

Meera Ramadas
Ajith Abraham

Metaheuristics for Data Clustering and Image Segmentation

Intelligent Systems Reference Library

Volume 152

Series editors

Janusz Kacprzyk, Polish Academy of Sciences, Warsaw, Poland
e-mail: kacprzyk@ibspan.waw.pl

Lakhmi C. Jain, Faculty of Engineering and Information Technology, Centre for Artificial Intelligence, University of Technology, Sydney, NSW, Australia;
Faculty of Science, Technology and Mathematics, University of Canberra, Canberra, ACT, Australia;
Faculty of Science, Liverpool Hope University, Liverpool, UK
KES International, Shoreham-by-Sea, UK
e-mail: jainlakhmi@gmail.com; jainlc2002@yahoo.co.uk

The aim of this series is to publish a Reference Library, including novel advances and developments in all aspects of Intelligent Systems in an easily accessible and well structured form. The series includes reference works, handbooks, compendia, textbooks, well-structured monographs, dictionaries, and encyclopedias. It contains well integrated knowledge and current information in the field of Intelligent Systems. The series covers the theory, applications, and design methods of Intelligent Systems. Virtually all disciplines such as engineering, computer science, avionics, business, e-commerce, environment, healthcare, physics and life science are included. The list of topics spans all the areas of modern intelligent systems such as: Ambient intelligence, Computational intelligence, Social intelligence, Computational neuroscience, Artificial life, Virtual society, Cognitive systems, DNA and immunity-based systems, e-Learning and teaching, Human-centred computing and Machine ethics, Intelligent control, Intelligent data analysis, Knowledge-based paradigms, Knowledge management, Intelligent agents, Intelligent decision making, Intelligent network security, Interactive entertainment, Learning paradigms, Recommender systems, Robotics and Mechatronics including human-machine teaming, Self-organizing and adaptive systems, Soft computing including Neural systems, Fuzzy systems, Evolutionary computing and the Fusion of these paradigms, Perception and Vision, Web intelligence and Multimedia.

More information about this series at <http://www.springer.com/series/8578>

Meera Ramadas · Ajith Abraham

Metaheuristics for Data Clustering and Image Segmentation



Springer

Meera Ramadas
Information Technology
University College of Bahrain
Manama, Bahrain

Ajith Abraham
Scientific Network for Innovation and
Research Excellence
Machine Intelligence Research Labs
(MIR Labs)
Auburn, WA, USA

ISSN 1868-4394

ISSN 1868-4408 (electronic)

Intelligent Systems Reference Library

ISBN 978-3-030-04096-3

ISBN 978-3-030-04097-0 (eBook)

<https://doi.org/10.1007/978-3-030-04097-0>

Library of Congress Control Number: 2018961197

© Springer Nature Switzerland AG 2019

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

Metaheuristic algorithms like genetic algorithms (GAs), differential evolution (DE), artificial bee colony (ABC), and particle swarm optimization (PSO) have emerged as potential algorithms for dealing with complex optimization problems, which are otherwise difficult to solve by traditional methods. Researchers have continued with their work to create different variants of the existing algorithms to find the most optimal method. Differential evolution (DE) is considered to be a dominant technique for optimization and is being used to solve various real-time problems. DE is a powerful, population-based, stochastic optimization algorithm. The mutation strategy of DE algorithm is an important operator as it aids in generating a new solution vector. Differential evolution algorithm has undergone various changes and evolution by means of numerous researches. The performance of each of these algorithms depends on the changes in mutation and crossover strategies. In this research work, DE and their modified variants are applied to clustering of data and images. Evolutionary algorithms work in a robust and efficient manner for clustering. The ever-growing demand for acquiring, storing, and processing a huge amount of data for analysis and the need for producing customized results have given space for researchers to invent innovative ways for handling such large volume of data. Clustering is an area where the data formed in clusters exhibit efficient customer segmentation, organization of document, information retrieval, extraction of topics, classification, collaborative filtering, visualization, and indexing. Quick and accurate clustering of documents is instrumental in the area of information retrieval system. In the case of images, image segmentation is performed where grouping or clustering of bitmap images into different classes is done such that each class gives one particular information about the bitmap image within the entire image collection. In the past few years, image segmentation has emerged as an active area of research in different domains including medical imaging, satellite imaging. In computer vision, image segmentation is the process of extracting the data from the image, which is very simple and easier to analyze. Image segmentation is a complex task in computer science because it uses the whole image as a data input, which needs to be processed further. Clustering of data and image can be modeled as a nonlinear optimization problem for which suitable

techniques are needed to attain the global solution by overcoming the problem of local maxima. The volume is divided into eight chapters with Introduction as Chap. 1 and Conclusion as Chap. 8. In Chap. 2, detailed studies of the various traditional algorithms are elaborated. Also, the detailed literature study on the variants of the evolutionary algorithm and its application on clustering is given. Chapter 3 deals with the newly created variant of differential evolution named revised differential evolution (ReDE) and its comparison with the classical mutation strategy of DE. Chapters 4 and 5 elaborate new variants of evolutionary algorithm named as search strategy flower pollination algorithm with differential evolution (ssFPA/DE) and forced strategy differential evolution (FSDE) and its usage on data clustering. Chapter 6 deals with the usage of another variant of DE on the segmentation of weather images. The algorithms are elaborately studied and are tested on a large number of test images with their performance carefully analyzed in reference to various performance measures including statistical analysis. It is observed that although all the algorithms are suitable for dealing with clustering, DE and its variants perform better.

Manama, Bahrain
Auburn, USA

Meera Ramadas
Ajith Abraham

Contents

1	Introduction	1
1.1	Major Objectives	4
1.2	Organization	4
Reference		5
2	Metaheuristics and Data Clustering	7
2.1	Genetic Algorithm	7
2.1.1	Basics of Genetic Algorithms	7
2.1.2	Reproduction	8
2.1.3	Crossover	8
2.1.4	Mutation	9
2.2	Particle Swarm Optimization Algorithm	10
2.3	Differential Evolution	11
2.3.1	Initialization	12
2.3.2	Mutation	13
2.3.3	Crossover	13
2.3.4	Selection	14
2.3.5	Algorithm	14
2.4	Flower Pollination Algorithm	15
2.5	Data Clustering	18
2.5.1	Various Clustering Techniques	21
2.5.2	Distance Measures Used in Clustering	22
2.5.3	K-Means Technique	24
2.6	Image Segmentation	25
2.7	Work Done on Metaheuristics	30
2.8	Application of Metaheuristics on Data Clustering	35
2.9	Application of Metaheuristics on Image Segmentation	41
References		48

3 Revised Mutation Strategy for Differential Evolution	
Algorithm	57
3.1 Introduction	57
3.2 Revised Mutation Strategy	58
3.3 Experimental Setting	58
3.4 Graphical Results	62
3.5 Statistical Analysis	63
3.6 Summary	63
4 Search Strategy Flower Pollination Algorithm with Differential Evolution	67
4.1 Introduction	67
4.2 Proposed Hybrid Method ssFPA/DE	68
4.3 Experimental Results	70
4.4 Graphical Representation	71
4.5 ssFPA/DE in Clustering	77
4.6 Experimental Results on Clustering	82
4.6.1 Cluster Quality	85
4.6.2 Validation Index	90
4.6.3 Graphical Representation	92
4.7 Summary	93
References	94
5 Forced Strategy Differential Evolution Used for Data Clustering	95
5.1 Introduction	95
5.2 Proposed Variant of Differential Evolution	97
5.3 Test Problem	97
5.4 FSDE in Clustering	106
5.5 Experimental Results on Clustering	106
5.5.1 Cluster Quality	110
5.5.2 Validation Index	114
5.5.3 Graphical Representation	118
5.6 Summary	118
References	119
6 Reconstructed Mutation Strategy for Differential Evolution	
Algorithm	121
6.1 Introduction	121
6.2 RDE Mutation Strategy	122
6.3 Experimental Results and Statistical Analysis	123
6.4 Multi-level Image Thresholding	123

6.5	Multi-level Thresholding Using RDE Strategy	129
6.6	Test Results on Image Thresholding	129
6.7	Summary	131
7	Enhanced Differential Evolution with Fuzzy c-Means Technique	137
7.1	Introduction	137
7.2	Fuzzy c-Means Clustering	138
7.3	Enhanced Differential Evolution	139
7.4	Experimental Setting	140
7.4.1	Statistical Analysis	141
7.5	eDE in FCM Clustering	141
7.6	Experimental Results on Clustering	141
7.6.1	Validation Index	145
7.6.2	Graphical Representation	152
7.7	Summary	152
	References	153
8	Conclusion and Future Scope	155
8.1	Future Scope	156
	Appendix	157
	Bibliography	161

Chapter 1

Introduction



In the field of computer science and engineering, metaheuristic is a high level procedure, which finds, selects or generates a low level procedure that will give an optimized solution to a complex problem. It is a high-level problem independent framework that gives a set of strategies for developing heuristic algorithms. Metaheuristics are used for combinatorial optimization like genetic algorithm, evolutionary algorithms, particle swarm optimization, simulated annealing etc. It explores the search space to find the near optimal solutions. Optimization is the selection of best element from the set of alternative solutions.

Two components in metaheuristics are local exploitation and global exploration. Exploitation consists of probing a limited region of search space in the hope of improving the solution at hand. Exploration consists of probing a much larger portion of the search space with the hope of finding other solutions that are yet to be refined. Exploration looks for new solutions in new search space while exploitation uses existing solutions and makes refinements to it to improve its fitness. Global exploration explores the different feasible region in the entire search space for obtaining global optimum. Local exploitation uses local information from local search to ensure maximum convergence. These two concepts are conflicting and a balance need to be maintained. There are different types of metaheuristics namely local search and global search metaheuristics, single solution and population based metaheuristics, hybridization and memetic algorithms, parallel metaheuristics and nature inspired metaheuristics.

In past decades, several metaheuristic techniques have evolved by exploring various concepts inspired from nature called nature inspired metaheuristics. These techniques essentially mimic the characteristics that uniquely exist in the Mother Nature. Nature has in itself evolved for many years continuously developing creative methods for problem solving and for adaption in this ever changing world environment. During the Second World War, Alan Turing used heuristic algorithm for the first

time. The search method that he used then was named as heuristic search. In 1948, a report was released by Turing in which he elaborated his innovative ideas on machine intelligence and learning, evolutionary algorithm and neural networks. Since 1950, the concept coined by Darwin started to make inroads into the world of computation world. The great scientist Darwin had coined the concept of survival of the fittest. It dealt with the origin, adaption and survival of the species in nature. The 1960s and 1970s saw a great development in evolutionary algorithms.

There is a trend set among the scientific community towards the use of nature inspired metaheuristics to effectively solve and model complex optimization problems. Nature inspired metaheuristics algorithm uses all evolutionary algorithm techniques. Evolutionary algorithm perform parallel search in complex search space. The main disadvantage of classical optimization problem is its inability to adapt its solution algorithm to a given problem. Owing to this need, more adaptable and flexible general purpose algorithms are in demand. Based on this spur, many nature inspired algorithms were developed like Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Ant Bee Colony (ABC), Flower Pollination Algorithm (FPA), etc. These algorithms provide better solutions in comparison to traditional algorithms. Evolutionary algorithm has become dominant problem solving methodology among researchers. Robustness, population based collective learning process and self-adaption are some of the advanced features of evolutionary algorithms when compared with other global optimization techniques.

Fogel [1] introduced the concept of evolutionary computing. Evolutionary Computing is a subfield of Artificial Intelligence that uses continuous combinatorial optimization problems. It uses iterative process of growth and development. Evolutionary algorithm is a subset of evolutionary computing. Evolutionary algorithm uses the basic steps of selection, crossover, mutation and replacement.

The basic steps for evolutionary algorithm are:

1. Initialize the population.
2. Calculate the fitness of each element in the population.
3. Repeat until termination.
4. Select the best fit individuals for reproduction.
5. Create new individuals through crossover and mutation.
6. Calculate the fitness of the new product.
7. Compare the new product with the elements in the population and replace the least fit element with the new product.

Exploration is obtained by randomization that enables the algorithm to reach the capacity to come out of the local optimum and thereby perform the search globally. Randomization is used in local searches near the current best if limited to local region. If steps are large, then search is done globally. Balancing global and local search and correct randomness are important for efficiency of metaheuristic algorithm. Exploration uses local information like history of search and gradients. Knowledge from simulation and observations on convergence behavior of algorithms implies that exploration increases speed of convergence and decreases rate of convergence of

algorithm. There are two types of optimality approach. In landscape-based optimality, the information about the type of search landscape is focused. In algorithm-based type, objective function is treated as a black box type and it uses the information during iteration to find the best optimal solution. At times, combination of both the types is also used.

Most of the metaheuristic approaches like DE, FPA has been applied over various optimization problems like clustering. These approaches can find an optimal partition of the dataset, when the number of clusters is given. In this research, variants of evolutionary algorithm are introduced and these variants are applied in the field of data clustering and image segmentation. Clustering is a process of breaking down large population into smaller groups that are similar in character. It is mainly used for data compression. This maximizes the inter class similarity. Good clustering has a high within cluster similarity but low inter cluster similarity. It is used for improving precision in information retrieval. Clustering is a type of unsupervised learning where the data has no target attribute. It studies how system can learn to represent particular input patterns in a way that reflects the statistical structure of the overall input pattern assembly.

Similarly, clustering of text documents plays a vital role in efficient document organization, summarization, topic extraction, information retrieval, customer segmentation, classification, collaborative filtering, visualization and indexing. Fast and accurate clustering of documents plays an important role in the field of automatic information retrieval system. It is considered as a multi objective optimization problem. It involves an iterative task of trial and error. Clustering can be classified as hard clustering and soft clustering. In hard clustering, each object belong to one cluster or do not belong to any cluster. In soft clustering, each object may belong to more than one cluster. Clustering algorithm can be classified into different types namely exclusive clustering, overlapping clustering, hierarchical clustering and probabilistic clustering. In exclusive clustering, each data is grouped into a particular cluster based on its similarity to that cluster. K-means algorithm is an example for exclusive clustering. Here, a given set of N data is partitioned into k different clusters. Grouping of data is done by minimizing the Euclidian distance between the data and centroid. It is one of the simplest unsupervised learning algorithms used for clustering. This algorithm is significantly sensitive to initial randomly selected centroid.

Evolutionary algorithm can also be applied on to the area of unsupervised image clustering where the implementation of cluster analysis can aid in easier understanding of the images. Image clustering is often applied to segment and have easier understanding of images based on the desired region of interest. The process of segmentation is done using clustering techniques. Image segmentation refers to the partitioning or dividing of a digital image into several smaller parts or segments in order to study a given image in a detailed manner. Image segmentation is required to study in detail some particular features (areas of interest) of a digital image. Thresholding is one of the simplest techniques used for image segmentation. It is the method of creating binary images from the given image. It forms an important and exigent part of image processing and requires an exhaustive and robust search technique for its implementation. Thresholding on colored images is done by manipulating the

color components based on color spaces. Problems like thresholding are classified as unsupervised learning and such issues can be solved using evolutionary algorithms.

Though evolutionary algorithms are being widely applied and accepted in diverse areas like engineering, medicine, commerce, business etc., in reality they give only marginal performance. Some of the major reasons attributed to the marginal performance are inappropriate selection of parameters, inappropriate representation etc. Hence, here there is tremendous scope and an urgent need to develop more hybrid evolutionary algorithms that has the ability to optimize the performance of direct evolutionary approach.

Hybridization of evolutionary algorithms are needed for the following:

- i. improving standard of output
- ii. improving the conduct
- iii. to integrate evolutionary algorithm to a larger system.

1.1 Major Objectives

The core objective of this work is to create new variants of evolutionary algorithm to solve specific optimization task. Subsequently, compare the newly created variant with some of existing metaheuristic techniques that are commonly available to check the efficiency of the new variants.

The basic objectives of this work are:

1. To propose the hybrids of Differential Evolution algorithm.
2. To perform the statistical analysis for each of the new variant to validate its efficiency.
3. These proposed hybrids will be applied on to clustering of data and images to demonstrate their application capability.

1.2 Organization

This volume is structured into several chapters with the first two chapters focusing on the supporting information with respect to this research work and each of the subsequent chapters explaining in detail the research work that was performed along with the research outcome.

The below is the organization for the following sections:

Chapter 2 contains a detailed study of Differential Evolution and Flower Pollination algorithm. This chapter not only explain in detail various steps involved in these algorithms but also an in-depth study is done on the basic concepts of data clustering and image segmentation. This chapter also discusses the literature review about previous research works carried out on different variants created in evolution-

ary algorithms and their findings. This chapter also has references to the works in the field of data clustering and image segmentation through clustering.

Hereafter, the following chapters form the core section of this work where each of the variants created using Differential Evolution and Flower Pollination Algorithm is explained in detail in each of Chaps. 3, 4, 5, 6 and 7 respectively. Each of these chapters also comprise of the results obtained by applying these variants in the field of data clustering and image segmentation.

Chapter 8 explains in brief the conclusion of the research work and future scope of work that can follow up based on this research work.

Reference

Fogel, L.J., Owens, A.J., Walsh, M.J.: Intelligent decision making through a simulation of evolution. *Behav. Sci.* **11**(4), 253–272 (1966)

Chapter 2

Metaheuristics and Data Clustering



In this Chapter, the initial Sects. 2.1–2.4 give detailed analysis on types of metaheuristics namely Genetic Algorithm, Particle Swarm Optimization, Differential Evolution algorithm and Flower Pollination algorithm. Section 2.5 elaborates the basic data clustering technique and Sect. 2.6 summarizes about the image segmentation. Sections 2.7–2.9 explains the works related to metaheuristics and their application on data and image clustering respectively.

2.1 Genetic Algorithm

Holland [1] introduced genetic algorithms. Genetic algorithm (GA) is used to produce high quality solutions to search and optimize the problems using the concepts of mutation, crossover and selection. GA has been successfully applied in many areas like clustering, image segmentation, image watermark etc.

2.1.1 Basics of Genetic Algorithms

Genetic algorithms is based on Darwin's theory of survival of the fittest. The basic idea behind GA is natural selection and natural genetic systems in which fittest individuals will survive and the remaining individuals will be discarded. In GA, take a set of candidate solution and iteratively refine them by alternating and selecting the good solution for next generation. Select the candidate solutions on the basis of fitness function to evaluate quality. The five phases of genetic algorithm are initial population, fitness function, selection, crossover and mutation. Around the nature's scanty resources, competition within individuals lead to fittest dominating the weaker. They are a computerized searching and optimization algorithm that uses natural selection and natural genetics technique. They take large search spaces and look

for optimal combination of solutions. The three important aspects in GA include defining optimization function, defining and implementing genetic representation, and genetic operators.

GA begins with the chromosome representation of a parameter set $\{x_1, x_2, \dots, x_n\}$; generally, the chromosomes are designed as strings of 0's and 1's. GAs employs three basic operators over a limited number of strings for finding the global near-optimal solution. These operators are:

- (i) Reproduction/Selection
- (ii) Cross-over
- (iii) Mutation.

2.1.2 Reproduction

According to the objective function, copy the individual strings for reproduction process. The strings created by reproduction depends upon the Darwinian survival theory, where the fitter strings have more chance to get a higher number of offspring in the succeeding generation. All the successful strings form a mating pool which will behave like a new tentative population.

2.1.3 Crossover

Reproduction operation forms a mating pool of strings; after this phase, these strings will form offspring for the new generation using crossover operation. Crossover process consists of two steps: (1) all the members of the mating pool will mate randomly and (2) each pair of strings undergo crossover as follows:

- i. Select an integer position k at random between l and $l - 1$ where, l is the string length greater than 1.
- ii. Create two new strings by swapping all characters from position $k + 1$ to l .

Let a and b be two strings selected for the crossover operation from mating pool and the random number generated be 10 (ten).

$$\begin{aligned} a &= 110001010101000\dots0111110001 \\ b &= 100010111011101\dots0011010100 \end{aligned}$$

Then the newly produced offspring by swapping all characters after position 10 will be denoted as a' and b' :

$$\begin{aligned} a' &= 11000101011101\dots0011010100 \\ b' &= 100010111001000\dots0111110001 \end{aligned}$$

2.1.4 Mutation

In the simple GA, mutation is the occasional random alteration of the value of a string position. It helps to prevent the irrecoverable loss of potentially important genetic material. Select a random bit position of a random string and replace it by another character from the population.

For example, let the fifth bit of string ‘*a*’ given above, be selected for mutation. Then the transformed string after mutation will be:

$$a = 110011010101000 \dots 0111110001$$

The flow diagram for genetic algorithm is as given below in Fig. 2.1.

The general pseudo code for genetic algorithm is given below:

```

Initialize population with random candidate solution
Evaluate fitness of each candidate solution
While termination condition not true do
    Select the best individuals to be used by genetic operators
    Generate new individuals using crossover and mutation
    Evaluate fitness of new individuals
    Replace the worst individuals of population by best new individuals
End

```

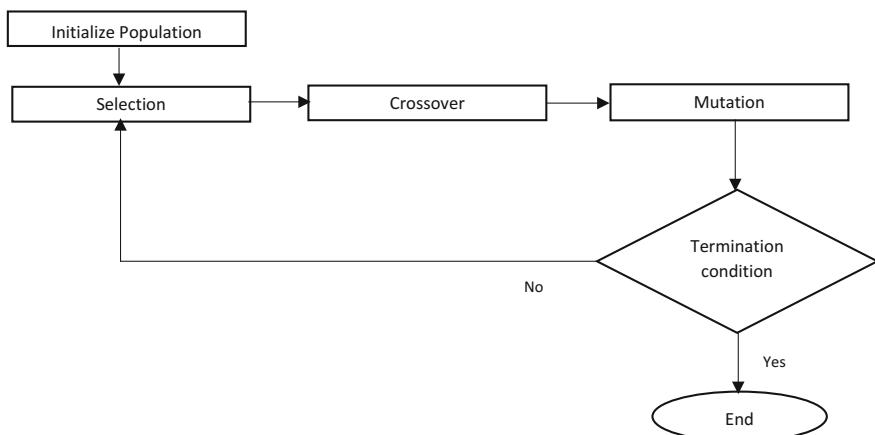


Fig. 2.1 Flow diagram for genetic algorithm

2.2 Particle Swarm Optimization Algorithm

Eberhart and Kennedy [2] introduced a method for solving continuous nonlinear function. It is the simulation of social behavior of birds and was termed as Particle Swarm Optimization (PSO). It moves solutions through search space. Swarm is an identified collection of active individuals, which tend to group together while each individual seems to be moving in arbitrary direction. It is very simple in concept, easy to implement and computationally efficient. This concept has its origin from evolutionary computing and artificial life, which generate the population by assigning random position and velocities. Unlike GA, PSO does not have operators like mutation and crossover.

Each particle keeps track of best position in the hyperspace. It also keeps track of the previous best position for individual particle, global best position in population and local best position in the defined neighborhood. Particles take benefit from the previous particle's experience and makes search for higher objective function values. Treat each particle as a point in N -dimensional space which adjusts its "flying" according to its own flying experience as well as the flying experience of other particles. The particles revise itself by using the internal velocities. This algorithm also utilizes memory space unlike other evolutionary algorithms.

In PSO algorithm, each particle consist of a data representing a solution, personnel best position pbest and velocity value. pbest is the personnel best position of a particle having its co-ordinate in the solution space with the best solution achieved by the particle. Velocity value indicates how much the value has changed. PSO keeps track of three global variables namely target value, global best position gbest and the stopping value indicating when the algorithm should terminate. gbest is the global best value achieved by a particle. Basic idea behind PSO is to accelerate each particle towards its gbest and pbest positions using a random weighted acceleration. Position of each particle will change on the basis of the (i) current positions (ii) current velocities (iii) distance between pbest and current positions and (iv) the distance between gbest and current positions. Therefore, the modifications will be done on the basis of given equation:

$$\begin{aligned} v_{i,k+1} &= v_{i,k} + c_1 \cdot \text{rand}(0, 1) \cdot (pbest_i - present_i) \\ &\quad + c_2 \cdot \text{rand}(0, 1) \cdot (gbest - present_i) \end{aligned} \quad (2.1)$$

$$present_{i+1} = present_i + v_{i,k+1} \quad (2.2)$$

where, c_1 and c_2 are the weighted factor, $present_i$ is the current position of the candidate, $v_{i,k+1}$ is the velocity of the candidate.

