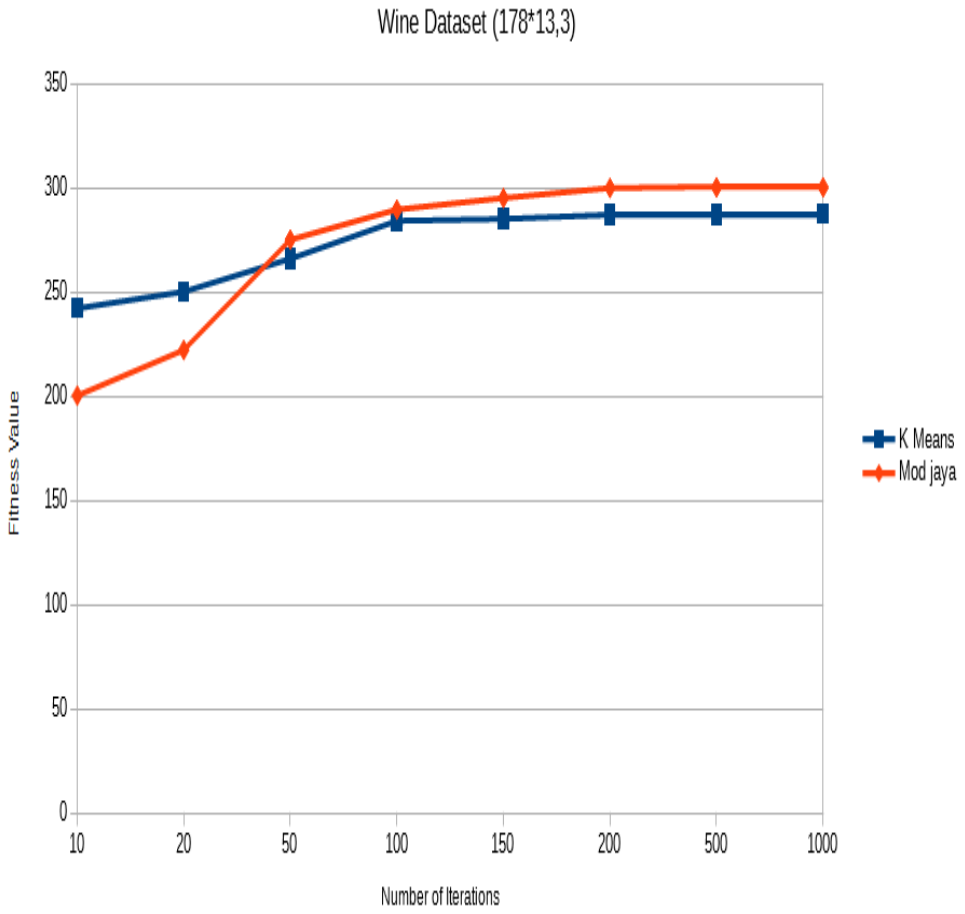


Figure 6: Graph for fitness value vs No. of Iterations

Table 6: Comparison of Fitness Value Iteration wise

For Glass dataset (214*9,6)			
No of iterations	Fitness Value for K-means	Fitness Value for Modified Jaya	Percent of Improvement
10	-15.66	-19.25	-22.92
20	-10.55	-18.45	-74.88
30	-8.33	-14.56	-74.78
40	-6.99	-10.21	-46.06
50	-6.77	-9.87	-45.79
60	-5.59	-8.85	-58.31
70	-5.50	-7.54	-37.09
80	-5.48	-6.65	-21.35
90	-5.45	-5.55	-1.83
100	-5.44	-5.55	-2.02
150	-5.43	-5.56	-2.39
200	-5.40	-5.89	-9.07
500	-5.38	-5.75	-6.87
1000	-5.37	-5.38	-0.18

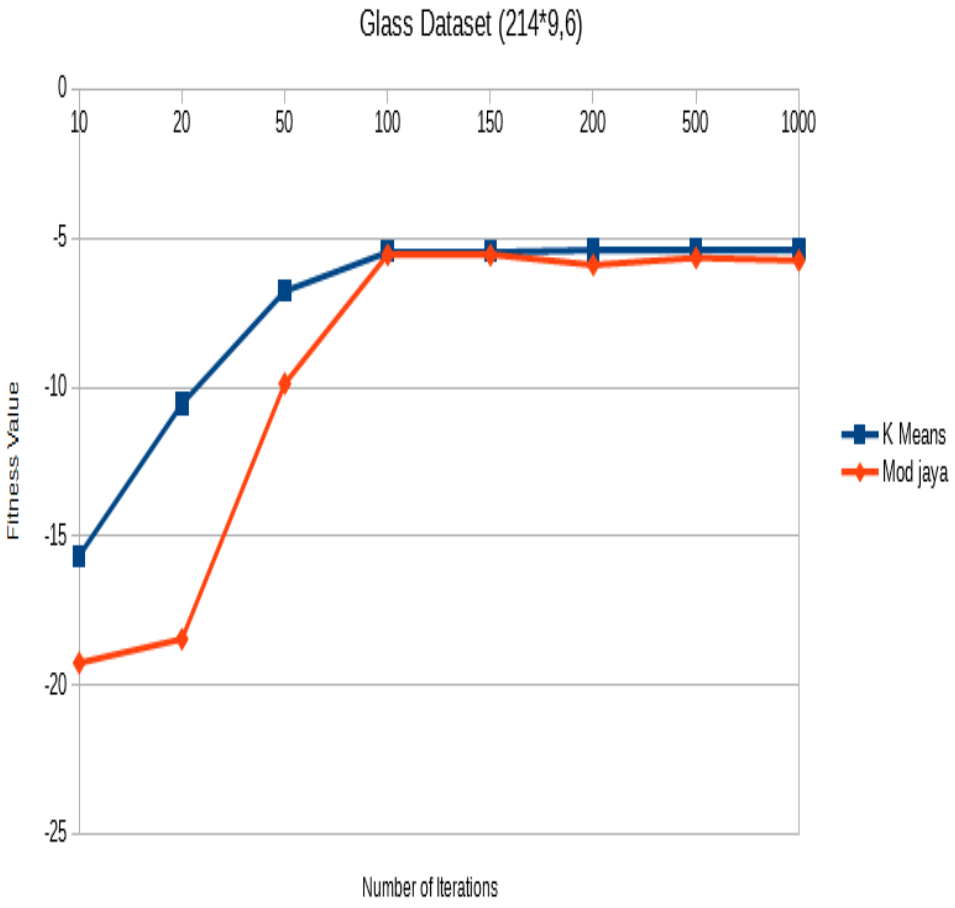


Figure 7: Graph for fitness value vs No. of Iterations

Table 7: Comparison of Fitness Value Iteration wise

For Sonar dataset (204*60,2)			
No of iterations	Fitness Value for K-means	Fitness Value for Modified Jaya	Percent of Improvement
10	4.66	2.05	-56.00
20	4.95	2.45	-50.50
30	5.03	4.26	-15.33
40	5.09	4.55	-7.73
50	5.17	4.77	-5.79
60	5.19	4.85	-6.55
70	5.22	5.04	-3.44
80	5.24	5.25	0.091
90	5.25	5.55	5.71
100	5.34	6.00	12.35
150	5.40	6.56	21.48
200	5.45	6.89	26.42
500	5.46	8.00	46.52
1000	5.46	8.09	48.25