The general algorithm for PSO is given as follows:

```

Initialize velocity and position of each particles
While stopping criteria not attained do
    For each particle
        Calculate fitness value
        If fitness value is better than pbest value in history
            Set current value as new pbest
        End
        Choose the particle with best fitness value of all particles as gbest
        For each particle
            Calculate particle velocity according to eq. 2.1
            Calculate particle position according to eq. 2.2
        End
    End
End

```

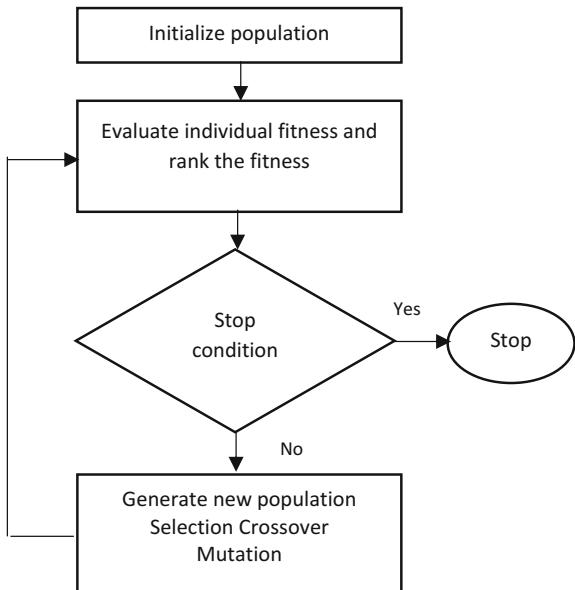
PSO algorithm differs from other evolutionary algorithms as it doesn't have the operations of mutation and crossover. The particles revise itself by using the internal velocities. This algorithm also utilizes memory space unlike other evolutionary algorithms. Flow diagram for PSO is given in Fig. 2.2.

2.3 Differential Evolution

Differential Evolution (DE) is a heuristic optimization technique proposed by Storn and Price in [3]. It is used for minimizing non-differential, non-linear, continuous space functions with real valued parameters. The most important characteristic of DE is that it uses the differences between randomly sampled pairs of object vectors to guide the mutation operation unlike the other EAs, which instead use the probability distribution functions. DE is a technique that optimizes a problem by iteratively trying to improve candidate solution with respect to given measure of quality.

The operations of crossover, mutation and selection make DE different from other evolutionary algorithms. For genetic algorithm and evolutionary algorithm, mutation is treated as a random change in some parameter. DE mutates the base vector with scaled population of derived difference vector. Evolutionary programming requires adaption of absolute step size for each variable over generation, while DE requires only specification of single relative scale factor for all variables. In crossover, both DE and evolutionary computing uses crossover to create a single trial vector, while genetic algorithm recombine two vectors to produce trial vectors by one point

Fig. 2.2 Flow diagram for particle swarm optimization



crossover. In selection, genetic algorithm selects parent based on fitness while DE give all individuals an equal chance to be selected for mutation.

In a population of potential solution within an n-dimensional search space, a fixed number of vectors are randomly initialized which is evolved over time to explore search space and to locate the minima of objective function. At each iteration, new vectors are generated by combination of vectors randomly chosen from current population. The outcome vectors are then combined with a predefined target vector. This operation is called recombination. Thereafter, the trial vector is accepted for next generation if and only if it yields a reduction in value of objective function. This is called selection. The various steps involved in Differential Evolution approach are explained in detail in the below section.

2.3.1 Initialization

DE is a parallel direct search method using a population of N parameter vectors for each generation. At generation G , the population PG is composed of $X_{i,G}$ where $i = 1, 2, \dots, N$. For a given parameter $X_{i,G}$, three vectors $X_{r1,G}, X_{r2,G}$ and $X_{r3,G}$ are randomly selected such that r_1, r_2, r_3 are distinct. This operation extends the workspace utilization as the individuals are chosen randomly by moving around in the search space and giving an equal chance for all individuals to be selected for mutation. The initial population can be chosen randomly under uniform probability distribution if there is no prior information about the problem to be optimized. If

any prior knowledge about the problem is available, the preliminary solution can be included to the initial population by adding normally distributed random deviations to the nominal solution.

2.3.2 Mutation

The key characteristic of a DE is the way in which it generates trial parameter vectors throughout the generation. This operator makes DE different from other Evolutionary Algorithms. It computes the weighted difference between the vectors in population. A weighted difference vector between two individuals is added to a third individual to form a new parameter vector. The newly generated vector is evaluated by the objective function. For each parent parameter vector, DE generates a candidate child vector based on the distance of two other parameter vectors. For each dimension i , the donor vector $V_{i,G}$ is computed as:

$$V_{i,G} = X_{r1,G} + F \cdot (X_{r2,G} - X_{r3,G}) \quad (2.3)$$

Here F is the mutation factor which is a constant in $(0,2)$. The above mutation scheme is named as DE/rand/1. Mutation function demarcates one DE scheme from another. The most often used DE codes are given below:

$$\text{DE/rand/2 } V_{i,G} = X_{r1,G} + F \cdot (X_{r2,G} - X_{r3,G}) + F \cdot (X_{r4,G} - X_{r5,G}) \quad (2.4)$$

$$\text{DE/best/1 } V_{i,G} = X_{best,G} + F \cdot (X_{r1,G} - X_{r2,G}) \quad (2.5)$$

$$\text{DE/best/2 } V_{i,G} = X_{best,G} + F \cdot (X_{r1,G} - X_{r2,G}) + F \cdot (X_{r3,G} - X_{r4,G}) \quad (2.6)$$

$$\text{DE/rand-to-best/1 } V_{i,G} = X_{r1,G} + F \cdot (X_{best,G} - X_{r2,G}) + F \cdot (X_{r3,G} - X_{r4,G}) \quad (2.7)$$

where, $i = 1, 2 \dots NP$, $r_1, r_2, r_3 \in \{1, 2, \dots, NP\}$ are randomly selected and $r_1 \neq r_2 \neq r_3 \neq i$, $F \in [0, 2]$, F is the control parameter proposed by Storn and Price [3].

2.3.3 Crossover

This process, also called recombination, incorporates successful solutions into the population. The value of the corresponding objective function is compared with a pre-determined individual. If the newly generated parameter vector has lower objective function value, it will replace the predetermined parameter vector. The best parameter vector is evaluated for every generation in order to track the progress made throughout the minimization process. The random deviations of DE are generated by using both the search distance and the direction information from the population. Correspondingly, this adaptive approach is associated with the normally fast

convergence properties of a DE. The trial vector $U_{i,G}$ is created for the target vector $X_{i,G}$ through binomial crossover. Elements of donor vector enter trial vector with probability $C_r \in [0, 1]$. C_r is the crossover probability which is selected along with population size $NP \geq 4$.

$$U_{j,i,G+1} = \begin{cases} V_{j,i,G+1} & \text{if } rand_{i,j}[0, 1] \leq C_r \text{ or if } j = I_{rand} \\ X_{j,i,G+1} & \text{if } rand_{i,j}[0, 1] > C_r \text{ or if } j \neq I_{rand} \end{cases} \quad (2.8)$$

Here, $rand_{ij} \cong \cup[0, 1]$ and I_{rand} is random integer from 1, 2, ..., NP.

2.3.4 Selection

This operation differs from the selection operation of other evolutionary algorithms. Here, the population for next generation is selected from vectors in current population and its corresponding trial vectors. The target vector $X_{i,G}$ is compared with the trial vector $V_{i,G}$ and the lowest function value is taken into next generation.

$$X_{i,G+1} = \begin{cases} U_{i,G+1} & \text{if } f(U_{i,G+1}) \leq f(X_{i,G}) \text{ where } i = 1, 2, \dots, NP \\ X_{i,G} & \text{otherwise} \end{cases} \quad (2.9)$$

Mutation, crossover and selection operations are continued until some stopping criteria is reached.

2.3.5 Algorithm

The algorithm for Differential Evolution approach is given below:

for every element X in the group do

Select three elements $X_{r1,G}, X_{r2,G}, X_{r3,G}$ from the group which follows

$$X_{r1,G} \neq X_{r2,G} \neq X_{r3,G} \neq X$$

Select index R in n randomly where n is the size of the problem to be optimized
Compute the element's new location $V_{i,G}$ as :

$$V_{i,G} = X_{r1,G} + F \cdot (X_{r2,G} - X_{r3,G})$$

for every candidate, pick a uniformly distributed number r and $C_r \in [0,1]$

```

If  $rand_{j,i} \leq C_r$  or  $j = rand_{j,i}$  then
    Set  $U_{j,i,G+1} = V_{j,i,G+1}$ 
Else
    Set  $U_{j,i,G+1} = X_{j,i,G+1}$ 
end if
end for

If  $f(U_{i,G+1}) \leq f(X_{i,G})$  then change the element in the group with the new
improved result

Identify the element from the group with the lowest cost or highest fitness
return the value as the best candidate result
end for
End

```

The flow chart for Differential Evolution algorithm is given in Fig. 2.3.

2.4 Flower Pollination Algorithm

From the point of view of biological evolution, the purpose of pollination in flower is to facilitate the optimal reproduction of plants taking into account their numbers and

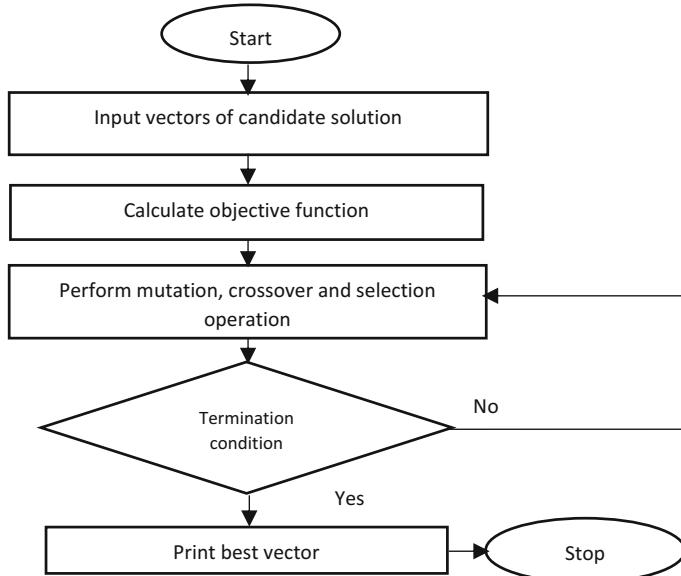


Fig. 2.3 Differential Evolution algorithm

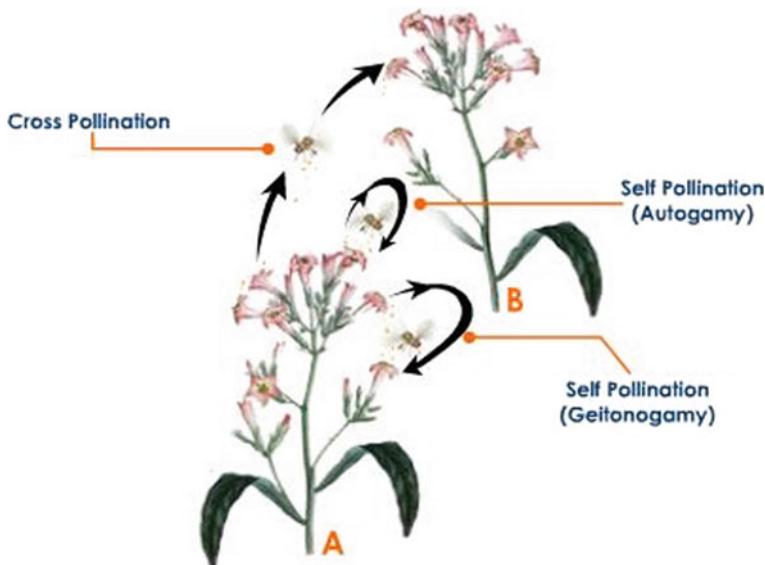


Fig. 2.4 Pollination in flowers (Source <https://biology.tutorvista.com/>)

the fittest. This is in effect an optimization process that take place within plant species. The Flower Pollination Algorithm (FPA) uses the technique from the pollination of flowering plants. Flowers reproduce using the concept of pollination. Yang et al. [4] proposed a detailed work on FPA for multi-objective situations. Pollination (see Fig. 2.4) is the method of relocating pollens from the male anther to the female stigma of a flower. The goal of all living things is to produce offspring for the next genesis. One of the methods that plants follow to produce offspring is by making seeds.

Pollination in plants is of two types:

- Cross Pollination
- Self-Pollination.

Cross Pollination occurs when the pollens from one plant is transferred to a flower that belong to another plant. This process takes place with biotic and abiotic creatures. Abiotic pollination occurs when external agents like wind, water etc. are involved. Very few plants come under this category of pollination. Biotic pollination is moving pollen from anther of one plant to stigma of another plant using external pollinators like birds, insects, bees, etc. The pollinators are attracted by color, odor and availability of nectar.

Self-Pollination on the other hand occurs when the flower pollinates within the same plant. These pollination processes involves complex techniques been undertaken for plant production strategies. Algorithm was developed based on these fundamental techniques. The global pollination occurs for cross pollination and biotic pollen transfer whereas self-pollination and abiotic transfers are associated with local

reproduction strategy. Plants can follow either method of pollination. The tendency of individual pollinators to visit a plant depends on the attractiveness of that plant. A solution of optimization problem is associated with a pollen gamete. Here, the flower constancy is compared to solution fitness.

Biotic pollination takes place in 90% of flowering plants where pollen is transferred by a pollinator such as insects and animals. About 10% of pollination takes in abiotic form, requiring no pollinators. Wind and water diffuses pollination in these plants with flowers and grass being a good example. Pollinators that are sometimes called pollen vectors can be very diverse. They estimate that there are at least 200,000 varieties of pollinators, such as insects, bats and birds. Bees are considered to be a good example of pollinators, which has also developed the so-called floral constancy. These pollinators tend to visit certain unique species of flowers bypassing other species of flowers. Such evidence may have evolutionary floral benefits, since this process maximize the transfer of flower pollen within the same plant or of the same species plants and thus maximizing the reproduction of the same species of flowers. Instead of focusing on some unpredictable but potentially more rewarding new flower species, flower constancy may require minimal investment cost and most likely guaranteed intake nectar.

Using local and global pollination techniques, the pollens will be transferred to a plant. Global method uses pollinators to transfer pollens to long distances towards individual flowers characterized by higher fitness. Local pollination occurs in a limited range of individual flower using pollination mediators like air, water etc. Birds, flies, bees and bats are the most common pollinators that fly for long distance. These pollinators are considered as the carriers of global pollination.

FPA was formed by Yang [5]. This algorithm follows four rules.

- Self-pollination and abiotic method are considered as local pollination.
- Cross pollination and biotic method are considered as global pollination.
- Duplication probability is directly proportional to the likeness of two flowers involved. So flower consistency is considered.
- Switch probability p controls the local and global pollination.

These rules can be converted to equations as given below:

$$x_i^{t+1} = x_i^t + \varepsilon(x_i^t - x_k^t) \quad (2.10)$$

$$x_i^{t+1} = x_i^t + L(x_i^t - g_*) \quad (2.11)$$

where x_i^t is the solution vector and g_* is the current best during the iterations.

L is the step size from Levy distribution which denotes the strength of pollination. Levy flight is used to imitate the movement of insects over a long distance.

The parameter L is computed as:

$$L \sim \frac{\lambda \Gamma(\lambda) \sin(\pi \lambda / 2)}{\pi} \frac{1}{S^{1+\lambda}}, \quad (S \gg S_0 > 0) \quad (2.12)$$

where, λ is the probability density function, $\Gamma(\lambda)$ is the standard gamma function and S is the step size of pollination. This is a continuous probability distribution for positive random numbers. Flower Pollination algorithm has been used widely in the field of fuzzy logic and engineering. It is flexible, simple and is exponentially better to solve.

The algorithm for FPA is given below:

```

Objective function  $f(x), x = (x_1, x_2, \dots, x_d)$ . Initialize  $n$  pollen seeds

Select optimal result ( $g^*$ ) in the original group

State  $t$  as number of iterations and a switching probability as  $p \in [0,1]$ 

While ( $Max\ Generation > t$ )
    for all  $n$  flowers in the group
        If ( $p > rand$ )
            For Levy distribution, select a step vector  $L$  of  $d$  dimension
            Do global pollination  $x_i^{t+1} = x_i^t + L(x_i^t - g^*)$ 
        Else
            Select item from a uniform distribution within  $[0,1]$ 
            Do local pollination  $x_i^{t+1} = x_i^t + \varepsilon(x_i^t - x_{ij}^t)$ 
        end if
        Calculate new solution
        Update population if the new solutions are improved
    end for
    Find current best solution

end while

Output solution

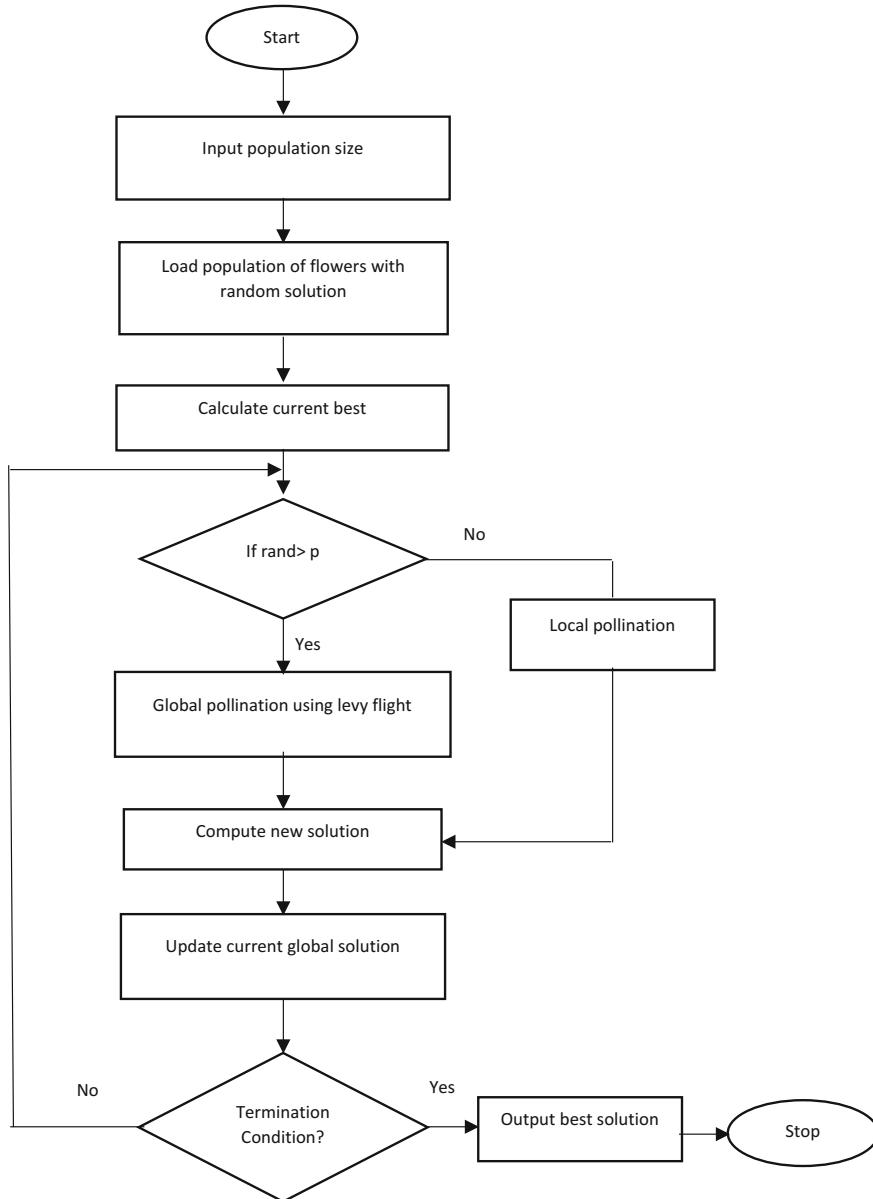
End

```

The flowchart of Flower Pollination Algorithm is given below in Fig. 2.5.

2.5 Data Clustering

With the advances in technology, the need for acquiring, storing and processing a huge amount of data is ever increasing. Clustering is a process of partitioning a set of data (or objects) into a set of meaningful sub-classes, called clusters. It helps the user to understand the natural grouping or structure in a data set. It is breaking down of large population into smaller groups that are similar in character. A cluster is therefore a collection of objects which are similar internally, but clearly dissimilar to

**Fig. 2.5** Flowchart for Flower Pollination algorithm

the objects belonging to other clusters. Dissimilarities and similarities are assessed based on the attribute properties describing the objects. It is a connected region of multidimensional space, containing a relatively high density. The quality of a

clustering result also depends on both the similarity measure used by the method and its implementation. The quality of a clustering method is also measured by its ability to discover some or all of the hidden patterns. By organizing data into clusters, there should be high intra cluster similarity and low inter cluster similarity. In clustering, the objects of analysis could be persons, salaries, opinions, software entities and many others. These objects must be carefully presented in terms of their characteristics. These characteristics are the main variables of the problem and their choice greatly influences the results of a clustering algorithm.

Clustering is based on three characteristics namely nesting, exclusiveness and completeness. In the nested type, separation is built on characteristics of nesting clusters. Hierarchical clustering is nested meaning it clusters to exist within the bigger clusters. In hierarchical type, a set of nested clusters is organized into a hierarchical tree. Partitioned clustering prohibits subsets of cluster. Here, a division of data objects is grouped into non-overlapping subsets (clusters) such that each data object is in exactly one subset. Exclusive separation is the characteristics in which the data object is allowed to exist in one or more than one cluster. But, in exclusive clustering, each of the data object shall exist in only one cluster. The data objects can be grouped into two or more clusters through overlapping. For example, in overlapping clustering, it can allow a student to be grouped as employee also while in case of exclusive clustering, it demands that the person be grouped to the once which are more important. In case of fuzzy clustering before also known as probabilistic clustering, each of data object shall belong to every cluster. The data objects here possess membership weightage between 0 and 1 where 0 shall point that it does not belong to that cluster whereas 1 denote it belong to that particular cluster. Completeness is a type of separation, which requires all the data objects to be grouped. In complete clustering, every object is assigned to a cluster. Various steps are involved in data clustering namely, data collection, initial screening, representation, clustering tendency, clustering strategy, validation and interpretation. Data collection includes extracting related data from the source. Initial screening is the messaging of data after extraction from source. Representation means preparing data in order to make it proper for algorithm. Clustering tendency verifies if the data in hand can be considered in a cluster or not. Clustering strategy chooses the correct algorithm and parameter. Validation is the manual examination of the data for validity of technique. Interpretation includes combining clustering results with other studies and suggesting further analysis.

In document clustering (see Fig. 2.6), it is possible to view the clustering problem as an optimization problem that locates the optimal centroids of the clusters rather than to find an optimal partition. This view offers us a chance to apply evolutionary algorithms for clustering problem. The appropriate clustering algorithm and parameter settings (including values such as the distance function to use, a density threshold or the number of expected clusters) depend on the individual data set and their intended use of the results. Partitional algorithm permits to decompose data into sets of disjoint clusters using similarity criterion. It allows updating of cluster members if the clustering performance is improved. Partitional clustering is an optimization problem as it minimizes the cluster assignment in probability density

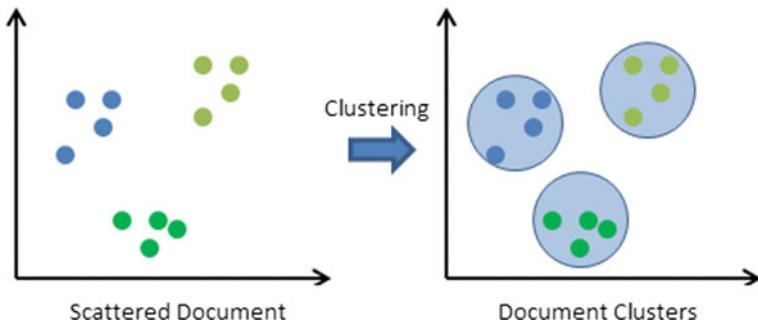


Fig. 2.6 Clustering (Source <https://www.codeproject.com/>)

function. Clustering is applied in two modes namely crisp (hard) and fuzzy (soft). In fuzzy clustering, patterns will be assigned to all the clusters based on a fuzzy membership pattern. In crisp clustering, each of the patterns are assigned to one cluster only. The clusters will be disjoint but not overlapping. K-means algorithm is a type of crisp algorithm and fuzzy c means is a type of fuzzy based clustering.