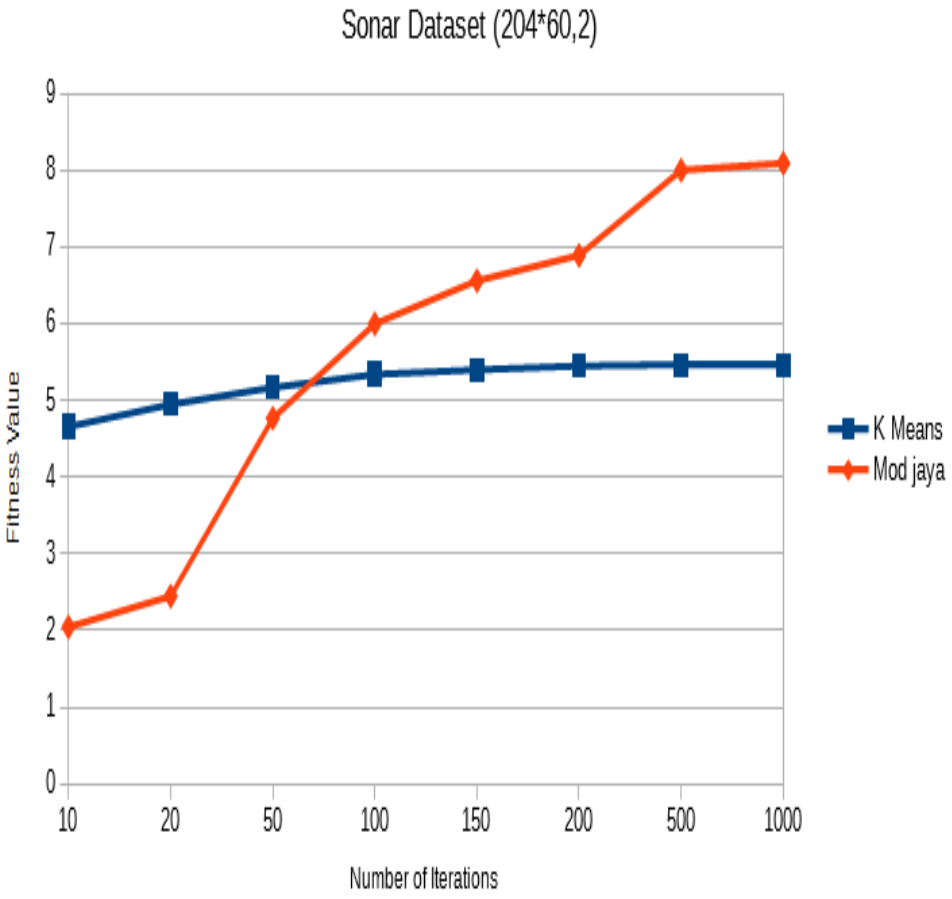


Figure 8: Graph for fitness value vs No. of Iterations

Table 8: Comparison of Fitness Value Iteration wise

For Ionosphere dataset (351*34,2)			
No of iterations	Fitness Value for K-means	Fitness Value for Modified Jaya	Percent of Improvement
10	-45.38	-50.25	-10.73
20	-39.09	-42.35	-8.33
30	-36.09	-35.56	0.014
40	-29.44	-28.59	2.88
50	-23.11	-21.33	7.70
60	-20.98	-19.54	6.86
70	-19.89	-17.42	12.41
80	-16.98	-15.31	9.83
90	-14.56	-12.27	15.72
100	-12.99	-10.66	17.93
150	-12.45	-8.95	28.11
200	-12.23	-8.01	34.50
500	-12.00	-7.25	39.58
1000	-11.96	-7.11	40.55

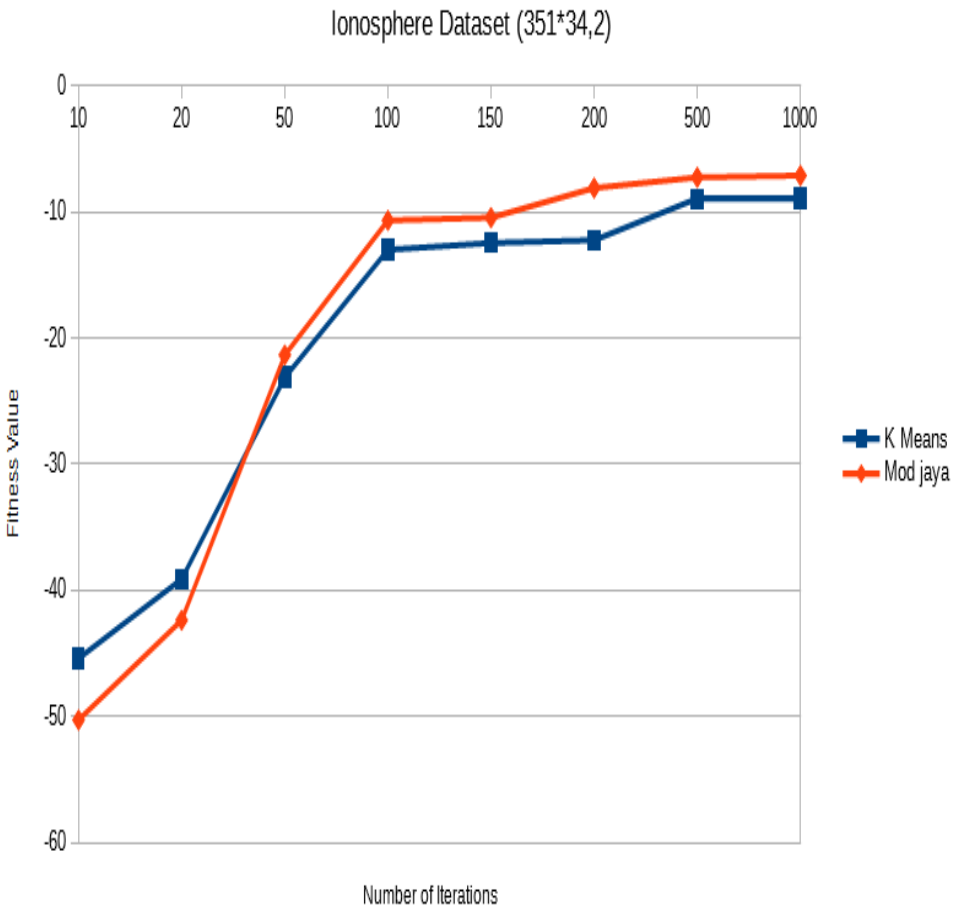


Figure 9: Graph for fitness value vs No. of Iterations

Table 9: Comparison of Fitness Value Iteration wise

For Parkinsons dataset (195*22,2)			
No of iterations	Fitness Value for K-means	Fitness Value for Modified Jaya	Percent of Improvement
10	320.15	289.21	-9.66
20	322.11	295.32	-8.31
30	327.87	315.48	-3.77
40	329.85	324.69	-1.56
50	333.33	335.11	0.53
60	335.43	339.45	1.22
70	338.11	344.12	1.77
80	338.76	345.21	2.07
90	340.01	350.22	2.94
100	340.34	352.14	3.52
150	341.01	353.69	3.81
200	341.80	355.23	4.10
500	341.84	356.21	4.39
1000	341.84	357.14	4.69

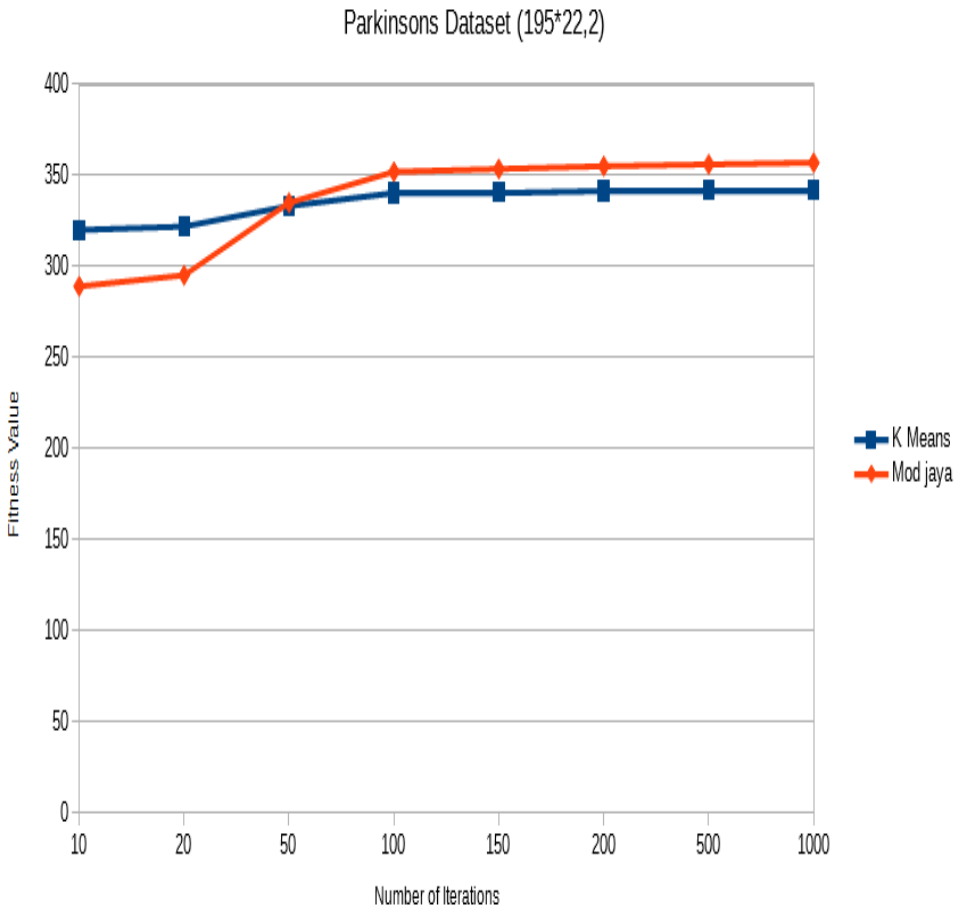


Figure 10: Graph for fitness value vs No. of Iterations

Table 10: Comparison of Fitness Value Iteration wise

For Segmentation dataset (214*19,7)			
No of iterations	Fitness Value for K-means	Fitness Value for Modified Jaya	Percent of Improvement
10	22.65	20.25	-10.50
20	27.45	25.69	-6.41
30	27.98	27.00	-3.50
40	28.65	29.12	1.64
50	29.99	32.56	8.56
60	31.00	34.15	10.16
70	31.67	36.69	16.12
80	32.12	37.89	17.18
90	33.27	39.65	18.18
100	38.52	40.23	5.26
150	38.57	41.25	7.89
200	38.61	42.21	10.52
500	38.64	42.33	10.65
1000	38.64	45.25	18.42