2.5.1 Various Clustering Techniques

There are basically two standard types of clustering namely partitioning algorithm and hierarchical algorithm. Partitioning algorithm specifies an initial number of groups and iteratively performs the reallocation of objects among groups to convergence. K-means is a well-known partitioning algorithm. Objects are placed into k-groups where k is chosen beforehand. Centroid for each group is calculated and each object is placed into the group based on its distance to centroid. This method reduces the overall within cluster dispersion by iteratively reallocating cluster member. Each document is treated as vector $d = [w_1, w_2 \dots w_n]$ where w_i is weight of each term in a document. The weight of term i in document j is given as:

$$w_{ji} = tf_{ji} \times \log_2(N / df_{ji}) \quad (2.13)$$

where, tf_{ji} is the number of occurrences of term i in the document j , df_{ji} indicates the term frequency in the collections of documents. Cosine distance is

$$\cos(\vec{d}_1, \vec{d}_2) = \vec{d}_1 \cdot \vec{d}_2 / \|\vec{d}_1\| \cdot \|\vec{d}_2\| \quad (2.14)$$

where, \cdot refer to ‘dot product’ and $\|\cdot\|$ refer to the norm of a vector. Norm is the magnitude of vector. The centroid vector “ m ” for each set “ S ” of documents is given by,

$$\vec{m} = \frac{1}{N} \sum_{\vec{d} \in s} \vec{d} \quad (2.15)$$

where, n denotes total number of documents within the collection. Now,

$$\vec{d}_i \cdot \vec{m} = \frac{1}{N} \sum_{\vec{d} \in s} \vec{d} \cdot \vec{d}_i = \frac{1}{N} \frac{1}{N} \sum_{\vec{d} \in s} \cos(\vec{d}_1, \vec{d}_2) \quad (2.16)$$

Find the partition C^* with near-optimal or optimal adequacy when comparing with all the other achievable solution $C = \{C^1, C^2, \dots, C^{N(n,k)}\}$ where number of partitions is given as:

$$\text{Number of partition} = N(n, k) = \frac{1}{k!} \sum_{i=1}^k (-1)^i \binom{k}{i} (k-i)^i \quad (2.17)$$

in which N is the number of documents in dataset S and k is the number of clusters. Hierarchical algorithm divides or combines the existing group to create a hierarchical structure. The pseudo algorithm can be given as:

- Each item need to be assigned to one cluster such that if there are N number of items, they will have equal number of clusters with each cluster containing only one item. The distances (similarities) in between clusters are same as distance (similarities) in between items that it contains.
- The pair of clusters that are closest (most similar) are to be merged into one single cluster such that one cluster is reduced.
- Now, calculate the distances (similarities) between the old clusters and new cluster.
- Steps 2 and 3 are to be repeated till all items are clustered on to one single cluster of size N .

Various advantages of this method are:

- It is more versatile
- Ease of handling the similarity and distance
- Embedded flexibility regarding granularity
- It can be applied to any data type.

2.5.2 Distance Measures Used in Clustering

The joining or tree clustering method uses the dissimilarities or the distance between objects when forming the clusters. This distance can be based on a single dimension or multiple dimension. The most straightforward way of computing distance between objects in a multi-dimensional space is to compute Euclidean distance. x_i and y_i are the two points between which the distance has to be calculated.

Euclidean distance This is probably the most commonly chosen type of distance. It is simply the geometric distance in the multidimensional space. It is computed as:

$$\text{distance}(x, y) = \left\{ \sum_i (x_i - y_i)^2 \right\}^{\frac{1}{2}} \quad (2.18)$$

Squared Euclidean distance One may want to square the standard Euclidean distance in order to place progressively greater weight on objects that are further apart. This distance is computed as:

$$\text{distance}(x, y) = \sum_i (x_i - y_i)^2 \quad (2.19)$$

City-block (Manhattan) distance This distance is simply the average difference across dimensions. In most cases, this distance measure yield results similar to the simple Euclidean distance. The city block distance is calculated by:

$$\text{distance}(x, y) = \sum_i |x_i - y_i| \quad (2.20)$$

Chebychev distance When two objects differ on any one of its dimension and if there is a need to define two objects as “different”, then this type of distance measure shall be suitable.

The Chebychev distance can be calculated by:

$$\text{distance}(x, y) = \max |x_i - y_i| \quad (2.21)$$

Power distance At times, it may be required to decrement or increment the progressive weight which is positioned on dimensions where the corresponding weights are different.

This is achieved by power distance. The power distance can be calculated by:

$$\text{distance}(x, y) = \left(\sum_i |x_i - y_i|^p \right)^{1/r} \quad (2.22)$$

where r and p shall be user defined parameters. If r and p are equal to 2, this distance shall then be equal to Euclidean distance.

Percent disagreement If the values for dimensions used in analysis are categorical in nature, this type of measure is suitable. The distance can be calculated by:

$$\text{distance}(x, y) = (\text{number of } x_i \neq y_i)/i \quad (2.23)$$

A cluster of data objects can be treated as one group. While doing the cluster analysis, at first partition the set of data into groups based on data similarity and then assign the label to the groups. The main advantage of clustering over classification is that, it is adaptable to changes and help single out useful features that distinguish different groups. No labelled documents are provided in clustering; hence, clustering

is also known as unsupervised learning. Document clustering is widely applicable in areas such as search engines, web mining, information retrieval and topological analysis.

2.5.3 K-Means Technique

Clustering is of two types namely hard clustering and soft clustering. In hard clustering, each data is a member of one cluster. In soft clustering or fuzzy clustering, each data may belong to multiple clusters. One of the most popularly used soft clustering approaches is the k-means algorithm. Stuard Lloyd proposed the standard algorithm for k-means in 1957. The aim of this algorithm is to find the best partition of n entities into k groups or clusters in such a way that the total distance between the group members and its centroid are minimized. It iterates between updating assignment of data to clusters and updating cluster's summarization or centers. In this technique, k centroid is initialized. These centers are chosen at random initially. Then, each point is assigned to the nearest centroid by calculating the Euclidean distance between the centroid and the data point. Euclidean distance is the geometric distance in the multidimensional space. It is computed as:

$$\text{distance}(x, y) = \left\{ \sum_i (x_i - y_i)^2 \right\}^{\frac{1}{2}} \quad (2.24)$$

where x_i, y_i are the two points between which the distance has to be calculated. Now, new centers are calculated for each group. If there is a change in the centroid assignment, again the distance from each data to the new centroid is calculated. These are iteratively performed until there is no change in the centroid. Suppose there are k clusters and C is the centroid where $C = 1, 2, 3 \dots k$. Let the data set be X where $X = x_1, x_2, \dots, x_n$ and set of centers be denoted as V where $V = V_1, V_2, \dots, V_c$. The centroid of the cluster is calculated as:

$$V_i = \frac{1}{c_i} \sum_{i=1}^{c_i} x_i \quad (2.25)$$

where, c_i is the number of data points in the i th cluster. The basic algorithm for k-means is given as below:

Select k points as the initial centers.

Repeat until centers do not change:

Form k clusters by grouping data near the centroid to its corresponding cluster.
Recalculate the centroid for each cluster.

End loop.

Closeness of the data to the centroid is usually calculated using the Euclidian distance. K-means algorithm is easy to implement and computationally fast. Differential Evolution technique is a simple heuristic technique for global optimization proposed by Storn and Price [3]. Combining k-means algorithm with Differential evolution stabilizes the local and global search. K-means algorithm finds its application among various fields like feature learning, cluster analysis and vector quantization. The evaluation of cluster quality is done by internal quality, external quality, entropy, F measure and overall similarity. External quality evaluates how good clustering is working by comparing the groups created by clustering to known classes. Internal quality compares different set of clusters without any reference to external knowledge. There are three types of cluster similarity namely single link, complete link and group average. Single link is the measure of similarity of the most similar members. Group average is the measure of average similarity between members. Complete link is the measure of similarity of two least similar members. The various applications of clustering are:

- Finding similar documents.
- Search optimization.
- Detecting duplicate contents.
- Recommendation system.
- Organizing collection of large documents.

The flow chart for K-means algorithm is given below in Fig. 2.7.

2.6 Image Segmentation

Partitioning a digital image into multiple segments (sets of pixels) is referred to as image segmentation. The goal of segmentation is to simplify and/or to change the representation of an image into something that is more meaningful and easier to analyze. In areas like the medical imaging, satellite imaging, and security purposes, image segmentation plays a vital role in better understanding about any particular section. Its applications vary from satellite imaging (for locating roads, forests etc. in a satellite image) to medical imaging (for locating tumors, for analyzing the anatomical structure etc.) from machine vision to fingerprint recognition and so on. In fact, segmentation can be applied to study in detail any real life problem where digital image can be developed. However, it is considered to be quite a demanding task because of the presence of multiple objects in an image and sometimes due to the intrinsic nature of an image. An example of image segmentation is shown in Fig. 2.8.

Image segmentation can be classified into edge based, region based, model based, feature based clustering and threshold based. Edge based method (see Fig. 2.9) uses the edge detection technique where, the edge of an image or a section within the image is identified by detecting the discontinuity of brightness across the edges. It is an image processing technique to find the boundary of objects within the image. Edges consist of meaningful information and features. In this method, an edge filter

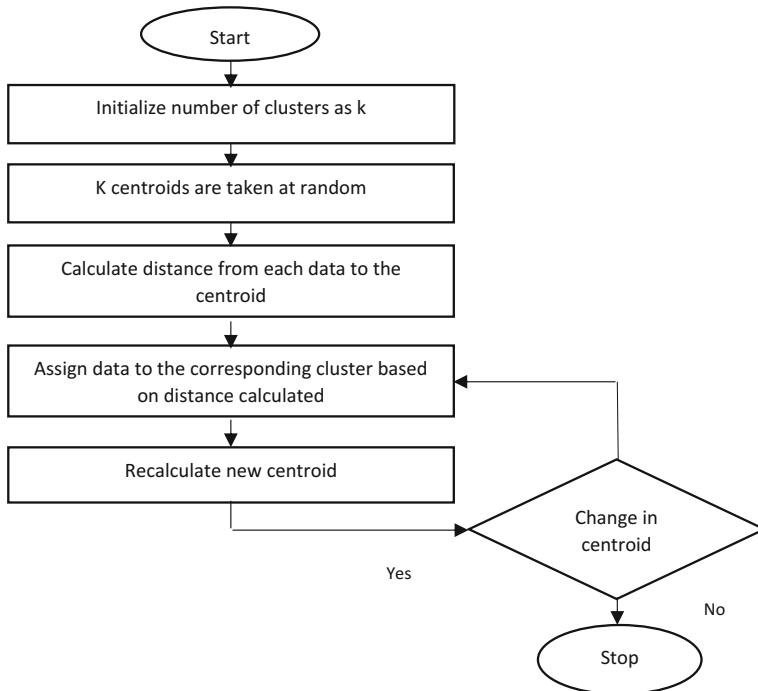


Fig. 2.7 Flowchart for k-means clustering

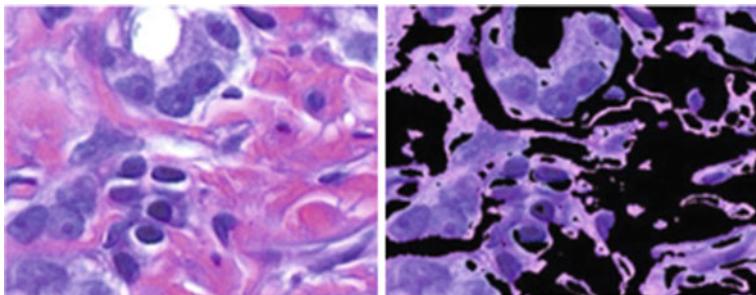


Fig. 2.8 Image segmentation

is applied on an image. The pixels will be classified as edge or non-edge type based on filter output. Pixels, which are not separated by edge, will be applied to same category. Various edge detection algorithms used are Roberts, Canny, Sobel, Prewitt and fuzzy logic techniques.

Region based method (see Fig. 2.10) works iteratively by grouping nearby pixels having similar values and by breaking groups of pixels having dissimilar values. Region is the group of pixels with similar properties. In this approach, neighboring



Fig. 2.9 Edge based segmentation

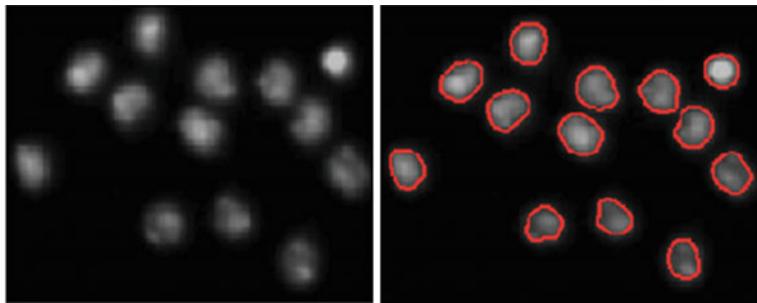


Fig. 2.10 Region based segmentation

pixels of initial seed points are examined to check if the pixel neighbors should be added to the region. This process is iterated till segmentation is completed. This technique is a combination of region merging and region splitting. The main principles used in this approach are similarity and spatial proximity.

Model based segmentation (see Fig. 2.11) is also called Markov Random Field (MRF) based segmentation. In model based image segmentation, the image of interest is supposed to have a repetitive geometry. So a probabilistic model can be considered for explaining variation in shape of the region and so during segmentation, image apply constraints using this model as prior. These tasks involve probabilistic representation of variation of registered sample, statistical inference between image and model and registration of training samples to common pose. Here, an inbuilt region constraint is given to the model used for segmentation of color image. Color pixel components are considered as independent random variables. This method is merged with edge detection for defining the edges correctly. It has spatial region smoothness constraints which are related to other color components. This is faster than traditional approach. The process goes in an iterative manner. Resolution based segmentation is done only to a part of image, making the process fast.

Feature based segmentation (see Fig. 2.12) technique looks out for features like image edges, corners etc. and traces them while moving from frame to frame. This

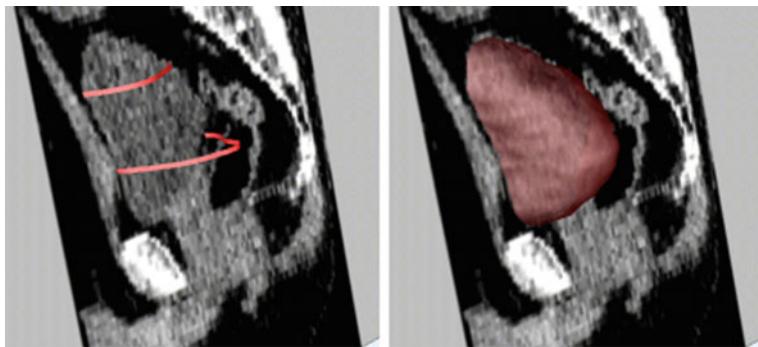


Fig. 2.11 Model based segmentation



Fig. 2.12 Feature based segmentation



Fig. 2.13 Block diagram for thresholding

technique is followed in two stages. In the first stage, features are derived from two or more images. This technique reduces the workload and gives a higher level of understanding of the image by removing the unwanted regions. In the second phase, features are matched in frames. Features from one frame are used as seed points to be used by other methods to find flow. The main aim of this technique is to separate objects of similar texture patterns from image based on features like contrast, gray-level occurrence etc.

Thresholding (see Fig. 2.14) is the basic technique of image segmentation in which a binary image can be created from a grayscale image. The main aim of thresholding is to remove those pixels from the image which represents an object (see Fig. 2.13). Binarization is the process to label pixels that categorize the foreground regions with



Fig. 2.14 Thresholding

single intensity and background with different intensities. It separates dark and light regions.

There are two key methods available for colour image thresholding namely histogram based method and neighbourhood based method. Histograms are created by splitting range of data into same size classes. For colour images, three RGB values are given for each pixel. In histogram based method, a 3D histogram will be constructed for each colour values and threshold values are plotted. Objects are identified by assigning random pixel values to regions separated by threshold. There are two types in histogram called bimodal histogram and multimodal histogram. In bimodal histogram, there are two peak values. In multimodal histogram, there are multiple peak values. In neighbourhood based technique, thresholding is done based on neighbourhood similarity. Thresholding can be bi-level or multilevel. In bi-level thresholding, image is divided into two regions. Threshold value is denoted by T . Object pixels contain pixels with gray value greater than T and background pixels consist of gray values lesser than T . In multilevel thresholding, image is separated into various distinct regions. This technique considers more than one threshold for the image and separates image into certain brightness regions. The output image has one background and several object pixels. This technique works well for complex or colored images.

Despite the fact that several methods have been suggested in literature for segmentation process, researchers are still trying to develop efficient and robust algorithms, which can meet effectively the challenges of segmentation. Because of its simplicity and stability, thresholding has become one of the most popular image segmentation methods. Proper threshold values should be assigned to optimize a criterion such as entropy or between-class variance to produce a successful segmentation. Thresholding is a base for computer vision and object recognition. Unlike gray images, segmenting color images are complicated and time consuming. Image segmentation can be modeled as an optimization problem for which suitable techniques are needed for attaining global solution. These problems are classified as unsupervised learning and such issues can be solved using evolutionary algorithms like DE, PSO, FPA etc.

2.7 Work Done on Metaheuristics

Storn and Price [3] proposed DE as a simple and fast technique for optimization. Since then a lot of work has been done in this area. Pahner and Hameyer [6] designed a new approach for DE where the optimization scheme was characterized by adaptive and direct coupling of the algorithms. This scheme greatly reduced the function calls thereby reducing the computational expenses. Abbass et al. [7] designed a novel Pareto-frontier Differential Evolution algorithm (PDE) to solve multi objective problems. The solution from the new algorithm was better than the existing method. Zaharie [8] showed the relationship between the control parameters of DE algorithm and the evolution of population variance. Through this concept, the behavior of DE could be curbed by controlling population diversity through some appropriate choice of parameters. Madavan and Bryan [9] extended the Differential evolution approach to a multi objective optimization problem using a Pareto based approach. The result is a simple strategy that is self-adaptive and maintains the diversity of Pareto set. Babu and Jehan [10] used the Differential evolution approach to solve a multi objective optimization problem with two objective functions and a classical HimmelBlau function. In both the approaches, the results obtained were better. Xue et al. [11] devised a newly developed evolutionary approach—Pareto based Multi Objective Differential Evolution (MODE). The concept of differential evolution algorithm was extended for multiobjective problem domain. Fan and Lampinen [12] developed trigonometric region based mutation operation where the trigonometric region was devised to limit the generated mutant vector. Iorio and Li [13] showed that the self-adaptive technique of Differential evolution can be used for solving multiobjective optimization problems whose parameters are interdependent. Non-dominated sorting Differential Evolution (NSDE) was a combination of Differential Evolution with NSGA II. NSDE demonstrated superior performance and rotational invariance over traditional algorithms. Rene et al. [14] improvised the DE algorithm with the crowding scheme. This scheme could track and maintain the local optima. Qin and Suganthan [15] proposed a novel self-adaptive Differential evolution (SaDE) in which the control parameters need not be specified. The performance was evaluated and compared. Robič and Filipič [16] developed a multiobjective optimization based DE (DEMO). This algorithm combined the advantages of DE with the techniques of crowding distance sorting and Pareto based ranking. Krishnanand and Ghose [17] developed glowworm swarm optimization algorithm (GSO). It uses the capability of glow worms to change the intensity of luciferin emission and thus to glow at different intensities. The two significant parts of the algorithm are:

- (a) In the first part, GSO makes the agent to glow at different intensities proportional to the function value being optimized.
- (b) In the second part, algorithm utilizes a dynamic decision range by which effect of distant glowworms are reduced when glowworm has sufficient number of neighbors.

The second half of the algorithm differentiates the algorithm from other evolutionary algorithms. This step allows the group of glowworms to automatically subdivide

into subgroups which can then converge to multiple local optima simultaneously. This property of GSO allows the algorithm to be used in identifying multiple peaks of multi modal function. Chakraborty et al. [18] devised an improved variant of classical DE2 scheme by taking the concept of local neighborhood of each vector, which improves exploration and exploitation ability of DE without using any additional function evaluation. Efren et al. (2006) did a comparative study on variants of Differential evolution on global optimization. After the study, the best strategy among different variants was obtained. Kim et al. [19] gave some modifications to the differential evolution strategy for the constraint global optimization problem. This method was applied to the design of gas circuit breaker to increase small current interruption performance. Zhang et al. [20] studied that DE is mostly based on crossover probability and mutation factor. Changes done to these parameters will affect the performance of DE. Noman and Iba [21] proposed a crossover-based adaptive local search operation for improving the performance of standard DE. Determining a single local search (LS) length for serving wide set of problems is a major concern. A Local Search (LS) technique using the concept of hill climbing was developed to solve the problem of determining a single LS length by adaptively adjusting the length of search. Rahnamayan et al. [22] presented a novel algorithm to accelerate differential evolution. This method employs opposition-based learning for population initialization and generation jump.

Yang [23] developed the Firefly algorithm for multimodal optimization problems. Flashing behavior of fireflies was the inspiration for this metaheuristics algorithm. The purpose of firefly flash is to aid as a signal system to attract other fireflies. The algorithm uses the following assumptions:

- (a) All fireflies are unisexual, thereby any individual firefly will be attracted to all other fireflies.
- (b) Attractiveness is proportional to brightness but brightness decreases as mutual distance increases.
- (c) If there are no fireflies brighter than a given one, it will move randomly.

This algorithm is a nature inspired metaheuristic optimization algorithm. The objective function is associated with brightness. It is a variant of PSO algorithm.

Qin et al. [24] proposed a self-adaptive DE algorithm where the control parameters and trial vector generation strategies are gradually self-adapted by learning from previous experience. This method eliminated time-consuming search for trial vector generation and control parameters. Qin et al. [24] proposed a self-adaptive DE (SaDE) algorithm. Here, associated control parameter value and trial vector generation strategies are self-adaptive by learning from the previous experiences in generating good results. Das et al. [25] developed a family of improved variants of DE/target-to-best/1/bin strategy, which uses the concept of neighborhood of each population member. It also investigated the application of the variant to real life problems having parameter estimation for frequency modulated sound wave and spread spectrum radar poly-phase code design.

Jeyakumar and Velayutham [26] gave an empirical, comparative performance analysis of fourteen variants of DE and Dynamic Differential Evolution (DDE). DDE

is used to solve unconstraint global optimization problems. DDE was compared with DE and a comparative analysis was done to identify which variant performs well on problems with different systems. DE responded to population progress with a time lag. Whole population in DE remains unchanged until a new population replaces it. It results in slower convergence. To remove this problem, a dynamic version of DE was introduced. It updates both the current optimal individual with new competitive individual and non-optimal individual dynamically. Here, creation of every NP trial vectors is considered as one generation of DDE. There are two schemes of crossover used: binomial and exponential. The new variant DDE performed better than the conventional DE technique. Zhang and Sanderson [27] developed a new DE algorithm called JADE to improve optimization performance by implementing a new mutation strategy DE/current-to- p best with optional external archive and by updating the control parameters in an adaptive manner. The simulation results showed that JADE was better compared to the adaptive DE algorithms. Gong et al. [28] proposed a hybrid DE with biography-based optimization (BBO). DE/BBO combined the exploration of DE with the exploitation of BBO and could generate the required candidate results. Takahama and Setsuko [29] proposed the ϵ -constrained differential evolution (ϵ DE) combining Differential evolution with ϵ -constrained technique. This algorithm has faster execution and high quality results. Das and Suganthan [30] did a survey on Differential Evolution. It was stated that DE operates with similar computation steps as employed members with scaled differences of randomly selected and distinct population members. So no separate probability distribution has to be used for generating the offspring. Evolution computation uses iterative progress of growth and development in population. Various variants of DE are implemented for specific application of numeric optimization. Various drawbacks of DE are elaborated. Firstly, DE faces difficulties on function that are non-linearly separable. Mutation strategy of DE lacks sufficient selection pressure when appointing target and donor vectors to have satisfactory exploitative power or non-separable function. Research has to be done to make DE robust against strong inter dependency of search variable. DE has limited ability to move its population large distances across search space if population is clustered in limited portion of it.