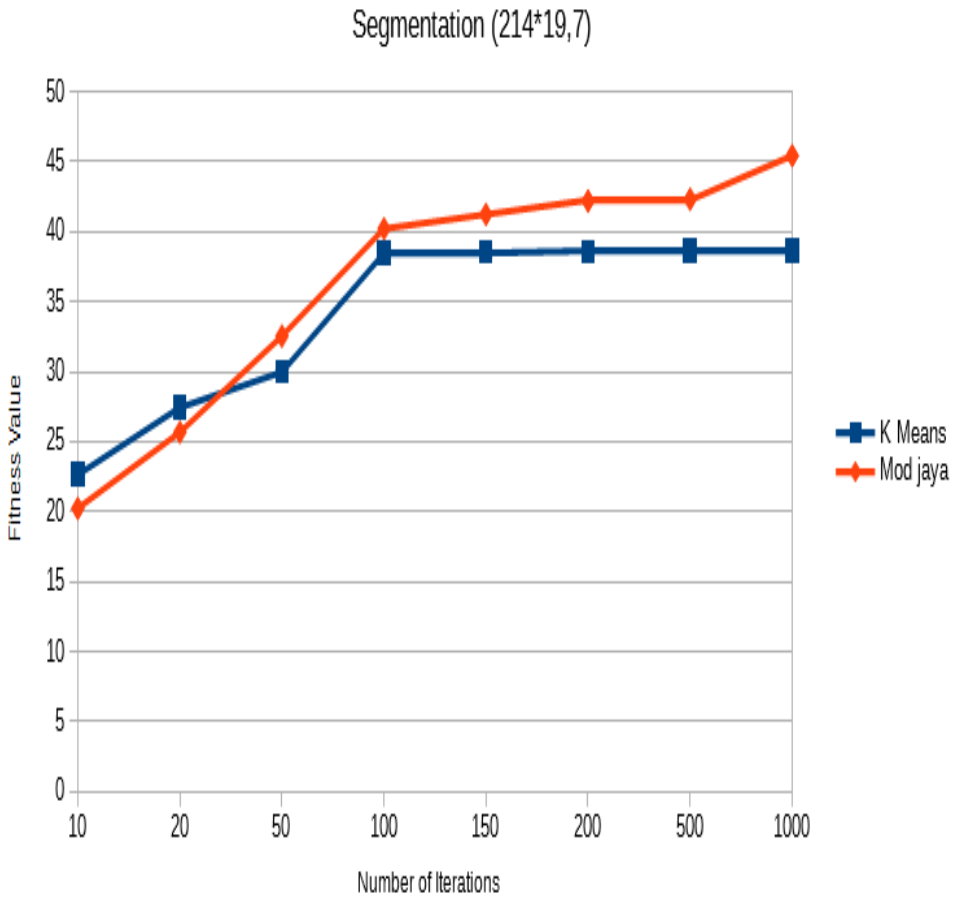


Figure 11: Graph for fitness value vs No. of Iterations

Table 11: Comparison of Fitness Value Iteration wise

For Ripple Set (250*3,2)			
No of iterations	Fitness Value for K-means	Fitness Value for Modified Jaya	Percent of Improvement
10	26.25	24.12	-8.11
20	28.45	25.22	-11.35
30	31.11	30.17	-3.24
40	34.98	33.14	-5.26
50	38.14	35.64	-6.55
60	44.36	42.18	-4.91
70	47.98	48.99	2.00
80	48.95	50.22	2.59
90	51.76	53.74	3.92
100	54.23	57.69	5.55
150	55.89	60.46	9.09
200	59.14	63.74	7.79
500	62.21	68.47	9.37
1000	63.95	72.25	14.28