Wang et al. [31] did a study on improving the performance of DE by combining the trial vectors strategies with control parameter settings. They also proposed a method Composite DE (CoDE) using three trial vectors and three control parameters. Qu et al. [32] developed neighborhood based differential evolution in which the mutation is performed within each Euclidean neighborhood. The mutation neighborhood maintains the multiple optima found during evolution and progress towards the respective global/local optimum. Das et al. [30] proposed a new mutation strategy of fitness induced parent selection scheme for binomial crossover of DE. It uses the best of a group of randomly selected solutions from current generation to disturb the target vector. In the new technique of recombination, a biased method of parent selection scheme is used by allowing each mutant to undergo usual crossover with one of few top ranked individuals from the current population. Elsayed et al. [33] gave an improved Differential evolution that uses a combination of differential mutation operations. This algorithm permits a covariance adaptation matrix evolu-

tion strategy algorithm for local search. Reed et al. [34] developed a method to adjust both the crossover rates and mutation during optimization in a way that increases the convergence rate to desired solution. Performance is shown on challenging problem of identifying imperfections in submerged shell structures. Juárez-Abad et al. [35] gave the results of implementation of differential evolution algorithm on FPGA using floating point representation with double precision useful in real numeric problem. Verilog Hardware Description Language (VHDL) was used for Altera hardware design. Schematics of modules of differential evolution algorithm were presented. Performance of design is evaluated through six different function problems implemented in hardware. Implementation of Differential Evolution Algorithm (DEA) on software is used in applications where an optimization of parametric model is carried out in conventional computer equipment. This method uses a number of function variables and population size using double precision floating point representation. DEA finds global optimum of function over a continuous space. FPGA (Field Programmable Gate Array) is a device that is used to design a dedicated digital system or embedded platform that perform specific task in a system. Altera FPGA is a device with programmable logic blocks and memory elements, which are interconnected to perform complex combinatorial and sequential functions. Altera provides a free library of parameterized intellectual property block called megafunction. These floating point megafunction implements hardware modules for performing customized floating point operations. DEA performs floating point operations only for generating the offspring individuals in mutation and crossover process and so needs only one module for floating point. This design does not exploit parallelism approach as this technique depends on specific application.

Yu et al. [36] proposed an Adaptive DE (ADE) with two level adaptive parameter control scheme and a new mutation strategy. This technique has the advantage of balancing between fast convergence and population diversity. Cai et al. [37] devised a new method called adaptive direction information based ND_iDE (aND_i-DE). By using this technique, a good balance between exploration and exploitation can be dynamically obtained. The method was applied to DE/rand/1 algorithm and was shown to be efficient. Later Cai et al. [38] extended it to two adaptive operator selection (AOS) to adaptively select the most appropriate type of direction information for specific mutation strategy during evolutionary process. Gong et al. [39] devised the adaptive ranking mutation operator (ARMOR) for differential evolution. This technique is expected to make DE achieve feasible solutions faster and converge faster. This technique is simple and can be easily combined with most of constrained DE variants.

Guo et al. [40] proposed an Eigen vector based crossover operator. This method utilizes Eigen vectors of covariance matrix of individual solutions, which make the crossover rotationally invariant. The method showed improved performance of DE. Alam et al. [41] introduced a DE variant called DE with alternating strategies (DE-AES). This effectively combines the exploitative and explorative characteristics of five different DE variants by randomly alternating and executing these DE variants into one single algorithm. The experimental results showed better performance. Guo et al. [42] proposed a self-adaptive differential evolution with global neighborhood

search (NSSDE). In this, the control parameters are self adaptively tuned based on feedback from search process. The global neighborhood search strategy increases speed of execution. Qiao and Grantham [43] gave an algorithm by combining the heuristics with modified Differential evolution algorithm for situations where the communication range value is unknown. His proposed method gave better results compared to other nonconvex optimization techniques.

Sakr et al. [44] proposed an adaptive differential evolution procedure and this was presented to solve optimal reactive power dispatch problem. Zaheer et al. [45] proposed a strategy named DE/rand-to-best-best/2. It makes use of an additional parameter called guiding force parameter K, which takes a value between (0,1) besides using the scaling factor F, which has a fixed value. The result shows that this strategy performs well in comparison to other mutation strategies of DE. Zhang et al. [46] proposed a multiple variants coordination (MVC) framework with two mechanisms, namely, the multiple variants adaptive selecting mechanism and the multiple variants adaptive solutions preserving mechanisms (MV-APM). The best performing optimizer was determined and then utilized in EG in the same segment. Qian et al. [47] proposed an improved binary DE (IBDE) algorithm for optimizing PWM control laws of power inverters. The proposed algorithm focused on the designs of the adaptive crossover and parameter less mutation strategies without imposing an additional computational burden. Wu et al. [48] investigated the high-level ensemble of multiple existing efficient DE variants and a multi-population based framework (MPF) was proposed to realize the ensemble of multiple DE variants to derive a new algorithm named EDEV.

Das and Suganthan [30] stated various techniques of differential evolution. They developed various new areas of work in DE and its applications were listed. But, functions that are linearly separable cannot apply DE algorithm. In addition, more work needs to be done to make DE robust. There is also a limited possibility when trying to move its population over large space. DE faces many difficulties on functions that are non-linearly separable. DE has limited ability to move its population in large distances across search space if its population is clustered within a limited portion of its population. Probabilistic convergence, martingale theory, drift analysis, stochastic Lyapunov energy function of DE are all open issues having huge scope for research. More research work also need to be done on integrating opposition number based initialization and generation jumping with self-adaptive DE variants for improving the performance.

FPA algorithm being practically new has been applied only for few problems. This algorithm was basically developed to be applied to optimization problems. Yang et al. (2012) extended single objective flower optimization algorithm to solve multi objective optimization problems. Tests indicated that this multi objective FPA was efficient with rapid convergence rate. Kaur et al. [49] proposed a new technique based on pollination to segment color images. This algorithm was implemented for extracting optimal clusters. Kaur and Singh [50] also developed a new technique based on color image segmentation. It was implemented for extraction of optimal clusters from colored images. This algorithm was better in terms of colored image and time to segment images.

Kaur et al. [49] created a new technique for better fractal image compression through pollination. The visual image was better with PBO (Pollination Based Optimization). Abdel-Raouf et al. [51] developed a hybrid of FPA called FPCHS (Flower Pollination Algorithm with Chaotic Harmony Search). As it was developed to increase the speed of searching, it was applied to Sudoku problems. It was more efficient and accurate compared to harmonic search algorithm. The encoding time and overall performance were shown to be better than the other optimization techniques. The idea also concluded that it was suited in areas of application requiring fast access to high quality images. Abdel-Raouf and Abdel-Baset [52] also developed hybrid Flower Pollination with Particle Swarm (FPPOS) to solve constraint optimization problem. Results were compared and the algorithm was found to be accurate, efficient, reliable and more superior to existing algorithms. Wang and Zhou [53] proposed FPA with dimension-by-dimension improvement. This algorithm used local search approach to improve searching. The new technique improved the quality of solution and convergence speed. Nguyen et al. [54], proposed a strategy for parallelized Flower Pollination Algorithm for solving numerical optimization problems. In this method, the population flowers are split into several independent groups based on original structure of Flower Pollination Algorithm. This method provides the information flow for flowers to communicate in different groups. Zhou et al. [55] proposed an Elite Opposition based Flower Pollination Algorithm (EOFPA). The improved method involves two major optimization strategies. Global elite opposition-based learning improves the diversity of the population, and the local self-adaptive greedy strategy improves its exploitation ability. This method has a fast convergence speed and a high degree of stability.

2.8 Application of Metaheuristics on Data Clustering

Applying the concept of evolutionary computing to clustering problem has been one of important topic of research for a long time. Numerous variants of evolutionary algorithms like DE, GA, PSO were created and these variants were applied to the clustering problems. Paterlini and Krink [56] gave a performance comparison on genetic algorithm (GA), PSO and Differential Evolution (DE) for a medoid evolutionary clustering approach. Medoid is an object of cluster, which has minimal average dissimilarity to all objects in the cluster. The results show that DE approach was far superior compared to GA and PSO and that DE should be considered over the other algorithms for clustering. Zaharie [57] studied the applicability of crowding Differential evolution to unsupervised clustering. This approach allows the identification of clusters with arbitrary shapes by using multi center descriptions for them. Lu [58] gave a new evolutionary algorithm that applies on an entropy-based principle, which properly estimated optimum number of clusters that will be present in a dataset. Here two sets of data namely (a) synthetic data and (b) standard Irish data set are used for algorithm validation. Further this can be extended for application in network intrusion detection. For a clustering algorithm like k-means, if the initial

partition is not chosen correctly then clustering will not be effective. Another clustering technique that faces this issue is Gaussian Mixture Model (GMM). It is built with assumption that the data on which clustering should be performed is put into one of the Gaussian distributions. Several statistical methods were used to estimate number of mixing components present in GMM technique. Since this technique may stop doing additional search when corresponding criteria reaches a threshold, these techniques are likely to lead to local optima. Now evolutionary computation does not lead to local maxima as the search space can be extended by optimization technique and genetic selection. In this new algorithm, three genetic operators namely (a) splitting, (b) merging and (c) deletions produce the new individual. The algorithm performs expectation maximization steps to maximize the likelihood along with accelerating the speed and also ensuring the fitness to the model and data. Three of performance metrics were considered. They are (a) Average number of Evaluation on Success (AES), (b) Mean Best Fit (MBF) and (c) Success Rate (SR). SR is the measure of number of times algorithm successfully meets the global optima. AES counts the speed of optimization. MBF measure how much close solution is to the global optimum. Based on these measures, the test showed that this algorithm has the ability of converging to a global optimal solution having a large merger probability and low splitting probability. Here synthetic data set and a standard Irish data set was used to validate the algorithm. This algorithm can further be prolonged to be used in intrusion detection. When applying clustering to intrusion detection, two clusters are used namely intrusion cluster and normal cluster. Here if all clusters are normal, then the error rate will be high. So to improve performance of intrusion detection, accurate estimation of clusters is required before clustering.

Martínez-Estudillo et al. [59] proposed hybrid evolutionary algorithm to solve nonlinear regression problem. The algorithm was a mixture of clustering processes, an evolutionary algorithm and local search procedure where, only some individuals were subject to local optimization. This was a good compromise between computational cost and performance. A local search was incorporated to improve search. In cluster analysis technique, the clusters are closely placed points that will correspond to area of attraction. Thus the local search procedure is performed once in every such area. The main advantage of the methods used here is that the computational cost does not affect the time spent on algorithm. Depending on cluster partition and local searches, two versions of hybrid evolutionary algorithm was developed. Abraham et al. [60] describes a novel approach for partitioning text document into clusters using an improved version of classical differential evolution. A modified mutation scheme was introduced to improve convergence properties. This modified DE was then used for clustering text document for retrieving important information. A new validation index was also proposed for high dimensional document clustering problems by modifying the CS measure. This technique when applied to clustering was found to be superior in speed and quality. Alves et al. [61] gave an improved version of evolutionary algorithm for clustering called F-EAC (Fast Evolutionary Algorithm for Clustering). Here, the influence of fitness function was reduced in the assessment process. The work was further extended to optimize fuzzy clusters in 2007.

Alves et al. [61] used a fuzzy cluster validity criterion and a fuzzy based local search algorithm named EAC-FCM. Fuzzy clustering algorithm requires the number of clusters to be known beforehand. This can be made possible by executing fuzzy clustering algorithm repeatedly for different cluster numbers and then choosing a particular cluster number, which gives best result in accordance to a specific cluster validity criterion. Fuzzy c-means (FCM) is a fuzzy extension of the k-means algorithm. Let $x_j \in \mathbb{R}^n (j = 1, 2, \dots, N)$ be the data objects for clustering into c clusters, $v_i \in \mathbb{R}^n (i = 1, 2, \dots, N)$ be cluster prototypes, p_{ij} be the membership of the j th object of the i th fuzzy based cluster, then the algorithm is given as:

- Select number of fuzzy cluster c .
- Select initial cluster prototypes $v_1, v_2 \dots v_c$.
- Compute the distance using Euclidian formula.
- Compute fuzzy partition matrix.
- Update cluster prototypes.

The manuscript presented that EA for fuzzy based clustering will be more efficient than the classical approaches if the cluster number is unknown. Zhang et al. [20] showed that implementation of DE is mostly based on crossover probability and mutation factor. Changes done to these parameters will affect the performance of DE. Zhang et al. [62] also introduced a genetic algorithm to solve the clustering aggregation problem. Here the key of this algorithm is the coding clustering division where this clustering division is the chromosome of genetic algorithm. The clustering division is coded as a bit string. Integer coding for the individual and operations of selection, mutation are for further extension. The performance of cluster division is estimated by defining cluster precision and its features. Clustering ensemble learning has two parts namely individual production and combination of individuals. For individual production, different tagged individual production is developed. For combination of individuals, the tagged individuals are integrated through voting. Zhang et al. [63] also proposed an advanced PSO and differential evolution method for spatial clustering with obstacle constraints (SCOC). The proposed method shows better quantization error and constringency speed. Das et al. [64] elaborates an application of DE for automatic clustering of large unlabeled data sets. The proposed algorithm requires no prior knowledge of data for classification. It calculates the optimal partition of data on the run. Clustering of data is of two types: crisp and fuzzy. In crisp clustering, the clusters are disjoint and non-overlapping in nature. Any pattern may belong to one and only one class. In fuzzy clustering, a pattern may belong to all class with a certain fuzzy membership grade. In the technique developed by Das, crisp clustering is used. In evolutionary approach, clustering of data set is viewed as an optimization problem and is solved by using an evolutionary search heuristics approach such as genetic algorithm. Indrajit et al. [65] proposed an application of differential evolution to fuzzy clustering for categorical data sets. The proposed algorithm effectively optimizes the fuzzy-medoids error function globally.

Zheng et al. [66] introduced a technique of unsupervised evolutionary clustering used for mixed data types. K-means algorithm is usually used for mixed data types. It is sensitive to initialize and converge to local optimum easily. An advantage for

evolutionary algorithm is the global searching ability. This method is more robust than the traditional k-means approach. Fuzzy logic introduced the concept of partial set membership as a way of handling imprecision in mathematical modelling. This concept was applied to clustering. Rough set theory is used for classification problems where prior group membership is known and results are expressed in terms of rules for group membership. Pawlak [67] introduced the concept of rough or approximation sets. Rough set is based on the assumption that with every object of information system, there is a certain amount of information. This information is expressed by means of attributes used as descriptions of objects. Data is treated for perspective of set theory and none of the traditional assumptions of multivariate analysis are relevant. In rough set, objects have same knowledge from equivalent relation. In lower approximation of rough set, it contains objects that are fully in the subset. In upper approximation, objects may or may not be in the subset. Rough clustering is defined with an upper and lower approximation, which allows multiple cluster membership for objects in dataset. Already existing algorithm for clustering had the issue of large number of clusters and uncertainty in lower approximation whether each cluster will give the most efficient data set coverage. Rough clustering was different from k-means in its concept of multiple cluster membership of objects. Rough clustering produces more number of clusters compared to k-means. Evolutionary algorithm is used in lower approximation of rough cluster to provide maximum coverage of data set using minimum number of clusters. Fitness measure in rough clustering is performed by:

- a. Maximum dataset coverage
- b. Minimum number of template in cluster solution
- c. Maximum accuracy of each template.

Template is the basic data structure for describing rough cluster. Coverage is the total of cardinal values of lower approximation of each template of cluster solution divided by cardinal value of full data set. Accuracy is the sum of cardinal value of lower approximation divided by cardinal value of upper approximation for templates in cluster solution. For the mutation and recombination operation, a multipoint operation was used. Size of offspring was randomly selected from size of parent. This rough cluster permitted an object to belong to multiple clusters. More research has to be done to include generalized dataset and to compare this with other data sets.

Maulik and Saha [68] devised a modified differential evolution (DE) based fuzzy c-mediod (FCMdd) clustering of categorical data set. This technique shows the superiority of integrated clustering and supervised learning approach. Maulik and Saha [69] also proposed a new real-coded modified differential evolution based automatic fuzzy clustering algorithm that automatically calculates the number of clusters and the proper partition from a data set. In this paper, the assignment of points to different clusters is based on a Xie-Beni index which, considers the Euclidian distance. Zheng et al. [66] suggested an unsupervised evolutionary clustering algorithm for mixed data type named Evolutionary k-prototype algorithm (EKP). They compared the traditional k-prototype (KP) algorithm. KP is applied as local search strategy and works on EA framework. The study from this paper shows that EKP is more robust and

gives better results than KP for mixed type data. This algorithm uses square Euclid distance for numerical attributes and Hamming distance for categorical attributes. This study can be extended to find a more flexible similarity measure.

Alguliev et al. [70] proposed a document summarization model which separates the key sentences from the given document while removing the redundant information in the summary. The results show that the proposed method was superior to the earlier summarization models. Pham et al. [71] introduced a new approach to cluster datasets of mixed data type. RANKPRO (random search with k prototype) combined the bee algorithm with the k prototype. RANKPRO algorithm proved to be more efficient than the k prototype approach. Hatamlou [72] devised a new heuristic method inspired from the black hole phenomena. The experimental results showed that the technique outperformed the existing classical methods. This method was applied to the field of clustering. Suarez-Alvarez et al. [73] introduced a unified statistical approach to normalize all attributes of mixed datasets. Clustering of several standard datasets are also performed in this paper. Qu et al. [32] gave a neighborhood mutation strategy and combined it with various niching differential evolution (DE) algorithms to solve multimodal optimization problems. This technique has faster convergence with higher accuracy. The mutation strategy in this technique was able to generate a stable niching behavior and was able to locate and maintain multiple global optima. Voges and Pope [74] proposed an evolutionary based rough clustering algorithm which did not require specifying the number of clusters in advance and was independent of initial starting point. It gave an overview on rough clusters and on evolutionary algorithm for development of viable cluster solution containing optimum number of template that provides description of clusters. Both in k-means and k mode method, the number of clusters will be defined in the beginning. Fuzzy logic presented the idea of partial set membership as a way to manage imprecision in mathematical modelling which were applied in clusters. Evolutionary algorithm is used in lower approximation of rough cluster to provide maximum coverage of dataset using minimum number of clusters. Fitness measure in rough clustering is performed by (i) maximum dataset coverage (ii) minimum number of templates in a cluster solution (iii) maximum accuracy of each template. Template is basic data structure that describes a rough cluster. Coverage is total cardinal value of lower approximation of each template of cluster solution divide by cardinal value of full data set. Accuracy is sum of the cardinal value of lower approximation divided by the cardinal value of upper approximation for the template in a cluster solution. For the mutation and recombination operation, a multipoint operator was used. Size of offspring is randomly selected from size of parent. Rough clustering permits an object to belong to multiple clusters. More research has to be done to include generalized dataset and to compare with other dataset.

He et al. [75] proposed an H-K clustering algorithm. Traditional H-K algorithm can solve randomness and apriority of initial centers of k-means clustering algorithm. It will lead to dimensional disaster when applying to high dimensional dataset clustering due to its high computational complexity. So to remove these drawbacks, a new technique was introduced to improve the performance of traditional H-K clustering in high dimensional datasets. The new algorithm was named Ensemble principle com-

ponent analysis Hierarchical K-means clustering. Here, the high dimensional dataset is mapped into a low dimensional space using PCA method. Saha and Bandyopadhyay [76] devised a new Multi Objective (MO) clustering technique (GenClustMOO) which can automatically partition data into appropriate clusters. The effectiveness of the method was compared against k-means and single linkage method. Singh et al. [77] gave a solution to clustering after analyzing and removing the drawbacks of Euclidean distance and point symmetry based distance measures and merging the improved versions into one method to get best of both methods. This method speeds up the computation time. Ameryan et al. [78] gave a novel clustering method based on Cuckoo Optimization Algorithm (COA), which is inspired by, the nesting behavior and immigration of cuckoo birds. Here, an individual cuckoo represents a candidate solution, which consists of cluster centroid. Fitness function calculates sum of intra cluster distance. Here, three approaches differing in initial step of original COA algorithm was proposed. In Random COA clustering, the initial population is produced randomly. In Chaotic COA clustering, Chaotic Arnold's Cat map is used to produce initial population to cover whole search space. In K-means COA clustering, K-means algorithm is used to produce initial cuckoos. Cuckoo algorithm is based on the aggressive egg laying and immigration pattern of cuckoos. COA estimates the global optima approximately. Here, a cuckoo and an egg present a candidate solution of optimization problem. Cuckoo lives in group. The cuckoos are divided into given number of clusters using k-means. Cuckoos migrate to better places. In COA, average fitness of each cluster is calculated and the best point of cluster is placed as destination point. Considering the interval distance between current position and destination position, cuckoo moves towards this point.

Thein and Khin [79] proposed differential evolution for clustering and compares its purity with k-means algorithm. The results were tested on medical datasets of Pima, Liver and Heart from UCI data repository. According to the results obtained, DE outperformed the k-means algorithm for medical datasets. This work shows that DE performs better when robust clustering is needed. This work also eliminates the disadvantages of k-means technique. Ozturk et al. [80] studied the improved binary artificial bee colony algorithm and applied it on dynamic clustering. The uncapacitated facility location (UFLP) problem was dealt with the similarity measure based discrete artificial bee colony (ABC) algorithm. The discrete ABC depends on the similarity measure between binary vectors. The algorithm is modified by two improved selection scheme through genetic operators in the process of new solution generation. It uses an improved crossover function and a swap operator. In swap operation, it exchanges information between objects such that two positions of 0 and 1 are switched between each other. Solution is generated with respect to each similarity case and then crossover and swap are applied. Performance is analyzed based on dynamic clustering. This algorithm satisfies the optimum number of clusters and well obtained quality values. Mukherjee et al. [81] gave a modified version of Differential Evolution for solving dynamic optimization problems (DOP) efficiently. The algorithm was named as Modified DE with Locality induced Genetic Operators (MDE-LiGO) and it integrates changes in three stages of classical DE framework. Wu et al. [82] devised a multi-population based approach to achieve a unit

of multiple strategies. The resulting new variant named multi-population ensembles DE (MPEDE) consist of three mutation strategies.

Cheng et al. [83] introduces a novel optimization model, named as Fuzzy Clustering Chaotic-based Differential Evolution for solving Resource leveling (FCDERL). Fuzzy Clustering Chaotic-based Differential Evolution (FCDE) is developed by integrating original Differential Evolution with fuzzy c-means clustering and chaotic techniques to tackle complex optimization problems. Experimental results revealed that the new optimization model is a promising alternative to assist project managers in dealing with construction project resource leveling. Hancer and Karaboga [84] introduces to the concepts and review methods related to automatic cluster evolution from a theoretical perspective. Saha and Das [85] developed two bio-inspired fuzzy clustering algorithms by incorporating the optimization techniques, namely differential evolution and particle swarm optimization. The paper presented a comparative analysis of some metaheuristic-based clustering approaches, namely newly proposed two techniques and the already existing automatic genetic clustering techniques, VGAPS, GCUK, HNGA. The obtained results were compared with respect to some external cluster validity indices.

2.9 Application of Metaheuristics on Image Segmentation

Image segmentation refers to the partitioning or dividing of a digital image into several smaller parts or segments in order to study a given image in a detailed manner. Image segmentation is required to study in detail some particular features (areas of interest) of a digital image. It forms an important and exigent part of image processing and requires an exhaustive and robust search technique for its implementation. Its applications vary from satellite imaging (for locating roads, forests etc. in a satellite image) to medical imaging (for locating tumors, for analyzing the anatomical structure etc.); from machine vision to fingerprint recognition and so on. In fact, segmentation can be applied to study in detail any real life problem where digital image can be developed. However, it is considered to be quite a demanding task because of the presence of multiple objects in an image and sometimes due to the intrinsic nature of an image.

Researchers have worked on gray image segmentation using evolutionary algorithms and results shows a comparative difference with traditional methods. Similarly, the color image segmentation is more complex task than grey image segmentation since it require huge amount of data to be processed. It reveals that evolutionary algorithms perform much better than traditional algorithms. An immense amount of literature dedicated to image segmentation is available. Pal et al. [86] applied GA successfully for grey level image enhancement using all the basic property of GA. They designed a fitness function having different parameters. These parameters represented different domains and were considered as chromosomes. The basic GA operators were then applied for enhancing the image. Shyu and Leou [87] proposed an approach for color image enhancement based on GA and formulated color image

enhancement as an optimization problem. They combined four nonlinear color image enhancement transforms viz. (1) AC power measure (2) Compactness measure (3) Brenner's Measure and (4) Information-noise change measure together to formulate an fitness function which was optimized with the help of GA.