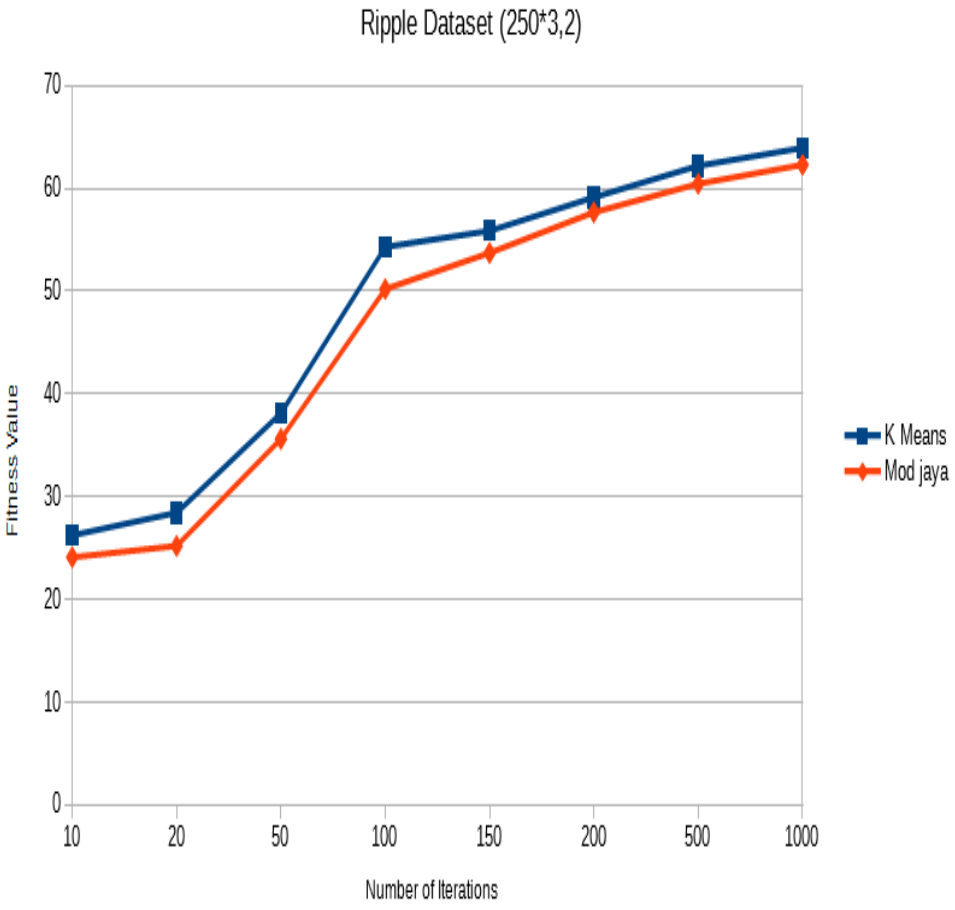


Figure 12: Graph for fitness value vs No. of Iterations

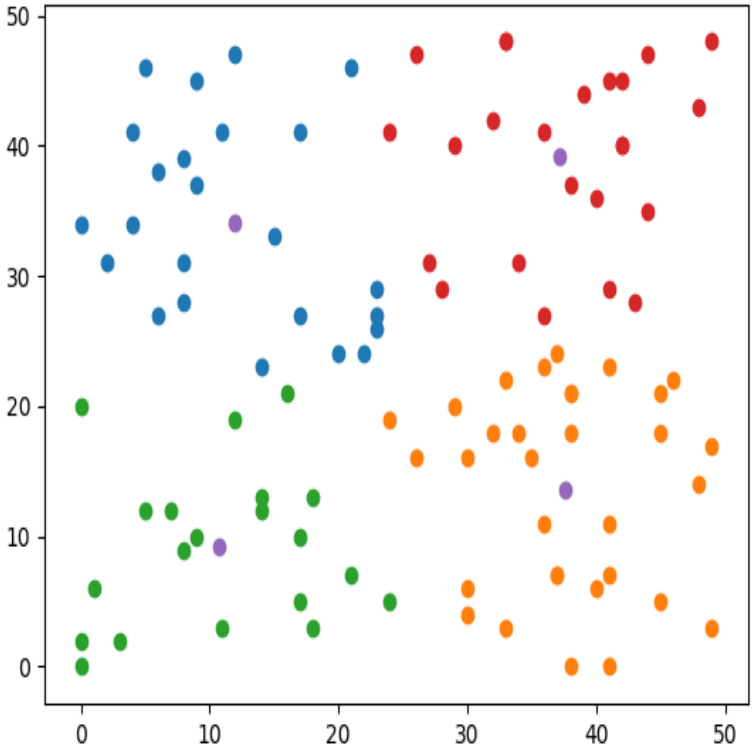


Figure 13: Clustering for 2-D data set by K-means algorithm

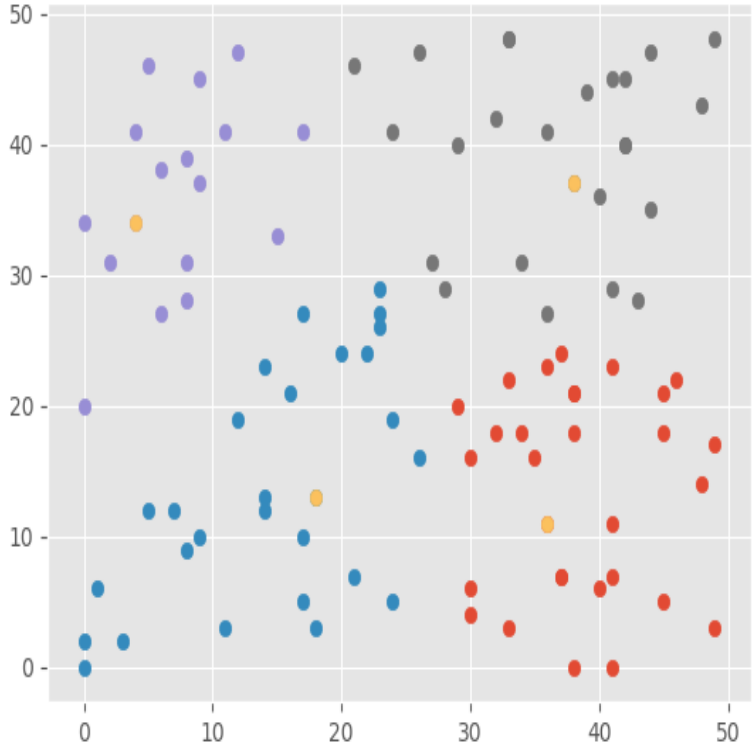


Figure 14: Clustering for 2-D data set by Jaya algorithm

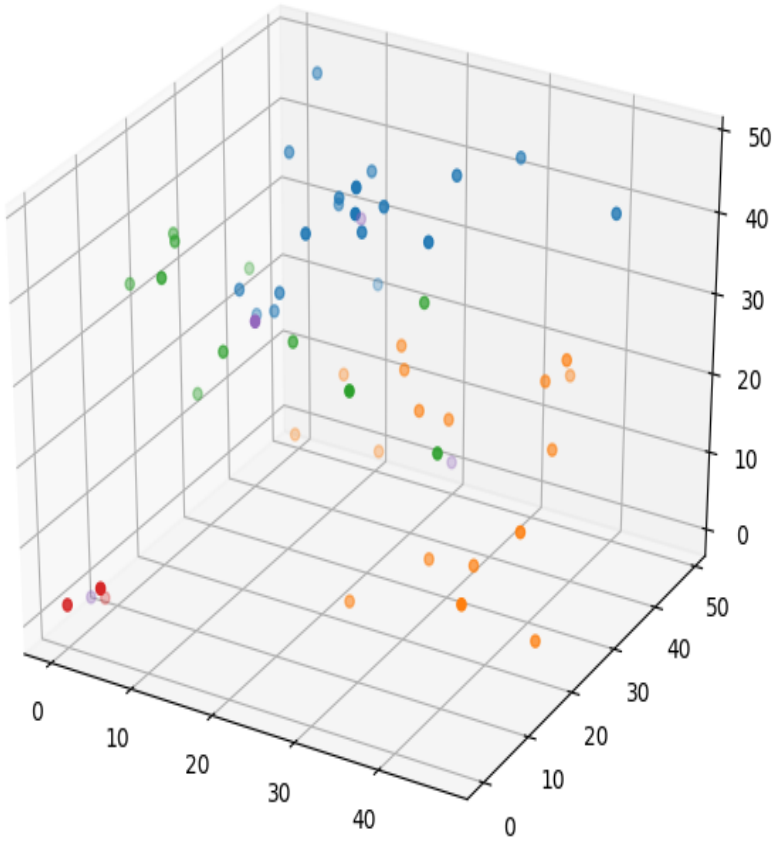


Figure 15: Clustering for 3-D data set by K-means algorithm

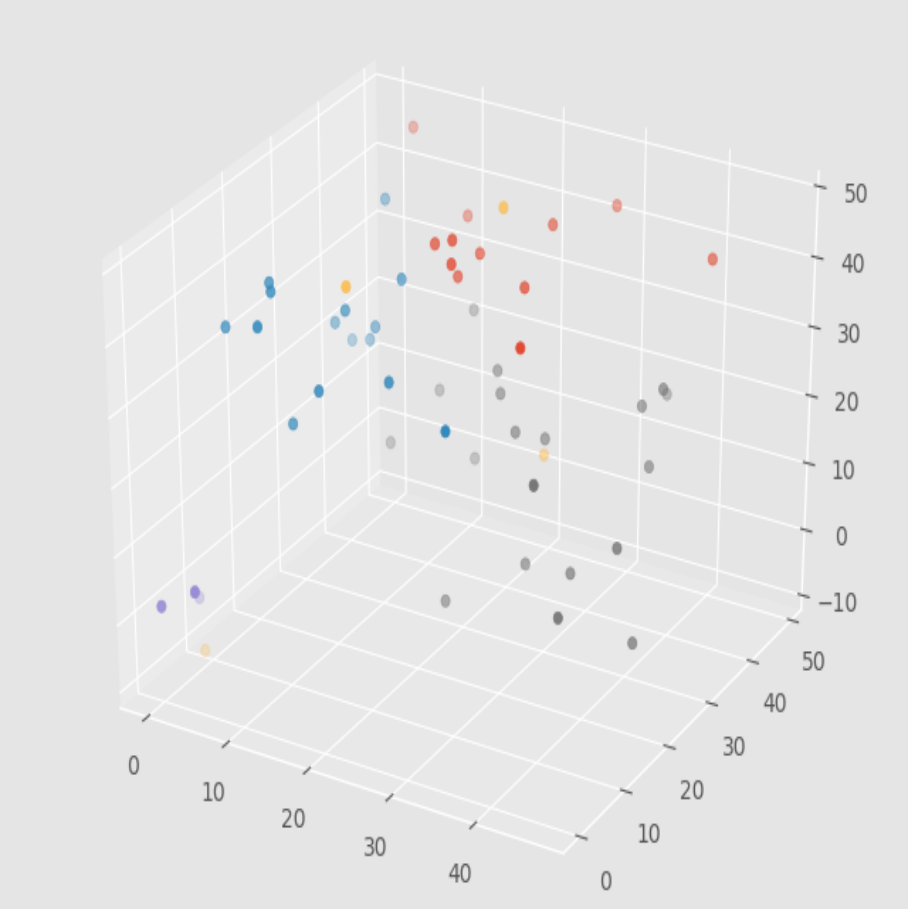


Figure 16: Clustering for 3-D data set by Jaya algorithm

Chapter 6

Conclusion and Future Work

In this article, a novel multi-objective clustering technique modified JAYA based on the newly developed Jaya Evolutionary algorithm is proposed. The explorations and exploitations enforced on the technique, to determine the proper number of clusters, proper partitioning from a given dataset and mere optimal values towards CVIs frontiers, by optimizing fitness functions simultaneously.