Tao et al. [88] proposed a GA based three level thresholding method for image segmentation. They portioned the image in three basic parts as dark, grey and white and then implemented fuzzy region as Z-function, P-function and S-function respectively. GA is used to find an optimal solution for all the fuzzy parameters. Magoulas et al. [89] formulated the problem for on-line neural network based malignant detection in colonoscopy images. They developed an on-line strategy with the help of Differential evolution for approximation of the optimum solution. Roula et al. [90] states the problems of automatic segmentation of nuclei in histopathological images and for solving global optimization problem, an active contour based evolutionary approach was proposed. Results tabulated show the efficiency of the method. Zahara et al. [91] proposed a hybrid optimization technique called NM-PSO-Otsu method (Otsu's method with Nelder-Mead simplex search and PSO) and NM-PSO-curve method (Gaussian curve fitting by Nelder-Mead simplex search and PSO).

Shih and Wu [92] proposed a method based on GA for removing the rounding errors to achieve a hidden watermark from the image. Omran et al. [93], based on DE developed a clustering method that is then applied to unsupervised classification and segmentation of images. Feng et al. [94] applied PSO for 2-D maximum entropy method to optimize the fitness function developed with the help of 2-D histogram. PSO provides the threshold values as local average intensity of pixels. Rahnamayan et al. [95] introduced a new optimization based thresholding approach using Differential evolution. This method is compared with the Kittler algorithm of thresholding and results showed the new method to be more efficient. Dehmeshki et al. [96] proposed nodules detection method based on genetic GA template-matching (GATM). For enhancement purpose, a spherically oriented and convolution based filters were used as a preprocessor. Fitness function used a 3D geometric shape feature for each voxel (value on a grid in a 3-D space). They combined these features into a global nodule intensity distribution. Aslantas and Tunckanat [97] explain the usage of different DE algorithms aimed at segmenting the wounds on skin. This technique removes the disadvantages of k-means clustering. Jiang et al. [98] proposed a method for micro calcification of clusters in digital mammograms based on GA. In this method, a window of size 9×9 computes mean and standard deviation for each pixel. These computed values of pixels behave like chromosomes for initial population GA detect micro calcification clusters with an optimized way. This method outperforms in comparison to traditional methods. Aslantas and Tunckanat [97] proposed a color-based region segmentation method for skin lesions using DE algorithm. Their method tried to reduce the disadvantages of K-Means clustering algorithm. Fan and Lin [99] proposed (PSO+EM) algorithm for estimation of Gaussian's parameters. PSO employed for global search and the best particle was updated through expectation maximization (EM). PSO+EM were applied for multilevel image thresholding. Yin [100] applied PSO for minimum cross entropy thresholding (MCET) successfully. Li and Li [101] showed a successful application of PSO for fuzzy entropy image seg-

mentation. Maitra and Chatterjee [102], proposed an improved variant of PSO called HCOCLPSO (hybrid cooperative–comprehensive learning based PSO) for computing optimal multilevel thresholding for histogram-based image segmentation. This approach consists of cooperative and comprehensive learning.

Falco et al. [103] developed software based on DE algorithm which automatically registered multi view and multi temporal images. Hammouche et al. [104] proposed a wavelet transform method combined with GA. This method reduced the original length of histogram using wavelet transform. GA is used to select the number of thresholds and values of thresholds in this reduced histogram. Rahnamayan et al. [22] introduced micro opposition based DE (micro-DE). This method performed better than Kittler algorithm in sixteen test images. Basturk and Gunay [105] proposed an edge detector based on CNN (cellular neural network) optimized by DE. Hasan et al. [106] aims on pulling out chained codes of thinned binary image using DE and PSO. The result demonstrates better performance of the proposed PSO approach compared to the proposed DE approach. Rahnamayan et al. [22] proposed a thresholding algorithm based on micro Opposition-based Differential Evolution (ODE) for minimizing dissimilarity between input image and threshold image. Comparison with Kittler algorithm showed better performance of ODE. Coelho et al. [107], proposed a chaotic differential evolution schemes for image enhancement. The fitness function is maximized for enhancing the contrast and details of the image by adapting the parameters using a contrast enhancement technique. Das and Konar [108] proposed an evolutionary-fuzzy clustering algorithm for designing the homogeneous regions in an image. Fitness function is designed as a fuzzy clustering task in intensity space and applies an improved variant of DE to determine the clustering in the image and for refining the cluster centers. Results are compared with genetic fuzzy clustering technique and the classical fuzzy c-means showing the superiority of the proposed technique in terms of robustness, speed and accuracy. Maulik and Saha [69] proposed a fuzzy clustering technique based on a modified DE challenging task of clustering. This technique successfully makes a difference among all the remote areas for satellite imagery. Forouzanfar et al. [109] proposed a PSO/GA based optimization method for finding an optimal neighborhood attraction. All the simulation of results show an improvement in image segmentation. Zhang et al. [110] proposed an undecimated wavelet transform (UWT) with GA for typhoon cloud enhancement. Modifying the undecimated wavelet coefficients make it efficient to reduce the noise in typhoon cloud image. GA is used for modifying the UWT coefficient efficiently. Hashemi et al. [111] proposed a contrast enhancement technique based on GA where chromosome representation is simple and novel together with corresponding operators. Using this method natural look of the images is preserved in comparison to other contrast enhancement methods especially for high dynamic images. Korürek et al. [112] proposed a method for modeling near field effect of X-ray source. Parameters of this model are estimated with GA. This method of near field effect corrected all the pixels successfully and efficiently. Application of this method was done for X-ray images having poor quality.

Perez et al. [113] proposed a template based on PSO for iris localization and iris recognition which is very successful having a 97.4% efficiency and 40% reduction in

processing time. Papa et al. [114] proposed a method in which relaxation parameter (k) is obtained from PSO, which is used by POCS (Projections on to Convex Sets) to restore the image. Das and Sil [115], proposed a modified DE algorithm for pixel clustering in images. They used kernel function which partitions the data in many forms like non-separable and non-hyper-spherical for the original image. Optimal number of clusters are selected with a novel search-variable representation scheme from several possible choices resulting in an efficient technique. Similarly Cuevas et al. [116] proposed a thresholding technique based on DE. In this technique, DE is used for calculation of Gaussian functions parameters. This Gaussian function fills the 1-D histogram of the image. Approximation of the histogram with Gaussian function represents a class of pixels or threshold value. This technique has no prior information for thresholding and was shown to be computationally efficient. Azarbad et al. [117], introduced a blend of Hierarchical Evolutionary Algorithm (HEA) and multi-level thresholding algorithm for segmenting magnetic resonance images. On the basis of an automatic multi-level thresholding approach, HEA uses an unsupervised clustering technique. The results were evaluated and performance was validated. Aslantas and Kurban [118] apply DE for optimizing the block size before combining them into one as a fused image. Abuhaiba and Hassan [119] proposed a DE embedded with Discrete Fourier Transform (DFT) operations for image encryption. A 2-D encrypted key is generated by Discrete Fourier Transform (DFT) with original image. The seed used for Linear Feedback Shift Register (LFSR) is initialized with a secret key. Crossover and mutation operations of DE are used for index produced by LFSR. LFSR shuffles the image pixels. The deciphering is a reversible process of the same as explained above.

Kumar et al. [120] combined the Otsu method with differential evolution technique to select the optimum threshold value. The results were verified by testing the method on four different images. Mesejo et al. [121] proposed a method based on evolutionary algorithms on biomedical images for localizing the hippocampus in histological images. They used different methods like DE, Levenberg–Marquardt, GA, PSO, Simulated Annealing, and Scatter Search for computation and it was shown that DE outperforms the other algorithms. Tang et al. [122] proposed a genetic algorithm based fastening threshold selection in multilevel Minimum cross entropy thresholding (MCET). To reduce computational complexity recursive programming is used for objective function. These values are used as a chromosome representation for GA, which search several optimal multilevel threshold values. To solve the complexity of watermarking Wang et al. [31], proposed a multi-objective GA based approach. For embedding watermarks, it automatically optimizes system parameters variable length mechanism. This method also helps in removing some issues related to optimal watermarking parameters and avoid the difficulty in single-objective watermarking schemes. Mukhopadhyay and Maulik [123] proposed a genetic algorithm embedded with a fuzzy clustering which was used in magnetic images segmentation for detecting human brain diseases. Kwedlo [124] proposed a new clustering algorithm DE-KM (Differential Evolution with K-Means algorithm). DE provide solutions candidate by mutation and crossover. These solutions are fine-tuned by K-means algorithm. Chander et al. [125] proposed a variant of PSO for multilevel threshold-

ing for reducing the complexity and computational time. Zhang et al. [126] applied PSO in fuzzy clustering for optimizing initial clustering centers for FCM (Fuzzy Clustering Mean). Similarly, Wang et al. [127] proposed a multidimensional PSO for unsupervised planar segmentation.

Masra et al. [128] proposed a Histogram Equalization (HE) method based on Particle Swarm Optimization (PSO). Images are split in red green and blue (RGB) channels and PSO finds the best fitness value for each channel. Chen et al. [129] proposed a method of saliency-directed color image interpolation using particle swarm optimization (PSO) and Artificial Neural Network (ANN). PSO is used for determining the weights in interpolation filtering masks and ANN is used to determine the orientation of each image pattern (block). It was shown that this method performs better for color image interpolation. Pavan et al. [130] proposed an Automatic Clustering using DE (ACDE) for tissue segmentation. Shape of tissues is captured automatically with this algorithm, which helps in identifying certain diseases like breast cancer, lung tissues in ultrasonic images. Santamaría et al. [131] proposed DE for achieving parameters accurate alignments. They combined two different population-based optimization approaches and used it for image registration methods. Nakib et al. [132] considered Gaussian distribution's parameter selection as a nonlinear optimization problem and solved it with new variant of DE called low-discrepancy sequences and a local search (LDE). The proposed method outperforms in comparison to traditional methods for multilevel thresholding. Vahedi et al. [133] proposed GA based watermarking approach for color images for copyright purposes. Li et al. [134] on the basis genetic algorithm for water area extraction during flood events and two-dimensional entropy introduced a type of image processing. In the areas of water monitoring, this technique was found to be quite efficient and reliable. Wu et al. [135] developed a computer aided diagnosis system which classifies an ultrasound for breast tumor. To identify a tumor as benign or malignant a GA based system extracts all the morphologic features and auto-covariance texture features. All these features are near optimal parameters for support vector machine (SVM).

Lee et al. [136] proposed a modified PSO for extracting high-level image semantics depending upon the color, intensity and orientation. Tsai et al. [137] proposed Image enhancement filter named DPSM (Decision tree, Particle swarm optimization, and Support vector regression Median-type filter). A varying 2-level hybrid impulse noise detector notes the impulse noise. The decision tree will be optimized by particle swarm optimization (PSO) algorithm. For reconstruction of corrupted pixels a median-type filter with the support vector regression (MTSVR) is used. Dong et al. [138] proposed an embedded algorithm of PSO, GA and CHA (Chaotic Hybrid Algorithm) for circle detection in images of all types. Population initialization is based on opposition based learning (OBL). The basic idea uses velocity and position update from PSO and selection, crossover and mutation from GA. Sumer and Turker [139] proposed an adaptive fuzzy-GA for building detection integrated with fundamental image processing operators. Vellasques et al. [140] proposed a DPSO (Dynamic Particle Swarm Optimization) method for intelligent watermarking using evolutionary computing based on memory of Gaussian Mixture Models (GMMs) having an efficiency of 97.7%. A most advance methodology using PSO initialization in FCM is

proposed by Benaichouche et al. [141]. Gao et al. [142] proposed an advance variant of PSO named (IDPSO) intermediate disturbance searching strategy for enhancing the search ability of particles and increases their convergence rates for applying on image segmentation. Osuna-Enciso et al. [143] successfully applied PSO, ABC and DE for making the calculation of parameters used in 1-D histogram. Gaussian function approximates the 1-D histogram, for calculation of threshold point in grey level image segmentation. Akay [144] applied global optimization algorithms, PSO and ABC, for calculating the optimal multilevel thresholding. Kapur's entropy and between class variance were used as fitness functions for these algorithms. Kwok et al. [145] proposed a multi-objective PSO for correcting the color effect and for balancing all effects regarding the scene and environment. Hoseini and Shayesteh [146] proposed an evolution based contrast enhancement technique embedded on three techniques named GA, Ant Colony Optimization (ACO) and Simulated Annealing (SA). Input intensities to the output intensities were mapped through ACO, and then SA worked as local search methods for modifying the transfer functions. The processed ants worked as chromosome for GA. Galbally et al. [147] proposed a GA based probabilistic approach for reconstruction of iris images binary templates and measure its similarity with reconstructed and original iris image. Sarkar and Das [148] presented a 2D histogram based multilevel thresholding approach for improving separation between objects. The results were found to be superior than the results achieved from 1D histogram based technique for bi level thresholding. Sarkar and Das [148] solved the multilevel thresholding with 2D histogram. They proposed a 2D histogram based approach which utilizes maximum Tsallis entropy. Comparison was made among nature inspired algorithms like GA, PSO, ABC and simulated annealing where DE gave the best results. Ugolotti et al. [149] proposed a model which automatically detects objects in images and video sequences. For optimization, PSO and DE algorithms are used in this 3-D model. The proposed parametric model for object detection converts into a global continuous optimization. An artificial 3D model of a human body is matched against a human body in action for detecting its postures. Novo et al. [150] proposed an embedded method which combines SPEA2 (Strength Pareto Evolutionary Algorithm 2), a very well-known multi-objective algorithm with DE for performing complex 2-D and 3-D type segmentation.

Paul and Bitan [151] gave image compression technique built on histogram for multi-level thresholding. Various image quality techniques were used for image comparison. Specific application of the method was also proposed. Ali et al. [152, 153], proposed an advance variant of DE named synergetic DE (SDE) that utilized entropy and approximation of normalized histogram for finding the optimal thresholds. Results showed the better performance of SDE in comparison to other methods. To outdo the over-enhancement of Histogram Equalization, Shanmugavadi and Balasubramanian [154] proposed a PSO based histogram equalization method. Ali et al. [152, 153] proposed a method which partitions the original image into blocks. Then each block is converted into Discrete Cosine Transform domain. An approximation low-resolution image is formed through DCT, then SVD (Singular Value Decomposition) is applied. These singular values are modified for embedding the watermark. Lei et al. [155] proposed a secure watermarking method for medical images based on DE. It is a very robust and promising method for secure watermarking. Ochoa-Montiel et al. [156] gave the thresholding of biological images using multi objective optimization technique. In the paper, a combination of Shannon technique with Otsu thresholding has been used for inter and intra class. The results were tabulated and verified showing less computational efforts for this technique. Allaoui and Nasri [157] proposed a technique to solve thresholding problem, initialization and sensitivity to noise. The suggested method is on region growing and evolutionary approach.

Samanta et al. [158] proposed a novel authentication system for robust biomedical content, which embeds multiple hospitals' logos or multiple electronic patient records within the medical image. A hybrid approach for watermarking based on discrete cosine transformation, discrete wavelet transformation, and singular value decomposition teamed with optimization algorithms is proposed. The results prove the dominance of both the quantum-inspired evolutionary algorithm and quantum-inspired genetic algorithm over the genetic algorithm. Zhong et al. [159] discussed the core potentials of computational intelligence for optical remote sensing image processing. The computational intelligence in optical remote sensing image processing, feature representation and selection based on computational intelligence were reviewed. Kumar et al. [160] suggested a fitness function based on pixel-by-pixel values and optimized these values through evolutionary algorithms like differential evolution (DE), particle swarm optimization (PSO) and genetic algorithms (GA). Experimental results showed that DE performed better in comparison to PSO and GA on the basis of computational time and quality of segmented image. Ali and Ahn [161] introduced cuckoo search (CS) for finding the optimal scaling factors (SFs) in digital image watermarking to improve robustness and imperceptibility. Empirical analysis of the results demonstrated the efficiency of the proposed technique.

References

1. Holland, J.H.: Genetic algorithms. *Sci. Am.* **267**(1), 66–73 (1992)
2. Eberhart, R., Kennedy, J.: A new optimizer using particle swarm theory. In: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, vol. 1, pp. 39–43 (1995)
3. Storn, R., Price, K.: Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J. Global Optim.* **11**(4), 341–359 (1997)
4. Yang, X.S., Karanoglu, M., He, X.: Flower pollination algorithm: a novel approach for multi objective optimization. *Eng. Optim.* **46**(9), 1222–1237 (2014)
5. Yang, X.S.: Flower pollination algorithm for global optimization. In: Proceedings of the International conference on unconventional computing and natural computation (pp. 240–249). Springer, Berlin, Heidelberg, (2012)
6. Pahner, U., Hameyer, K.: Adaptive coupling of differential evolution and multiquadratics approximation for the tuning of the optimization process. *IEEE Trans. Magn.* **36**(4), 1047–1051 (2000)
7. Abbass, H.A., Sarker, R., Newton, C.: PDE: a Pareto-frontier differential evolution approach for multi-objective optimization problems. In: Proceedings of the 2001 Congress in Evolutionary Computation, vol. 2, pp. 971–978 (2001)
8. Zaharie, D.: Critical values for the control parameters of differential evolution algorithms. In: Proceedings of MENDEL, vol. 2, pp. 62–67 (2002)
9. Madavan, N.K., Bryan, A.B.: Multiobjective optimization using a Pareto differential evolution approach. In: Proceedings of the Evolutionary Computation on 2002. CEC'02. Proceedings of the 2002 Congress-vol. 02, pp. 1145–1150, IEEE Computer Society (2002)
10. Babu, B.V., Jehan M.M.L.: Differential evolution for multi-objective optimization. In: CEC'03. The 2003 Congress in Evolutionary Computation, vol. 4, pp. 2696–2703. IEEE (2003)
11. Xue, F., Sanderson, A.C., Graves, R.J.: Pareto-based multi-objective differential evolution. In: CEC'03, The 2003 Congress in Evolutionary Computation, vol. 2, pp. 862–869. IEEE (2003)
12. Fan, H.-Y., Lampinen, J.: A trigonometric mutation operation to differential evolution. *J. Glob. Optim.* **27**(1), 105–129 (2003)
13. Iorio, A.W., Li, X.: Solving rotated multi-objective optimization problems using differential evolution. AI 2004: Advances in Artificial Intelligence, pp. 861–872. Springer, Berlin Heidelberg (2004)
14. Thomsen, R.: Multimodal optimization using crowding-based differential evolution. In: CEC2004. Congress in Evolutionary Computation, vol. 2, pp. 1382–1389. IEEE (2004)
15. Qin, A.K., Suganthan, P.N.: Self-adaptive differential evolution algorithm for numerical optimization. In: 2005 IEEE Congress on Evolutionary Computation, vol. 2, pp. 1785–1791 (2005)
16. Robič, T., Filipič, B.: DEMO: Differential evolution for multiobjective optimization. In: Evolutionary Multi-criterion Optimization, pp. 520–533. Springer, Berlin Heidelberg (2005)
17. Krishnanand, K.N., Ghose, D.: Glowworm swarm based optimization algorithm for multimodal functions with collective robotics applications. *Multiagent and grid systems*, **2**(3), 209–222 (2006)
18. Chakraborty, U.K., Das, S., Konar, A.: Differential evolution with local neighbourhood. In: CEC 2006. IEEE Congress in Evolutionary Computation, pp. 2042–2049 (2006)
19. Kim, H.K., Jin, K.C., Kyong, Y.P., David, A.L.: Differential evolution strategy for constrained global optimization and application to practical engineering problems. *IEEE Trans. Magn.* **43**(4), 1565–1568 (2007)
20. Zhang, J., Chung, H.S.H., Lo, W.L.: Clustering-based adaptive crossover and mutation probabilities for genetic algorithms. *IEEE Trans. Evol. Comput.* **11**(3), 326–335 (2007)
21. Noman, N., Iba, H.: Accelerating differential evolution using an adaptive local search. *IEEE Trans. Evol. Comput.* **12**(1), 107–125 (2008)

22. Rahnamayan, S., Tizhoosh, H. R., Salama, M. M.: Opposition-based differential evolution. *IEEE Trans. Evol. Comput.* **12**(1), 64–79 (2008)
23. Yang, X.S.: Harmony search as a metaheuristic algorithm. *Music-Inspired Harmony Search Algorithm*, pp. 1–14. Springer, Berlin Heidelberg (2009)
24. Qin, A.K., Vicky, L.H., Suganthan, P.N.: Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Trans. Evol. Comput.* **13**(2), 398–417 (2009)
25. Das, S., Abraham, A., Chakraborty, U.K., Konar, A.: Differential evolution using a neighborhood based mutation operator. *IEEE Trans. Evol. Comput.* **13**(3), 526–553 (2009)
26. Jeyakumar, G., Velayutham, C.S.: A comparative performance analysis of differential evolution and dynamic differential evolution variants. In: *NaBIC 2009. World Congress on Nature & Biologically Inspired Computing*, pp. 463–468. IEEE (2009)
27. Zhang, J., Sanderson, A.C.: JADE: adaptive differential evolution with optional external archive. *IEEE Trans. Evol. Comput.* **13**(5), 945–958 (2009)
28. Gong, W., Cai, Z., Ling, C.X.: DE/BBO: a hybrid differential evolution with biogeography based optimization for global numerical optimization. *Soft. Comput.* **15**(4), 645–665 (2010)
29. Takahama, T., Setsuko, S.: Constrained optimization by the ε constrained differential evolution with an archive and gradient-based mutation. In: *IEEE Congress on Evolutionary Computation*, pp. 1–9. IEEE (2010)
30. Das, S., Suganthan, P.N.: Differential evolution: a survey of the state-of-the-art. *IEEE Trans. Evol. Comput.* **15**(1), 4–31 (2011)
31. Wang, J., Peng, H., Shi, P.: An optimal image watermarking approach based on a multiobjective genetic algorithm. *Inf. Sci.* **181**, 5501–5514 (2011)
32. Qu, B.Y., Suganthan, P.N., Liang, J.J.: Differential evolution with neighbourhood mutation for multimodal optimization. *IEEE Trans. Evol. Comput.* **16**(5), 601–614 (2012)
33. Elsayed, S.M., Sarker, R.A., Essam, D.L.: An improved self-adaptive differential evolution algorithm for optimization problems. *IEEE Trans. Ind. Inf.* **9**(1), pp. 89–99 (2013)
34. Reed H.M., Nichols J.M., Earls C.J. A modified differential evolution algorithm for damage identification in submerged shell structures. *Mech. Syst. Signal Process* **39**, 396–408 (2013)
35. Juárez-Abad, J.A., Linares-Flores, J., Guzmán-Ramírez, E., Sira-Ramírez, H.: Generalized proportional integral tracking controller for a single-phase multilevel cascade inverter: An FPGA implementation. *IEEE Transactions on Industrial Informatics*, **10**(1), 256–266 (2014)
36. Yu, W.J., Meie, S., Wei, N.C., Zhi, H.Z., Yue, J.G., Ying, L., Ou, L., Jun, Z.: Differential evolution with two-level parameter adaptation. *IEEE Trans. Cybern.* **44**(7), 1080–1099 (2014)
37. Cai, Y., Du, J.: Enhanced differential evolution with adaptive direction information. In: *2014 IEEE Congress on Evolutionary Computation (CEC)*, pp. 305–312 (2014)
38. Cai, Y., Wang, J., Chen, Y., Wang, T., Tian, H., Luo, W.: Adaptive direction information in differential evolution for numerical optimization. *Soft Computing*, **20**(2), 465–494 (2016)
39. Gong, W., Cai, Z., Liang, D.: Adaptive ranking mutation operator based differential evolution for constrained optimization. *IEEE Trans. Cybern.* **45**(4), 716–727 (2015)
40. Guo, S.M., Tsai, J.S.H., Yang, C.C., Hsu, P.H.: A self-optimization approach for L-SHADE incorporated with eigenvector-based crossover and successful-parent-selecting framework on CEC 2015 benchmark set. In: *Evolutionary Computation, IEEE Congress on* (pp. 1003–1010). IEEE (2015)
41. Alam, S., Tawseef, M., Khan, F., Fattah, A.A., Kabir, M.R.: Differential evolution with alternating strategies: a novel algorithm for numeric function optimization. In: *Communications on Applied Electronics (CAE)*, vol. 4, no. 2, pp. 12–16. Foundation of Computer Science FCS, New York, USA. ISSN: 2394–4714 (2016)
42. Guo, Z., Liu, G., Li, D., Wang, S.: Self-adaptive differential evolution with global neighbourhood search. *Soft Comput.*, 1–10 (2016)
43. Qiao, D., Grantham, K.H.P.: A modified differential evolution with heuristic algorithm for nonconvex optimization on sensor network localization. *IEEE Trans. Veh. Technol.* **65**(3), 1676–1689 (2016)
44. Sakr, W.S., El-Sehiemy, R.A., Azmy A.M.: Adaptive differential evolution algorithm for efficient reactive power management. *Appl. Soft Comput.* **1**(53), 336–351 (2017)

45. Zaheer, H., Pant, M., Kumar, S., Monakhov, O., Monakhova, E., Deep, K.: A new guiding force strategy for differential evolution. *Int. J. Sys. Assur. Eng. Manag.* **8**(4), 2170–2183 (2017)
46. Zhang S.X., Zheng S.Y., Zheng L.M.: An efficient multiple variants coordination framework for differential evolution. *IEEE Trans. Cybern.* **47**(9), 2780–2793 (2017)
47. Qian, S., Ye, Y., Liu, Y., Xu, G.: An improved binary differential evolution algorithm for optimizing PWM control laws of power inverters. *Optim. Eng.* **19**(2), 271–296 (2018)
48. Wu, G., Shen, X., Li, H., Chen, H., Lin, A., Suganthan, P.N.: Ensemble of differential evolution variants. *Inf. Sci.* **423**, 172–186 (2018)
49. Kaur, G., Singh, D., Kaur, M.: Robust and efficient ‘RGB’ based fractal image compression: flower pollination based optimization. *Proc. Int. J. Comput. Appl.* **78**(10), 11–15 (2013)
50. Kaur, G., Singh, D.: Pollination based optimization for color image segmentation. *Int. J. Comput. Eng. Technol.* **3**(2), (2012)
51. Abdel-Raouf, O., El-Henawy, I., Abdel-Baset, M.: A novel hybrid flower pollination algorithm with chaotic harmony search for solving sudoku puzzles. *Int. J. Mod. Educ. Comput. Sci.* **6**(3), p. 38 (2014)
52. Abdel-Raouf, O., Abdel-Baset, M.: A new hybrid flower pollination algorithm for solving constrained global optimization problems. *Int. J. Appl. Oper. Res.—An Open Access Journal* **4**(2), 1–13 (2014)
53. Wang, R., Zhou, Y.: Flower pollination algorithm with dimension by dimension improvement. *Math. Probl. Eng.*, 1–9 (2014)
54. Nguyen, T.T., Shieh, C.S., Horng, M.F., Dao, T.K., Ngo, T.G.: Parallelized Flower Pollination algorithm with a communication strategy. In: Knowledge and Systems Engineering (KSE), 2015 Seventh International Conference on IEEE, pp. 103–107 (2015)
55. Zhou, Y., Wang, R., Luo, Q.: Elite opposition-based flower pollination algorithm. *Neurocomputing* **188**, 294–310 (2015)
56. Paterlini, S., Krink, T.: High performance clustering with differential evolution. In: CEC 2004. Congress on Evolutionary Computation, vol. 2, pp. 2004–2011. IEEE (2004)
57. Zaharie, D.: Density based clustering with crowding differential evolution. In: SYNASC 2005, Seventh International Symposium In Symbolic and Numeric Algorithms for Scientific Computing, p. 8. IEEE (2005)
58. Lu, W.: Unsupervised anomaly detection framework for multiple-connection based network intrusions (Doctoral dissertation). (2005)
59. Martínez-Estudillo, A.C., Hervás-Martínez, C., Martínez-Estudillo, F.J., García-Pedrajas, N.: Hybridization of evolutionary algorithms and local search by means of a clustering method. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, **36**(3), 534–545 (2005)
60. Abraham, A., Das, S., Konar, A.: Document clustering using differential evolution. In: CEC 2006, IEEE Congress in Evolutionary Computation, pp. 1784–1791 (2006)
61. Alves, V.S., Campello, R.J., Hruschka, E.R.: Towards a fast evolutionary algorithm for clustering. In: Evolutionary Computation, IEEE Congress on (pp. 1776–1783). IEEE (2006)
62. Zhang, Z., Cheng, H., Zhang, S., Chen, W., Fang, Q.: Clustering aggregation based on genetic algorithm for documents clustering. In: Evolutionary computation, IEEE World Congress on Computational Intelligence. IEEE Congress on (pp. 3156–3161). IEEE (2008)
63. Zhang, X., Ding, W., Wang, J., Fan, Z., Deng, G.: Spatial clustering with obstacles constraints using PSO-DV and K-medoids. In: ISKE 2008. 3rd International Conference in Intelligent System and Knowledge Engineering, vol. 1, pp. 246–251. IEEE (2008)
64. Das, S., Abraham, A., Konar, A.: Automatic clustering using an improved differential evolution algorithm. *IEEE Trans. Syst. Man Cybern.-Part A: Syst. Hum.* **38**(1), 218–237 (2008)
65. Indrajit, S., Ujjwal, M., Nilan, J.: Differential fuzzy clustering for categorical data. In: ICM2CS 2009. Proceeding of International Conference in Methods and Models in Computer Science, pp. 1–6. IEEE (2009)
66. Zheng, Z., Gong, M., Ma, L., Jiao, J., Wu, Q.: Unsupervised evolutionary clustering algorithm for mixed type data. In: 2010 IEEE Congress in Evolutionary Computation (CEC), pp. 1–8. IEEE (2010)

67. Pawlak, Z.: Rough set theory and its applications to data analysis. *Cybern. Syst.* **29**(7), 661–688 (1998)
68. Maulik, U., Saha, I.: Automatic fuzzy clustering using modified differential evolution for image classification. *IEEE Trans. Geosci. Remote Sens.* **48**(9), 3503–3510 (2010)
69. Maulik, U., Saha, I.: Modified differential evolution based fuzzy clustering for pixel classification in remote sensing imagery. *Pattern Recogn.* **42**, 2135–2149 (2009)
70. Alguliev, R.M., Aliguliyev, R.M., Hajirahimova, M.S., Mehdiyev, C.A.: MCMR: Maximum coverage and minimum redundant text summarization model. *Expert systems with applications*, **38**(12), 14514–14522 (2011)
71. Pham, D.T., Suarez-Alvarez, M.M., Prostov, Y.I.: Random search with k-prototypes algorithm for clustering mixed datasets. *Proc. R. Soc. Lond. A: Math. Phys. Eng. Sci.* **467**(2132), 2387–2403 (2011)
72. Hatamlou, A.: Black hole: a new heuristic optimization approach for data clustering. *Inf. Sci.* **222**, 175–184 (2013)
73. Suarez-Alvarez, M.M., Pham, D.T., Prostov, M.Y., Prostov, Y.I.: Statistical approach to normalization of feature vectors and clustering of mixed datasets. *Proc. R. Soc. A (p. rspa20110704)* (2012)
74. Voges, K., Pope, N.: Generating compact rough cluster descriptions using an evolutionary algorithm. In: *Genetic and Evolutionary Computation Conference* (pp. 1332–1333). Springer, Berlin, Heidelberg (2004)
75. He, Y., Jian W., Liang-xi Q., Lin M., Yan-feng S., Wen-fei, W.: A HK clustering algorithm based on ensemble learning. In: *IET International Conference in Smart and Sustainable City 2013 (ICSSC 2013)*, pp. 300–305. IET (2013)
76. Saha, S., Bandyopadhyay, S.: A symmetry based multiobjective clustering technique for automatic evolution of clusters. *Pattern recognition*, **43**(3), 738–751 (2010)
77. Singh, V., Saha, S.: Modified differential evolution based 0/1 clustering for classification of data points: using modified new point symmetry based distance and dynamically controlled parameters. In: *2014 International Conference in Contemporary Computing and Informatics (IC3I)*, pp. 1182–1187. IEEE (2014)
78. Ameryan, M., Totonchi M.R.A., Mahdavi S.J.S.: Clustering based on Cuckoo optimization algorithm. In: *2014 Iranian Conference in Intelligent Systems (ICIS)*, pp. 1–6. IEEE (2014)
79. Thein, H.T.T., Khin, M.M.T.: Evaluation of differential evolution and K-means algorithms on medical diagnosis. In: *5th National Symposium in Information Technology: Towards New Smart World (NSITNSW)*, pp. 1–4. IEEE (2015)
80. Ozturk, C., Hancer, E., Karaboga, D.: Dynamic clustering with improved binary artificial bee colony algorithm. *Appl. Soft Comput.* **28**, 69–80 (2015)
81. Mukherjee, R., Dechoudhury, S., Das, S.: Modified differential evolution with locality induced genetic operators for dynamic optimization. *Eur. J. Oper. Res.* **253**(2), 337–355 (2016)
82. Wu, G., Mallipeddi, R., Suganthan, P.N., Wang, R., Chen, H.: Differential evolution with multi-population based ensemble of mutation strategies. *Information Sciences*, **329**, 329–345 (2016)
83. Cheng, M.Y., Tran, D.H., Hoang, N.D.: Fuzzy clustering chaotic-based differential evolution for resource leveling in construction projects. *J. Civil Eng. Manag.* **23**(1), 113–124 (2017)
84. Hancer, E., Karaboga, D.: A comprehensive survey of traditional, merge-split and evolutionary approaches proposed for determination of cluster number. *Swarm Evol. Comput.* **32**, 49–67 (2017)
85. Saha, S., Das, R.: Exploring differential evolution and particle swarm optimization to develop some symmetry-based automatic clustering techniques: application to gene clustering. *Neural Comput. Appl.* **30**(3), 735–757 (2018)
86. Pal, S.K., Bhandari, D., Kundu, M.K.: Genetic algorithms for optimal image enhancement. *Pattern Recogn. Lett.* **15**, 261–271 (1994)
87. Shyu, M., Leou, J.: A genetic algorithm approach to color image enhancement. *Pattern Recogn.* **31**(7), 871–880 (1998)

88. Tao, W., Tian, J., Liu, J.: Image segmentation by three-level thresholding based on maximum fuzzy entropy and genetic algorithm. *Pattern Recogn. Lett.* **24**, 3069–3078 (2003)
89. Magoulas, G.D., Plagianakos, V.P., Vrahatis, M.N.: Neural network-based colonoscopic diagnosis using on-line learning and differential evolution. *Appl. Soft Comput.* **4**, 369–379 (2004)
90. Roula, M.A., Ahmed, B., Fatih, K.: An evolutionary snake algorithm for the segmentation of nuclei in histopathological images. In: ICIP'04. 2004 International Conference in Image Processing, vol. 1, pp. 127–130. IEEE (2004)
91. Zahara, E., Fan, S.S., Tsai, D.: Optimal multi-thresholding using a hybrid optimization approach. *Pattern Recogn. Lett.* **26**, 1082–1095 (2005)
92. Shih, F.Y., Wu, T.: Enhancement of image watermark retrieval based on genetic algorithms. *J. Vis. Commun. Image Represent.* **16**, 115–133 (2005)
93. Omran, M.G.H., Andries, P.E., Ayed, S.: Differential evolution methods for unsupervised image classification. In: 2005 IEEE Congress on Evolutionary Computation, vol. 2, pp. 966–973 (2005)
94. Feng, D., Wenkang, S., Liangzhou, C., Yong, D., Zhenfu, Z.: Infrared image segmentation with 2-D maximum entropy method based on particle swarm optimization (PSO). *Pattern Recogn. Lett.* **26**, 597–603 (2005)
95. Rahnamayan, S., Hamid R.T., Magdy, M.A.S.: Image thresholding using differential evolution. In: IPCV, pp. 244–249 (2006)
96. Dehmehki, J., Ye, X., Lin, X.Y., Valdiveiso, M., Amin, H.: Automated detection of lung nodules in CT images using shape-based genetic algorithm. *Comput. Med. Imaging Graph.* **31**, 408–417 (2007)
97. Aslantas, V., Tunçkanat, M.: Differential evolution algorithm for segmentation of wound images. In: WISP 2007. IEEE International Symposium on Intelligent Signal Processing, pp. 1–5. IEEE (2007)
98. Jiang, J., Yao, B., Wason, A.M.: A genetic algorithm design for micro calcification detection and classification in digital mammograms. *Comput. Med. Imaging Graph.* **31**, 49–61 (2007)
99. Fan, S.S., Lin, Y.: A multi-level thresholding approach using a hybrid optimal estimation algorithm. *Pattern Recogn. Lett.* **28**, 662–669 (2007)
100. Yin, P.: Multilevel minimum cross entropy threshold selection based on particle swarm optimization. *Appl. Math. Comput.* **184**, 503–513 (2007)
101. Li, L., Li, D.: Fuzzy entropy image segmentation based on particle swarm optimization. *Prog. Nat. Sci.* **18**, 1167–1171 (2008)
102. Maitra, M., Chatterjee, A.: A hybrid cooperative–comprehensive learning based PSO algorithm for image segmentation using multilevel thresholding. *Expert Syst. Appl.* **34**, 1341–1350 (2008)
103. Falco, I.D., Cioppa, A.D., Maisto, D., Tarantino, E.: Differential evolution as a viable tool for satellite image registration. *Appl. Soft Comput.* **8**, 1453–1462 (2008)
104. Hammouche, K., Diaf, M., Siarry, P.: A multilevel automatic thresholding method based on a genetic algorithm for a fast image segmentation. *Comput. Vis. Image Underst.* **109**, 163–175 (2008)
105. Basturk, A., Gunay, E.: Efficient edge detection in digital images using a cellular neural network optimized by differential evolution algorithm. *Expert Syst. Appl.* **36**, 2645–2650 (2009)
106. Hasan, H., Haron, H., Hashim, S.Z.: Freeman chain code extraction using differential evolution (DE) and particle swarm optimization (PSO). In: SOCPAR'09. International Conference in Soft Computing and Pattern Recognition, pp. 77–81. IEEE (2009)
107. Coelho, L.S., Sauer, J.G., Rudek, M.: Differential evolution optimization combined with chaotic sequences for image contrast enhancement. *Chaos, Solitons Fractals* **42**, 522–529 (2009)
108. Das, S., Konar, A.: Automatic image pixel clustering with an improved differential evolution. *Appl. Soft Comput.* **9**, 226–236 (2009)
109. Forouzanfar, M., Forghani, N., Teshnehlab, M.: Parameter optimization of improved fuzzy c-means clustering algorithm for brain MR image segmentation. *Eng. Appl. Artif. Intell.* **23**, 160–168 (2010)

110. Zhang, C., Wang, X., Duanmu, C.: Adaptive typhoon cloud image enhancement using genetic algorithm and non-linear gain operation in undecimated wavelet domain. *Eng. Appl. Artif. Intell.* **23**, 61–73 (2010)
111. Hashemi, S., Kiani, S., Noroozi, N., Moghaddam, M.E.: An image contrast enhancement method based on genetic algorithm. *Pattern Recogn. Lett.* **31**, 1816–1824 (2010)
112. Korürek, M., Yüksel, A., Iscan, Z., Dokur, Z., Ölmez, T.: Retrospective correction of near field effect of X-ray source in radiographic images by using genetic algorithms. *Expert Syst. Appl.* **37**, 1946–1954 (2010)
113. Perez, A.C., Aravena, C.M., Vallejos, J.I., Estevez, P.A., Held, C.M.: Face and iris localization using templates designed by particle swarm optimization. *Pattern Recogn. Lett.* **31**, 857–868 (2010)
114. Papa, J.P., Fonseca, L.M.G., de Carvalho, L.A.S.: Projections on to convex sets through particle swarm optimization and its application for remote sensing image restoration. *Pattern Recogn. Lett.* **31**, 1876–1886 (2010)
115. Das, S., Sil, S.: Kernel-induced fuzzy clustering of image pixels with an improved differential evolution algorithm. *Inf. Sci.* **180**, 1237–1256 (2010)
116. Cuevas, E., Zaldivar, D., Pérez-Cisneros, M.: A novel multi-threshold segmentation approach based on differential evolution optimization. *Expert Syst. Appl.* **37**, 5265–5271 (2010)
117. Azarbad, M., Ebrahimzadeh, A., Babajani-Feremi, A.: Brain tissue segmentation using an unsupervised clustering technique based on PSO algorithm. In: 2010 17th Iranian Conference in Biomedical Engineering (ICBME), pp. 1–6. IEEE (2010)
118. Aslantas, V., Kurban, R.: Fusion of multifocus images using differential evolution algorithm. *Expert Syst. Appl.* **37**, 8861–8870 (2010)
119. Abuhaiba, I.S.I., Hassan, M.A.S.: Image encryption using differential evolution approach in frequency domain. *Signal Image Process.* **2**, 51–69 (2011)
120. Kumar, S., Pant M., Ray A.K.: Differential evolution embedded Otsu's method for optimized image thresholding. In: 2011 World Congress on Information and Communication Technologies (WICT), pp. 325–329. IEEE (2011)
121. Mesejo, P., Ugolotti, R., Di Cunto, F., Giacobini, M., Cagnoni, S.: Automatic hippocampus localization in histological images using Differential Evolution-based deformable models. *Pattern Recogn. Lett.*, **34**(3), 299–307 (2013)
122. Tang, K., Yuan, X., Sun, T., Yang, J., Gao, S.: An improved scheme for minimum cross entropy threshold selection based on genetic algorithm. *Knowl.-Based Syst.* **24**, 1131–1138 (2011)
123. Mukhopadhyay, A., Maulik, U.: A multi-objective approach to MR brain image segmentation. *Appl. Soft Comput.* **11**, 872–880 (2011)
124. Kwedlo, W.: A clustering method combining differential evolution with the K-means algorithm. *Pattern Recogn. Lett.* **32**, 1613–1621 (2011)
125. Chander, A., Chatterjee, A., Siarry, P.: A new social and momentum component adaptive PSO algorithm for image segmentation. *Expert Syst. Appl.* **38**, 4998–5004 (2011)
126. Zhang, Y., Huang, D., Ji, M., Xie, F.: Image segmentation using PSO and PCM with Mahalanobis distance. *Expert Syst. Appl.* **38**, 9036–9040 (2011)
127. Wang, L., Cao, J., Han, C.: Multidimensional particle swarm optimization-based unsupervised planar segmentation algorithm of unorganized point clouds. *Pattern Recogn.* **45**, 4034–4043 (2012)
128. Masra, S.M.W., Pang, P.K., Muhammad, M.S., Kipli, K.: Application of particle swarm optimization in histogram equalization for image enhancement. In: IEEE Colloquium on Humanities, Science & Engineering Research, pp. 294–299 (2012)
129. Chen, H., Leou, J.: Saliency-directed color image interpolation using artificial neural network and particle swarm optimization. *J. Vis. Communun. Image Represent.* **23**, 343–358 (2012)
130. Pavani, K.K., Srinivas, V.S., SriKrishna, A., Reddy, B.E.: Automatic tissue segmentation in medical images using differential evolution. *J. Appl. Sci.* **12**(6), 587–592 (2012)
131. Santamaría, J., Damas, S., García-Torres, J.M., Cordón, O.: Self-adaptive evolutionary image registration using differential evolution and artificial immune systems. *Pattern Recogn. Lett.* **33**, 2065–2070 (2012)

132. Nakib, A., Daachi, B., Siarry, P.: Hybrid Differential evolution using low discrepancy sequences for image segmentation. In: IEEE 26th International Parallel and Distributed Processing Symposium Workshops & Ph.D. Forum (IPDPSW), pp. 634–640 (2012)
133. Vahedi, E., Zoroofi, R.A., Shiva, M.: Toward a new wavelet-based watermarking approach for color images using bio-inspired optimization principles. *Digit. Signal Process.* **22**, 153–162 (2012)
134. Li, Z., Xiuwan, C., Peng, L., Yuan, T.: Water area segmentation of the Yangcheng Lake with SAR data based on improved 2D maximum entropy and genetic algorithm. In: 2012 Second International Workshop in Earth Observation and Remote Sensing Applications (EORSA), pp. 263–267. IEEE (2012)
135. Wu, W., Lina, S., Moon, W.K.: Combining support vector machine with genetic algorithm to classify ultrasound breast tumor images. *Comput. Med. Imaging Graph.* **36**, 627–633 (2012)
136. Lee, C., Leou, J., Hsiao, H.: Saliency-directed color image segmentation using modified particle swarm optimization. *Sig. Process.* **92**, 1–18 (2012)
137. Tsai, H., Chang, B., Lin, X.: Using decision tree, particle swarm optimization, and support vector regression to design a median-type filter with a 2-level impulse detector for image enhancement. *Inf. Sci.* **195**, 103–123 (2012)
138. Dong, N., Wu, C., Ip, W., Chen, Z., Chan, C., Yung, K.: An opposition-based chaotic GA/PSO hybrid algorithm and its application in circle detection. *Comput. Math. Appl.* **64**, 1886–1902 (2012)
139. Sumer, E., Turker, M.: An adaptive fuzzy-genetic algorithm approach for building detection using high-resolution satellite images. *Comput. Environ. Urban Syst.* **39**, 48–62 (2013)
140. Vellasques, E., Sabourin, R., Granger, E.: Fast intelligent watermarking of heterogeneous image streams through mixture modeling of PSO populations. *Appl. Soft Comput.* **13**, 3130–3148 (2013)
141. Benaiachouche, A.N., Oulhadj, H., Siarry, P.: Improved spatial fuzzy c-means clustering for image segmentation using PSO initialization, Mahalanobis distance and post-segmentation correction. *Digital Signal Processing*, **23**(5), 1390–1400 (2013)
142. Gao, H., Kwong, S., Yang, J., Cao, J.: Particle swarm optimization based on intermediate disturbance strategy algorithm and its application in multi-threshold image segmentation. *Inf. Sci.* **250**, 82–112 (2013)
143. Osuna-Enciso, V., Cuevas, E., Sossa, H.: A comparison of nature inspired algorithms for multi-threshold image segmentation. *Expert Syst. Appl.* **40**, 1213–1219 (2013)
144. Akay, B.: A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding. *Appl. Soft Comput.* **13**, 3066–3091 (2013)
145. Kwok, N.M., Shi, H.Y., Ha, Q.P., Fang, G., Chen, S.Y., Jia, X.: Simultaneous image color correction and enhancement using particle swarm optimization. *Eng. Appl. Artif. Intell.* **26**, 2356–2371 (2013)
146. Hoseini, P., Shayesteh, M.G.: Efficient contrast enhancement of images using hybrid ant colony optimization, genetic algorithm, and simulated annealing. *Digit. Signal Process.* **23**, 879–893 (2013)
147. Galbally, J., Ross, A., Gomez-Barrero, M., Fierrez, J., Ortega-Garcia, Javier: Iris image reconstruction from binary templates: an efficient probabilistic approach based on genetic algorithms. *Comput. Vis. Image Underst.* **117**, 1512–1525 (2013)
148. Sarkar, S., Das, S.: Multilevel image thresholding based on 2D histogram and maximum Tsallis entropy—a differential evolution approach. *IEEE Trans. Image Process.* **22**(12), 4788–4797 (2013)
149. Ugolotti, R., Nashed, Y.S.G., Mesejo, P., Ivezkovi, S., Mussia, L., Cagnoni, S.: Particle swarm optimization and differential evolution for model-based object detection. *Appl. Soft Comput.* **13**, 3092–3105 (2013)
150. Novo, J., Santos, J., Penedo, M.G.: Multi-objective differential evolution in the optimization of topological active models. *Appl. Soft Comput.* **13**, 3167–3177 (2013)
151. Paul, S., Bitan, B.: A novel approach for image compression based on multi-level image thresholding using Shannon Entropy and Differential Evolution. In: Students' Technology Symposium (TechSym), pp. 56–61. IEEE (2014)

152. Ali, M., Ahn, C.W., Pant, M.: A robust image watermarking technique using SVD and differential evolution in DCT domain. *Optik* **125**, 428–434 (2014)
153. Ali, M., Ahn, C.W., Pant, M.: Multilevel image thresholding by synergetic differential evolution. *Appl. Soft Comput.* **17**, 1–11 (2014)
154. Shammugavadi, P., Balasubramanian, K.: Particle swarm optimized multiobjective histogram equalization for image enhancement. *Opt. Laser Technol.* **57**, 243–251 (2014)
155. Lei, B., Tan, E., Chen, S., Ni, D., Wang, T., Lei, H.: Reversible watermarking scheme for medical image based on differential evolution. *Expert Syst. Appl.* **41**, 3178–3188 (2014)
156. Ochoa-Montiel, R., Sánchez-López, C., González-Bernal, J.A.: Thresholding of biological images by using evolutionary algorithms. In: 2015 Latin America Congress on Computational Intelligence (LA-CCI), pp. 1–6 (2015)
157. Allaoui, A.E., Nasri, M.B.: Threshold optimization by genetic algorithm for segmentation of medical images by region growing. *International Journal of Emerging Trends and Technology in Computer Science (IJETTCS)*, **1**(2), 161–166 (2012)
158. Samanta, S.O., Choudhury, A.L., Dey, N., Ashour, A.S., Balas V.E.: Quantum-inspired evolutionary algorithm for scaling factor optimization during manifold medical information embedding. *Quant. Inspired Comput. Intell.*, 285–326 (2017)
159. Zhong Y., Ma, A., Soon Ong, Y., Zhu, Z., Zhang, L.: Computational intelligence in optical remote sensing image processing. *Appl. Soft Comput.* (2017)
160. Kumar, S., Pant, M., Kumar, M., Dutt, A.: Colour image segmentation with histogram and homogeneity histogram difference using evolutionary algorithms. *Int. J. Mach. Learn. Cybern.* **9**(1), 163–183 (2018)
161. Ali, M., Ahn, C.W.: An optimal image watermarking approach through cuckoo search algorithm in wavelet domain. *Int. J. Syst. Assur. Eng. Manag.* **9**(3), 602–611 (2018)

Chapter 3

Revised Mutation Strategy for Differential Evolution Algorithm



This chapter introduces a newly developed variant of Differential Evolution named ReDE—Revised Differential Evolution. This variant is created by changing the mutation strategy of traditional DE algorithm. ReDE uses two control parameters and two types of population in its mutation strategy. This strategy uses two weighted difference unlike traditional DE strategies. The new mutation strategy in ReDE gives progressive results in comparison to the traditional mutation strategies of DE.