Furthermore, it is observed that the aimed algorithm exhibits better performance in most of the considered real-time datasets and is able to cluster appropriate partitions. Much further work is needed to investigate the profound algorithm using different and more objectives, compare with well established clustering algorithm and to test the approach still more extensively over diversified domains of engineering.

References

[1] Baruah, P. Jaya with experienced learning. Master's thesis, Dept. of Computer Science and Engineering,NIT Trichy, 2017.

[2] .Erndira Rendn, Itzel Abundez, A. A., and Quiroz, E. M. Internal versus external cluster validation indexes. Master's thesis.

[3] https://archive.ics.uci.edu/ml/datasets/iris. *UCI Machine Learning Repository.*

[4] https://archive.ics.uci.edu/ml/datasets/wine. *UCI Machine Learning Repository.*

[5] Kurada, R. R., and Kanadam, D. K. P. Automatic unsupervised data classification using jaya evolutionary algorithm. Master's thesis, 2016.

[6] Pandey, H. M. Jaya a novel optimization algorithm: What, how and why? Master's thesis, Department of Computer Science and Engineering, NIT Surat journal= IEEE, year=2017.

[7] Patil, A. B. Bus driver scheduling problem using tlbo. In *International Journal of Computer Applications* (2016), p. Volume 145.

[8] Rao, R. *A Simple and new optimization algorithm for solving constrained and unconstrained optimization problems.* 2016, ch. International Journal of Industrial Engineering Computations, pp. 1–62.

[9] Tapas Kanungo, David M. Mount, N. S. N. C. D. P. R. S., and Wu., A. Y. An efficient k-means clustering algorithm: Analysis and implementation. Master's thesis.