3.1 Introduction

Many of the real time tasks in the scientific world can be consider as a constrained optimization problem. For solving these optimization problems, various evolutionary algorithms like Differential Evolution, Flower Pollination Algorithm etc. are in use. In today's scientific world, evolutionary algorithms are a major area for research, providing immense opportunity to researchers. Several researches are performed in this area to create variants of the evolutionary algorithms and to further improve their performance. Evolutionary algorithms are those algorithms that use the techniques prompted by biological evolution such as reproduction, selection, mutation and recombination.

In 1995, Storn and Price proposed the Differential Evolution (DE), which is a powerful and efficient algorithm to solve optimization problems. Like various other evolutionary algorithms, DE is also a population based stochastic method. DE is among the robust evolutionary algorithm for solving real valued test functions. DE

has various significant features like its ease to implement, better performance, low space complexity and fewer control parameters. The efficiency and performance of DE greatly depends on the trial vector generation and the control parameters used. Differential Evolution (DE) is a dominant technique for optimization and is used to solve various real time problems.

Various researches are in progress to improve the performance of DE by changing the control parameters and the trial vector generation strategy. In this section, a new mutation strategy for DE is introduced and is named as ReDE—Revised Differential Evolution. In this strategy, two types of control parameters are used—a variable parameter value between (0,2) and constant parameter value in the range (0,1) and two types of population—the population from previous generation and new population from current generation. This strategy has remarkably better results in comparison to the classical DE approach.

3.2 Revised Mutation Strategy

A new strategy for mutation called Revised DE strategy (ReDE) is proposed in this work. ReDE uses two control parameters and two types of population in its mutation strategy unlike traditional DE technique which uses one control parameter and one type of population in its mutation strategy. As the mutation strategy involves the best solution vector $X_{best,G}$, the algorithm coincides faster as compared to the traditional strategies having random vectors only. This strategy of mutation uses two sets of population: the old population represented as *popold* and the new population from which the variable $X_{r1,G}$ is chosen at random. This strategy uses two control parameters. The parameter F known as mutation factor takes a constant value between (0,2) while the new parameter N takes a varying value which lies between (0,1). The proposed strategy is given as:

$$X' = X_{best,G} + N \cdot (X_{best,G} - popold) - F \cdot (X_{best,G} - X_{best,G}) \quad (3.1)$$

The new mutation strategy is combined with the crossover technique of DE and the new algorithm formed is named as ReDE. By changing the mutation strategy, the result obtained from ReDE differs drastically from the results obtained from traditional DE technique.

3.3 Experimental Setting

Implementation of the revised algorithm is developed on i7 core processor, 64 bit operating system with 12 GB RAM using MATLABr2008b and comparative results are obtained with five different traditional mutation strategies of DE algorithm. In this

computation, fifteen different standard functions are taken and results are computed. Few of the results obtained are shown in Tables 3.1, 3.2, 3.3, 3.4 and 3.5. The Value to Reach (VTR) is the global minimum or maximum of the function, which when reached would stop optimization. Several values of VTR are taken for different dimensions during computation. By fixing the dimensions as 25, 50, 75 and 100, various results are tabulated for comparison with the existing strategies.

A comparative analysis is performed to study each of these techniques. The dimensions and value-to-reach (VTR) are set to calculate the best value, number of function evaluation (NFE) and the CPU time of different function strategies. By introducing new control parameters and two different population sets, the results obtained changes drastically in comparison to the other techniques. As a result, the proposed hybrid algorithm gives the best value for most of the standard functions. By changing the VTR for various dimensions, the proposed algorithm obtained the best value for most of the standard functions. In some functions, the results are good for both classical DE and the proposed algorithm. NFE for most of the standard functions are also best for the proposed algorithm. CPU time is also best for almost all standard functions using the proposed algorithm. The results shown in bold depict the best results obtained for different functions. From the results tabulated, strategy ReDE gives good results for most of the functions used. As ReDE coincides faster with the introduction of best value, the NFE obtained for ReDE is improved in comparison to the other mutation strategies.

Table 3.1 Comparative results for Best Value after 25 runs for $VTR = 1.0e-015$ for different DE strategies with ReDE

Function	DE					
	DE/best/1	DE/rand/1	DE/best-to-rand/1	De/best/2	DE/rand/2	ReDE
Sphere	9.73e-16	6.90e-16	7.53e-16	9.66e-16	7.17e+00	6.04e-16
Beale	3.27e-16	2.32e-16	3.71e-16	7.59e-16	7.73e-16	5.95e-16
Booth	3.50e-16	2.05e-16	6.07e-16	7.08e-16	8.35e-16	1.36e-16
Schwefel	-1.80e+03	-2.25e+03	-7.84e+01	-1.38e+003	-1.66e+003	-2.10e+03
Michlewicz	-7.6399e+00	-7.21e+00	-7.39e+00	-6.959e+00	-6.847e+00	-6.6e+00
Schaffer N.2	6.60e-16	8.88e-16	4.43e-16	6.55e-16	8.87e-16	2.22e-16
Schaffer N.2	3.05e-15	2.90e-01	2.92e-001	2.93e-01	2.89e-01	2.82e-01
HimmelBlau	1.60e-16	8.05e-16	3.83e-16	9.12e-16	1.46e-16	3.35e-16
Bird	-1.04e+02	-1.07e+02	-1.05e+02	-1.07e+02	-1.03e+02	-1.03e+02
Extended Cube	3.31e-15	4.98e-05	6.10e-08	1.93e-05	2.68e+00	8.60e-16
Ackley	7.19e-15	6.46e-12	7.99e-15	3.63e-13	3.09e+00	1.50e-14
Gold	3.00e+00	3.00e+00	3.00e+00	3.00e+00	3.00e+00	3.00e+00
Griewank	9.99e-16	9.99e-16	1.60e-13	6.56e-13	1.07e+00	2.40e-02
Rastrigin	1.79e+01	1.23e+02	7.47e+01	1.28e+02	1.52e+02	2.98e+01
Rosenbrock	9.60e-16	1.07e-08	7.88e-16	3.90e-09	1.07e+05	1.50e+01

Table 3.2 Comparative results for NFE after 25 runs for $VTR = 1.0e-015$ for different DE strategies with ReDE

Function	DE					
	DE/best/1	DE/rand/1	DE/best-to-rand/1	De/best/2	DE/rand/2	ReDE
Sphere	288,000	3,705,000	313,000	3,260,000	5,000,000	155,000
Beale	48,000	94,000	67,000	85,000	127,000	50,000
Booth	500,000	90,000	71,000	77,000	118,000	48,000
Schwefel	7000	12,000	13,000	4000	6000	2000
Michlewicz	1000	1000	1000	1000	1000	1000
Schaffer N.2	68,000	148,000	119,000	139,000	224,000	1000
Schaffer N.2	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000
HimmelBlau	45,000	95,000	67,000	77,000	199,000	55,000
Bird	1000	1000	1000	1000	1000	1000
Extended Cube	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000
Ackley	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000
Gold	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000
Griewank	2,880,000	4,579,000	500,000	500,000	500,000	5,000,000
Rastrigin	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000
Rosenbrock	60,900	5,000,000	77,000	5,000,000	5,000,000	5,000,000

Table 3.3 Comparative results of CPU time for $VTR = 1.0e-015$ for 25 runs for different strategies of DE with ReDE

Function	DE					
	DE/best/1	DE/rand/1	DE/best-to-rand/1	De/best/2	DE/rand/2	ReDE
Sphere	37.67	139.169	16.318	122.399	225.799	16.02
Beale	48.163	9.5786	8.263	8.533	8.81	32.65
Booth	11.6	8.25	6.817	8.494	8.5762	16.78
Schwefel	17.712	5.337	4.3044	4.956	4.5789	13.05
Michlewicz	2.73	2.217	2.817	2.064	2.074	6.4
Schaffer N.2	36.311	23.22	17.3	18.86	21.19	16.7
Schaffer N.2	190.07	227.9	261.1	238.23	245.6	360.1
HimmelBlau	14.8	16.94	18.4	12.3	15.95	43.2
Bird	12.39	11.8	8.25	8.71	8.07	16.64
Extended Cube	186.78	372.56	345.3	352.3	335.02	379.3
Ackley	314.6	310.42	325.56	312.2	300.4	362.3
Gold	257.45	425.8	331.62	312.52	319.5	314.5
Griewank	321.7	305.3	346.58	342.76	341.01	294.3
Rastrigin	254.69	316.8	265.4	310.1	323.52	300.6
Rosenbrock	52.01	36.23	52.79	324.06	34.53	39.3

Table 3.4 Comparative results for Best Value after 25 runs for $VTR = 1.0e-014$ for different strategies of DE with ReDE

Function	DE					
	DE/best/1	DE/rand/1	DE/best-to-rand/1	De/best/2	DE/rand/2	ReDE
Sphere	9.34e-15	9.35e-15	9.54e-15	9.94e-15	6.92e+00	8.80e-15
Beale	4.26e-15	7.72e-15	1.13e-15	1.36e-17	7.50e-15	1.19e-15
Booth	1.81e-15	7.55e-16	1.95e-15	2.75e-15	6.47e-15	2.32e-16
Schwefel	-2.22e+02	-4.80e+02	-1.67e+03	-4.47e+03	-1.50e+03	-6.80e+02
Michlewicz	-7.69e+00	-7.64e+00	-6.87e+00	-7.35e+00	-6.98e+00	-6.6e+00
Schaffer N.2	1.33e-15	1.33e-15	6.66e-16	5.30e-15	1.33e-15	4.40e-16
Schaffer N.2	2.92e-01	2.92e-01	2.92e-01	2.92e-01	2.92e-01	2.92e-01
HimmelBlau	4.83e-15	4.42e-15	1.90e-15	3.95e-15	5.14e-15	4.14e-15
Bird	-9.30e+01	-1.04e+02	-1.07e+02	-1.03e+02	-1.04e+02	-1.03e+02
Extended Cube	5.70e-08	5.21e-05	7.10e-08	1.73e-05	2.92e+09	5.47e-08
Ackley	7.99e-15	5.02e-15	7.99e-15	3.59e-13	3.21e+00	1.02e+00
Gold	3.00e+00	3.00e+00	3.00e+00	3.00e+00	3.00e+00	3.00e+00
Griewank	1.48e-02	9.21e-15	7.88e-15	5.07e-09	1.06e+00	6.66e-16
Rastrigin	3.61e+01	1.18e+02	8.17e+01	1.73e+02	1.67e+02	2.88e+00
Rosenbrock	3.98e+00	1.40e-08	6.90e-15	1.56e-11	7.15e-04	1.50e+00

Table 3.5 Comparative results of CPU time after 75 runs for $VTR = 1.e-016$ for different strategies of DE with ReDE

Function	DE					
	DE/best/1	DE/rand/1	DE/best-to-rand/1	De/best/2	DE/rand/2	ReDE
Sphere	11.92	70.7	41.13	72.9	100.83	13.76
Beale	10.4	7.9	7.5	6.3	6.9	5.5
Booth	6.91	8.83	6.8	9.2	7.6	5.78
Schwefel	7.17	9.3	5.7	5.13	6.5	5.9
Michlewicz	6.15	4.4	4.8	7.5	8.2	3.6
Schaffer N.2	7.03	7.36	8.6	8.5	7.2	7.6
Schaffer N.2	132.9	132.3	137.4	140.4	138.2	130.5
HimmelBlau	8.46	10.95	8.62	9.95	12.2	5.43
Bird	5.8	4.5	8.7	6.5	7.2	4.3
Extended Cube	171.78	161.24	169.23	174.9	166.9	163.2
Ackley	123.34	121.3	131.78	153.4	148.2	156.3
Gold	117.3	121.3	119.3	121.6	117.8	115.3
Griewank	129.1	92.1	123.4	125.6	122.3	9.5
Rastrigin	111.2	119.4	112.3	108.6	111.9	110.8
Rosenbrock	130.8	120.9	123.2	116.7	119.8	113.2

3.4 Graphical Results

Graphical representation of the above-tabulated values is shown in Figs. 3.1 and 3.2. The graphs show performance curve of six different function strategies. The x-axis represents the number of function evaluation and y-axis represents the objective function. The graph is depicted for the various values at each iteration for fixed VTR value of e-015 and dimension size of 25.

A comparative study based on the above graphs show that the revised mutation strategy provide better results in comparison to the existing mutation strategy for various functions. The graphical representation for Schwefel and Michelawicz function show that ReDE gives different results from the traditional mutation strategy.

Fig. 3.1 Graphical representation for Schwefel function

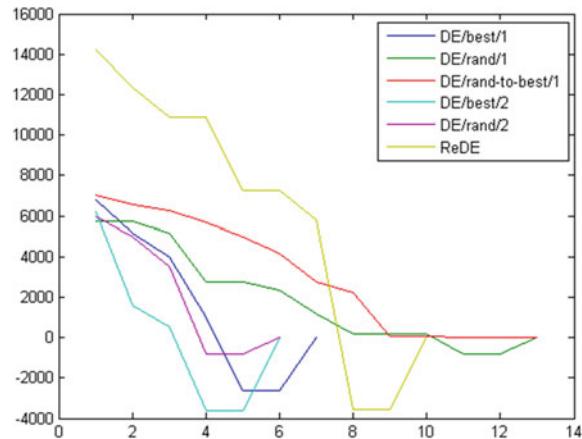
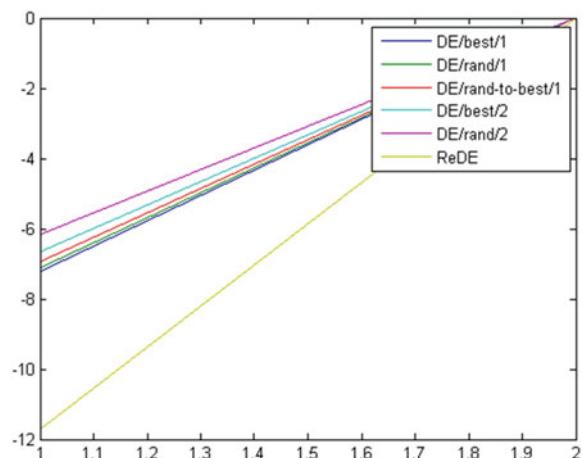


Fig. 3.2 Graphical representation for Michelawicz function



Since this strategy ReDE uses a combination of two sets of population with two different control parameters, the results are refined and the efficiency of the strategy is increased.

3.5 Statistical Analysis

Based on the values from Table 3.1, Friedman test is applied and the results obtained are tabulated. Table 3.6 represents the values obtained from the test and Table 3.7 depicts the rank of the various mutation strategies used based on best value, NFE and CPU time.

The above tables show that the new mutation strategy ReDE has significant performance in comparison to the existing mutation strategies. The rank obtained based on NFE is the best for ReDE. The rank obtained based on best value is comparatively better for ReDE. These rankings obtained based on Freidman's test justify the efficiency of ReDE strategy. Based on the ranks obtained, a graphical representation of the results are shown in Figs. 3.3, 3.4 and 3.5. The x-axis of the graph represents the various strategies used and the y-axis shows the ranks obtained.

3.6 Summary

In the proposed work, the revised strategy is compared with the existing mutation strategy to find improvements in efficiency. During the comparative study, the pro-

Table 3.6 Test statistics using Friedman's test

N	25
Chi sq	22.68
Df	5
Asymptotic significance	0.004

Table 3.7 Ranks of the different strategies

Strategies	Mean rank on best value	Mean rank on NFE
DE/best/1	2.7	3.0
De/rand/1	3.2	4.2
DE/best-to-rand/1	2.7	3.2
De/best/2	4.3	3.4
DE/rand/2	5.1	4.2
ReDE	3	2.86

Fig. 3.3 Bonferroni Dunn bar chart for rank on best value

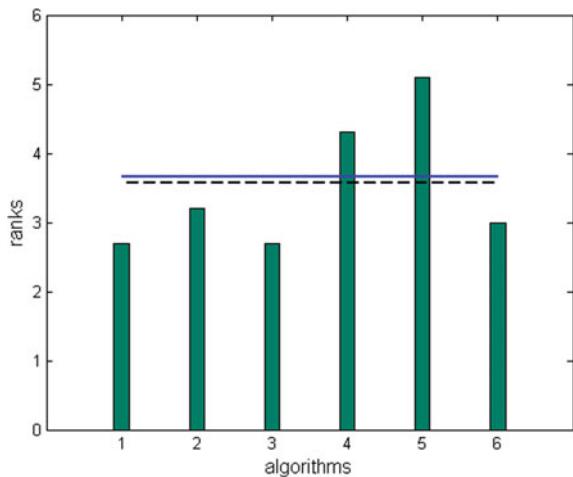
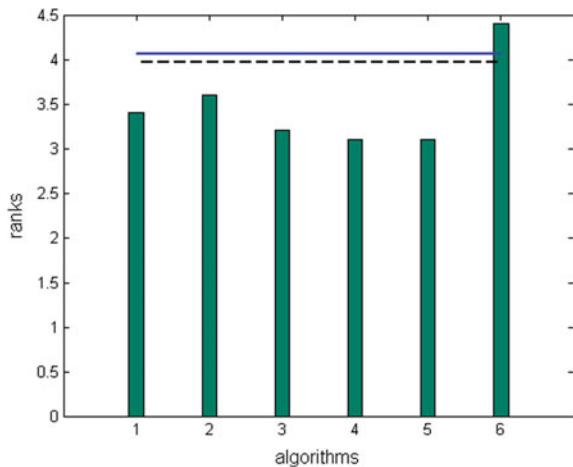
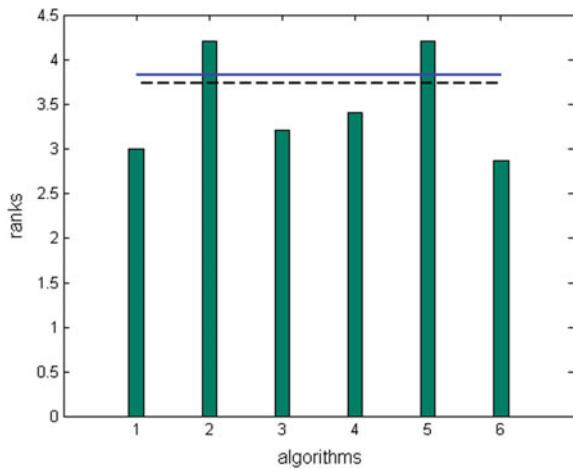


Fig. 3.4 Bonferroni Dunn bar chart for rank based on CPU time



posed new strategy ReDE produces better results for most of the standard function. The comparison of results is performed on the basis of VTR, CPU time and NFE and the graphs are plotted. The proposed strategy is easy and simple to apply and do not disturb the basic structure of DE. The comparison results also show that the proposed

Fig. 3.5 Bonferroni Dunn bar chart for rank based on NFE



strategy performs significantly better than several other classical strategies. ReDE can also be applied upon real time problems and optimization problems. This work can be further extended to various other areas like clustering, image thresholding, image enhancement, image analysis, digital image processing etc. in order to verify the performance of the new mutation strategy that is being created.

Chapter 4

Search Strategy Flower Pollination Algorithm with Differential Evolution



This chapter introduces another new variant of Evolutionary Algorithm named as ssFPA/DE—Search Strategy Flower Pollination Algorithm with Differential Evolution. In this novel approach, the search strategy of FPA algorithm is combined with the efficiency of DE to make ssFPA/DE robust. This variant is then applied on clustering of dataset.

4.1 Introduction

Metaheuristic is an efficient technique to solve complex problems. Research work involving nature inspired concepts are in progress for past several years. Researchers have used only a few characteristics of nature and there is always space for more ideas to be developed. For centuries, solutions to several optimization problems were derived based on various biological systems that exist in nature. Many of them depict improved efficiency in maximization of numerous evolutionary processes. Based on these biological behaviors, algorithms like the genetic algorithm (GA), firefly algorithm and particle swarm optimization algorithms were introduced. Evolutionary algorithm demands steady optimization. Genetic algorithm forms a component of the larger set of evolutionary algorithm that solves several optimization problems by replicating the techniques that exist in nature like selection, crossover, inheritance and mutation.

Storn and Price [1] attempted to change the traditional mutation and crossover operators in GA by another operant, and thereby introduced a differential operator to manage the situation . The algorithm they proposed was called Differential Evolution

(DE). Differential Evolution is a simple and powerful algorithm for optimization. DE has various properties such as robustness, ease of use, speed and compact structure. Numerous works are done to improve the efficiency of DE.

Flower Pollination Algorithm (FPA), developed by Xin She Yang [4], is one of the recent developments in evolutionary algorithm, which uses metaheuristic techniques. It uses the pollination concept from flowering plants. FPA is used for solving constrained and unconstrained optimization problems. FPA has the properties of fast execution and ease to modify. From the biological point of view of reproduction, the main objective of flower pollination is to create offspring for next generation. The more fit the plant is, better will be the reproduction. This is literally an optimization problem of the plant group.

In many professions, analysis of large amount of data in day-to-day work has become inevitable. With the vast amount of information currently available, the necessity to classify and cluster such data has become a necessity. Various techniques for clustering have been developed using various evolutionary algorithms. Evolutionary algorithms are of great interest to many researchers around the world. New algorithms are developed based on biological processes that exist in nature. In addition, different variants of the existing algorithms are also created with researchers working to find the most optimal method. Differential Evolution is an efficient technique that obtains optimum value using randomly selected vectors. However, it falls short in its intelligence to move to its global optimum. FPA, to the contrary, gives both local and global search strategies but it is not as efficient as DE algorithm. Therefore, the search strategy features of FPA are combined with the DE algorithm to yield an efficient algorithm ssFPA/DE. A description of the hybrid algorithm named ssFPA/DE that uses the search strategy of FPA and DE is explained along with their results.

4.2 Proposed Hybrid Method ssFPA/DE

A novel hybrid optimization method ssFPA/DE is proposed by using Differential Evolution and FPA algorithm. First, the initial population is developed using the concept of FPA. Using the concept of local and global pollination, form the different search strategies of local and global type. Subsequently, using the FPA algorithm, generate the best solutions from the initial population and treat it as new population. These best solutions obtained from FPA will further undergo differential evolution. The results obtained from FPA will be the initial population for DE. This population will undergo a series of mutation and crossover. The best individuals will survive as the final population. ssFPA/DE introduces the concept of elitism. Elitism follows the principle that ‘the best must survive’. In ssFPA/DE, the initial population gets refined

and the best individuals in the population survive and move to the next stage. In the second stage, the revised population after FPA is given to DE algorithm where it undergoes a series of mutation and crossover. During this stage, the best individuals from the revised generation gets filtered. The resultant population will have the best of the population. This generates the final population. The proposed algorithm is given below:

Proposed Algorithm:

Objective function $f(x), x = (x_1, x_2, \dots, x_d)$. Initialize n pollen seeds

Select optimal result (g^*) in the original group

State t as number of iterations and a switching probability as $p \in [0,1]$

While (*Max Generation* > t)

 for all n flowers in the group

 If ($p > rand$)

 For Levy distribution, select a step vector L of d dimension

 Do global pollination $x_i^{t+1} = x_i^t + L(x_i^t - g^*)$

 Else

 Select item from a uniform distribution within $[0,1]$

 Do local pollination $x_i^{t+1} = x_i^t + \mathcal{E}(x_i^t - x_{ki}^t)$

 end if

 Compute new results

 If new results are better, then revise them in group

 end for

end while

for every element X in the group do

Select three elements $X_{r1,G}, X_{r2,G}, X_{r3,G}$ from the group which follows
 $X_{r1,G} \neq X_{r2,G} \neq X_{r3,G} \neq X$

Select index R in n randomly where n is the size of the problem to be optimized

Compute the element's new location $V_{i,G}$ as :

$$V_{i,G} = X_{r1,G} + F.(X_{r2,G} - X_{r3,G})$$

for every t , pick a uniformly distributed number r and $C_r \in [0,1]$

If $rand_{j,i} \leq C_r$ or $j = rand_{j,i}$ then

$$\text{Set } U_{j,i,G+1} = V_{j,i,G+1}$$

Else

$$\text{Set } U_{j,i,G+1} = X_{j,i,G+1}$$

end if

end for

If $f(U_{i,G+1}) \leq f(X_{i,G})$ then change the element in the group with the new improved result

Identify the element from the group with the lowest cost or highest fitness return the value as the best candidate result

end for

End

Implementation of ssFPA/DE is done in MATLAB2008b and the results are obtained for few strategies of DE. The values are tabulated for the various benchmark functions. The results obtained are further compared with the original DE approach.

4.3 Experimental Results

The above stated hybrid algorithm is implemented on i7 core processor, 64- bit operating system with 12 GB RAM using MATLAB2008b and a comparative result is obtained with the original DE algorithm. Fifteen different functions are considered

and results are calculated by fixing the value to reach and number of iterations. The value to reach (VTR) is the global minimum or maximum of the function to stop the optimization problem. Tables show results of best value, Number of Function Evaluation (NFE) and CPU time taken by changing the dimensions and VTR value. The results are tabulated for comparison with the traditional DE algorithm in Tables 4.1, 4.2, 4.3 and 4.4. Few of the results obtained and their corresponding graphs are given below.

A comparative analysis is performed and a study is done on each technique. By setting the dimension as 25 and value-to-reach (VTR) as e-015, the best value, a number of function evaluation (NFE) and the CPU time of different function strategies are calculated. The proposed approach provide the optimum value for the standard functions. For some functions like Schwefel, Michalewicz functions, the results are good for both classical DE and the proposed hybrid algorithm. In the case of NFE, the proposed hybrid algorithm provide the best results for most of the functions. Some functions provide best NFE values for both the proposed algorithm and the classical DE. For NFE, DE/best/1 crossover method provide the best results for the proposed algorithm for most of the standard functions. The CPU time for different functions is best for proposed algorithm in almost all cases.

By changing the VTR value to e-014 for dimension of 25, the best value is obtained for proposed algorithm for most of the standard functions. In some functions, the results are good for both classical DE and proposed algorithm. NFE for most of the standard functions are also best for the proposed algorithm. By changing the dimension to 50 and setting the VTR to e-015, best value and NFE are better for most of the standard functions that used the proposed algorithm. CPU time is also best for almost all standard functions using the proposed algorithm.

Based on the values from Table 4.1, Friedman test is applied and the results obtained are tabulated. Table 4.5 represents the values obtained from the test performed on both DE and ssFPA/DE.

4.4 Graphical Representation

The graphical representation of above tabulated values are in Figs. 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.8, 4.9 and 4.10. The graphs represent performance curve of five different function strategies. The x-axis represents the number of function evaluations and y-axis represents the objective function. The graph is plotted for the various values at each iteration for fixed VTR value of e-015 and dimension size of 25.

A comparative study is done based on the above graphs. The study shows that the proposed algorithm take less CPU time for fixed VTR of e-015 for dimension size of 50 and 25. Number of function evaluations (NFE) is best for proposed algorithm especially for the DE/best/1 mutation method. In Sphere function, Beale function, Schwefel function and Michalewicz functions, better results are seen in proposed algorithm.

Table 4.1 Comparative results of Best Value for size = 25 and VTR = 1.e-015 for different strategies of DE and ssFPA/DE

Function	DE					ssFPA/DE				
	DE/best/1	DE/rand/1	DE/best-to-rand/1	DE/best/2	DE/rand/2	DE/best/1	DE/rand/1	DE/best-to-rand/1	DE/best/2	DE/rand/2
f1	9.73e-016	6.219e-016	7.532e-016	9.655e-016	7.17e+0	8.62e-016	8.967e-016	9.13e-016	9.325e-016	6.19e+00
f2	3.265e-016	2.318e-016	1.713e-016	7.587e-016	7.725e-016	5.13e-016	3.253e-016	3.193e-016	3.307e-016	4.4782e-016
f3	3.497e-016	2.0514e-016	6.0738e-016	7.0792e-016	8.35e-016	1.3042e-016	8.7327e-016	8.504e-016	3.45e-016	2.724e-016
f4	-1.8e+003	-2.2523e+003	-7.8403e+001	-1.38e+003	-1.166e+003	-8.009e+002	-4.6523e+002	-4.07e+001	-6.82e+002	-2.808e+002
f5	-7.6399e+00	-7.214e+00	-7.39e+00	-6.959e+00	-6.847e+00	-6.343e+00	-6.89e+00	-8.09e+00	-6.475e+00	-7.036e+00
f6	6.6e-016	8.88e-016	4.43e-016	6.55e-016	8.87e-016	2.22e-016	1.89e-016	6.66e-016	6.23e-016	2.22e-016
f7	3.05e-015	2.9e-001	2.92e-001	2.93e-001	2.89e-001	2.22e-011	2.32e-011	2.21e-011	2.42e-011	2.12e-011
f8	1.6e-016	8.05e-016	3.83e-016	9.12e-016	1.46e-016	4.37e-016	7.51e-016	1.28e-015	1.08e-017	5.25e-016
f9	-1.035e+002	-1.067e+002	-1.05e+002	-1.056e+002	-1.03e+002	-1.06e+002	-1.02e+002	-1.1e+002	-1.05e+002	-1.08e+002
f10	3.31e-015	4.98e-005	6.1e-008	1.93e-005	2.68e+00	1.65e-011	1.62e-011	1.67e-001	1.7e-011	1.63e-011
f11	7.19e-015	6.46e-012	7.89e-015	3.63e-013	3.09e+00	7.09e-013	6.04e-012	7.99e-011	7.16e-012	7.34e-012
f12	3.00e+00	3.00e+00	3.00e+00	3.00e+00	3.00e+00	3.00e+00	3.00e+00	3.00e+00	3.00e+00	3.00e+00
f13	9.99e-011	9.99e-016	1.6e-013	6.56e-013	1.07e+00	3.00e+00	1.53e-012	1.34e-011	1.53e-012	1.55e-012
f14	1.79e+001	1.23e+002	7.47e+001	1.28e+002	1.52e+002	2.68e+001	1.202e+002	7.37e+001	1.4173e+002	1.67e+002
f15	7.6e-016	1.07e-008	7.88e-016	3.9e-009	1.07e+005	7.07e-016	1.052e-008	8.43e-001	5.907e-011	8.307e+004

Table 4.2 Comparative results for NFE on for size = 25 and VTR = 1.e-015 for different strategies of DE and ssFPA/DE

Function	DE						ssFPA/DE			
	DE/best/1	DE/rand/1	DE/best-to-rand/1	DE/best/2	DE/rand/2	DE/best/1	DE/rand/1	DE/best-to-rand/1	DE/best/2	DE/rand/2
f1	2,880,000	3,705,000	3,130,000	3,260,000	5,000,000	281,000	370,000	317,000	3,348,000	5,000,000
f2	48,000	94,000	67,000	85,000	127,000	47,000	89,000	66,000	83,000	137,000
f3	500,000	90,000	61,000	77,000	118,000	49,000	91,000	62,000	80,000	123,000
f4	7000	12,000	13,000	4000	6000	6000	12,000	12,000	5000	6000
f5	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
f6	68,000	148,000	119,000	139,000	224,000	67,000	148,000	148,000	135,000	239,000
f7	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000
f8	45,000	95,000	67,000	77,000	199,000	42,000	97,000	66,000	88,000	292,000
f9	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
f10	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000
f11	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000
f12	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000
f13	2,880,000	4,579,000	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000
f14	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000
f15	609,000	5,000,000	770,000	5,000,000	5,000,000	577,000	5,000,000	788,000	5,000,000	5,000,000

Table 4.3 Comparative results of CPU time for size = 25 and VTR = 1.e-015 for different strategies of DE and ssFPA/DE

Function	DE		ssFPA/DE							
	DE/best/1	DE/rand/1	DE/best-to-rand/1	DE/best/2	DE/rand/2	DE/best/1	DE/rand/1	DE/best-to-rand/1	DE/best/2	DE/rand/2
f1	37.670	139.169	18.318	122.399	225.799	17.886	139.452	17.768	162.0907	226.40
f2	48.163	9.5786	8.263	8.533	8.81	5.997	10.8674	5.31725	8.56	11.5
f3	11.6	8.25	6.817	8.494	8.5762	8.829	8.9977	7.995	7.659	9.48
f4	17.712	5.337	4.3044	4.956	4.5789	4.688	8.27	5.3085	3.14	4.309
f5	2.73	2.217	2.817	2.064	2.074	3.7826	2.902	2.12	1.923	2.2811
f6	36.311	23.22	17.3	18.86	21.19	8.34	20.15	20.05	13.5	11.8
f7	190.07	227.9	261.1	238.23	245.6	234.84	221.12	218.37	210.7	220.8
f8	14.8	16.94	18.4	12.3	15.95	6.17	9.03	8.71	8.9	16.29
f9	12.39	11.8	8.25	8.71	8.07	6.97	7.85	13.97	6.52	11.6
f10	186.78	372.56	345.3	352.3	335.02	314.93	279.73	178.2	289.3	234.5
f11	314.6	310.42	325.56	312.2	300.4	278.01	270.16	265.4	234.2	276.2
f12	257.45	425.8	331.62	312.52	319.5	233.5	231.2	223.4	234.2	245.6
f13	321.7	305.3	346.58	342.76	341.01	240.3	214.5	245.4	256.3	232.3
f14	254.69	316.8	265.4	310.1	323.52	234.7	207.3	220.4	201.35	201.95
f15	52.01	36.23	52.79	324.06	34.53	28.96	199.79	39.57	206.33	195.497

Table 4.4 Comparative results of Best Value after 25 runs and VTR = 1.e-014 for different strategies DE and ssFPA/DE

Function	DE				ssFPA/DE			
	DE/best/1	DE/rand/1	DE/best-to-rand/1	DE/best/2	DE/rand/2	DE/best/1	DE/rand/1	DE/best-to-rand/1
f1	9.34e-015	9.35e-015	9.539e-015	9.942e-015	6.922e+000	8.82e-015	7.91e-015	7.72e-015
f2	4.26e-015	7.72e-015	1.125e-015	1.36e-017	7.5e-015	1.34e-015	8.2e-015	1.6e-015
f3	1.807e-015	7.55e-016	1.95e-015	2.75e-015	6.47e-015	5.89e-016	5.56e-014	5.6e-015
f4	-4.22+002	-4.8e+002	-1.67e+003	-4.47e+003	-1.5e+003	-3.7e+003	-7.04e+002	-7.76e+001
f5	-7.69e+00	-7.64e+00	-6.87e+00	-7.35e+00	-6.98e+00	-7.49e+00	-8.08e+00	-7.18e+00
f6	1.33e-015	1.33e-015	6.66e-016	5.3e-015	1.33e-015	4.4e-015	5.35e-015	4.21e-015
f7	2.92e-001	2.92e-001	2.92e-001	2.92e-001	2.92e-001	2.912e-001	2.79e-001	2.892e-001
f8	4.83e-015	4.42e-015	1.902e-015	3.95e-015	5.14e-015	3.27e-015	6.78e-015	2.96e-015
f9	-9.203e+001	-1.04e+002	-1.066e+002	-1.034e+002	-1.04e+002	-1.05e+002	-1.032e+002	-1.062e+002
f10	5.701e-008	5.212e-005	7.1003e-008	1.73e-005	2.92e-009	2.51e-013	2.45e-013	2.52e-013
f11	7.89e-015	5.02e-015	7.79e-015	3.59e-013	3.213e+000	7.99e-015	5.56e-012	7.99e-011
f12	3.00e+00	3.00e+00	3.00e+00	3.00e+00	3.00e+00	3.00e+00	3.00e+00	3.00e+00
f13	1.477e-002	9.214e-015	7.88e-015	5.07e-009	1.06e+000	7.39e-003	9.65e-001	9.43e-015
f14	3.61e+001	8.17e+001	1.727e+002	1.674e+002	2.48e+001	1.09e+002	8.22e+002	1.48e+002
f15	3.98e+00	1.403e-008	6.9e-015	1.56e-011	7.15e+004	7.39e-015	9.05e-009	8.47e-008

Table 4.5 Test statistics using Friedman's test

N = 25	DE	ssFPA/DE
Chi sq	15.3	14.12
Df	5	5
Asymptotic significance	0	0.004

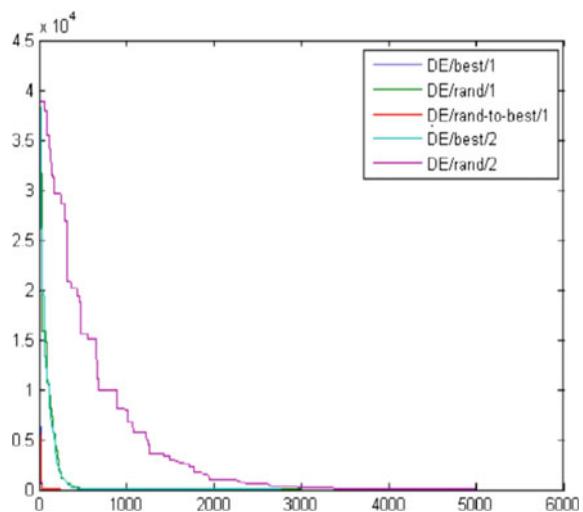
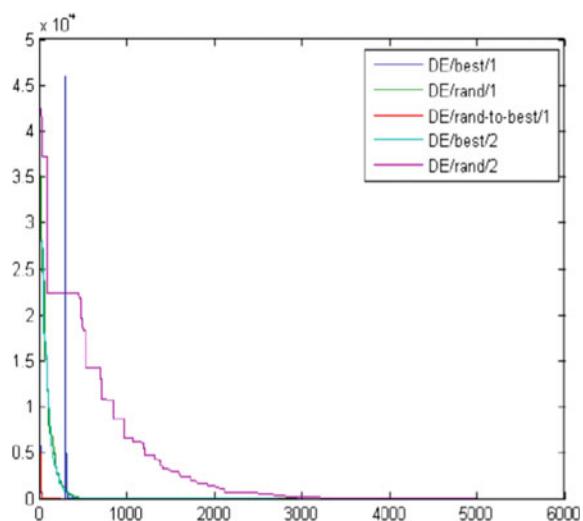
Fig. 4.1 Sphere function using DE algorithm**Fig. 4.2** Sphere function using ssFPA/DE

Fig. 4.3 Booth function using DE algorithm

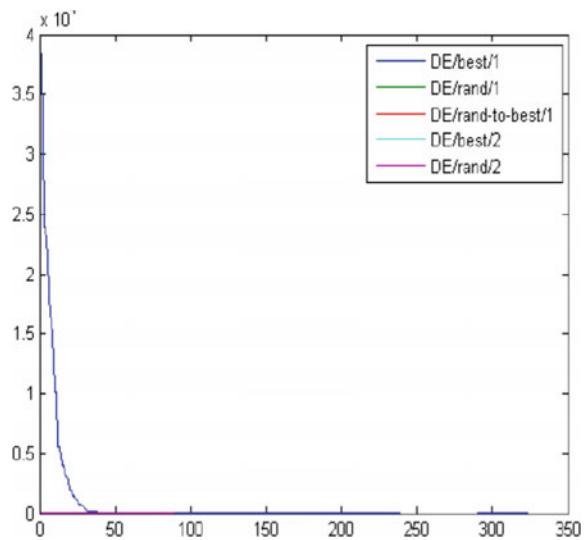
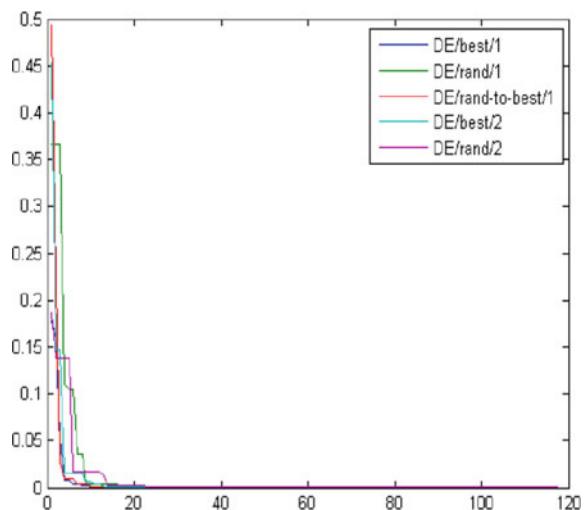


Fig. 4.4 Booth function using ssFPA/DE



4.5 ssFPA/DE in Clustering

This section discusses the implementation of ssFPA/DE in clustering using the k-means algorithm. Each record in a dataset is handled as random sample of population under consideration. Now consider that these datasets are clustered to k random groups. Partitions of the datasets are carried out on the basis of certain objective functions. This is a feature that opts to an optimization problem to minimize or maximize the function from a set of given available alternatives. This function determines

Fig. 4.5 Beale function using DE algorithm

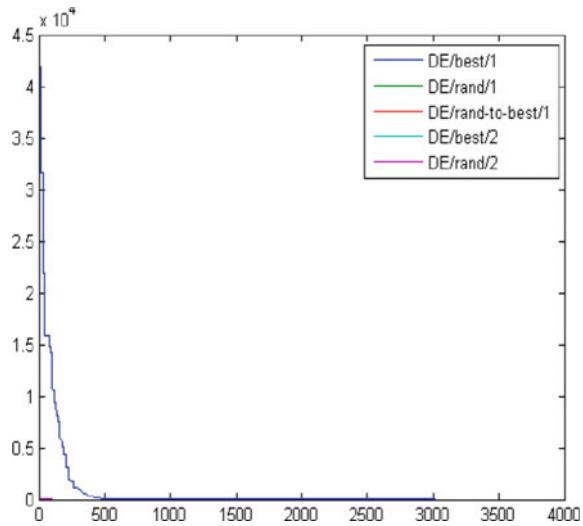
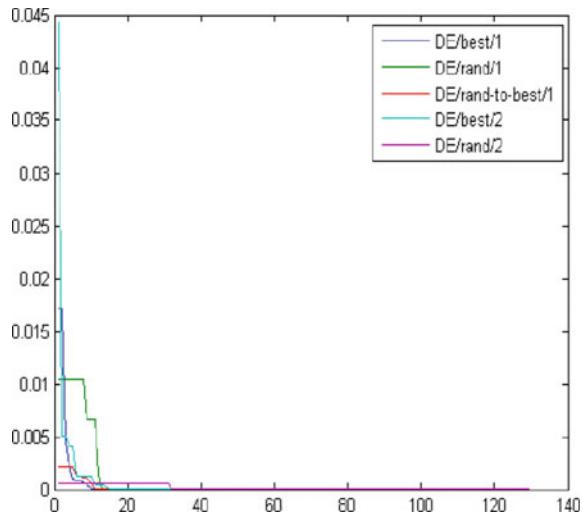


Fig. 4.6 Beale function using ssFPA/DE



how well the chosen solution performs. The fitness of each solution is performed by evaluating the distance between the centroid and the entity point, which is defined as:

$$Fitness(C) = \sum_{j=1}^k \sum_{i=1}^n \|x_i^j - c^j\|^2 \quad (4.1)$$

Fig. 4.7 Michalewicz function using DE algorithm

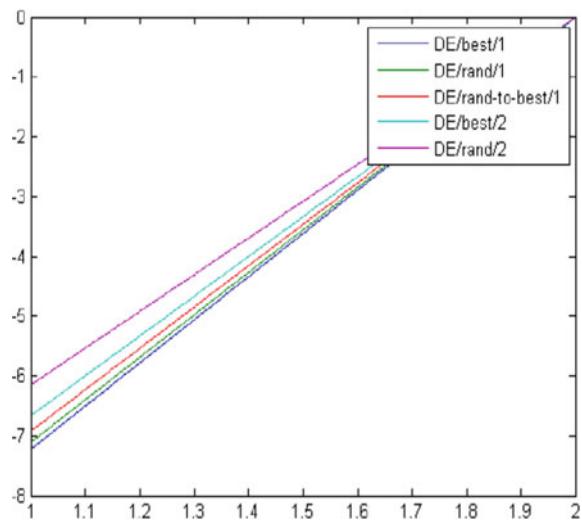
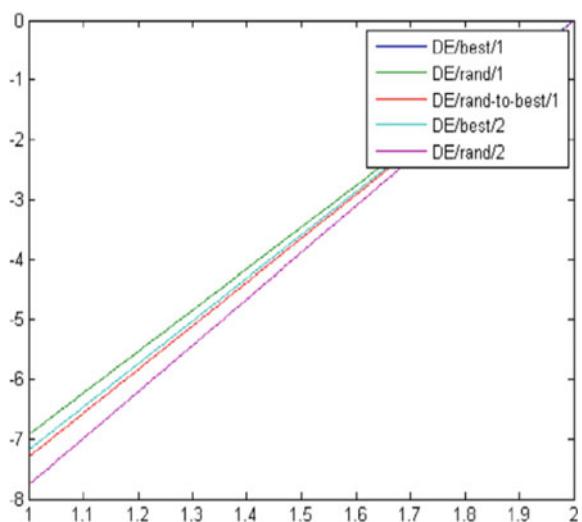


Fig. 4.8 Michalewicz function using ssFPA/DE



where x_i^j is the entity point, c^j is the centroid and $\|x_i^j - c^j\|$ gives the distance between the centroid and the entity point. Then for each k groups, select centroid arbitrarily. Using Euclidean distance method, calculate the distance for each data from its corresponding centroid in the group. K-means algorithm terminates if there is no change in centroid allocated. Result of the k- means algorithm is used as one of the elements of DE algorithm while the rest of the elements are initialized randomly.

Fig. 4.9 Schwefel function using DE algorithm

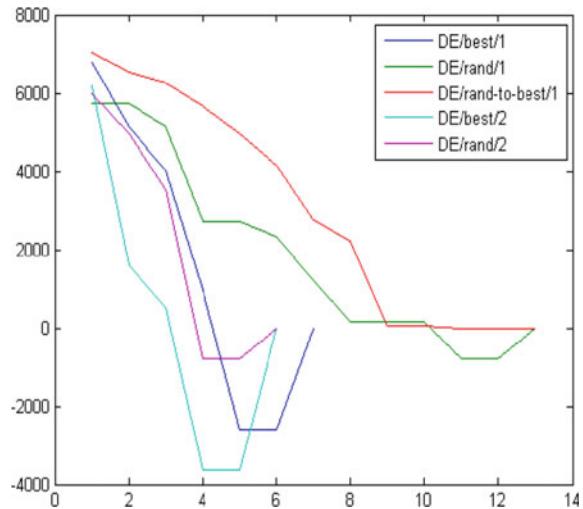
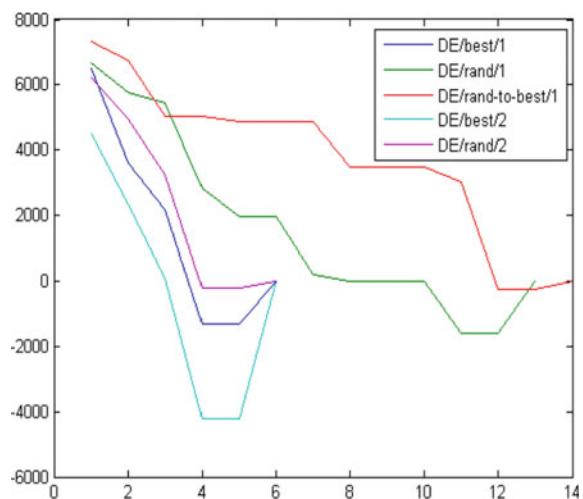


Fig. 4.10 Schwefel function using ssFPA/DE



ssFPA/DE algorithm performs the proposed technique in clustering the data. If the resultant value obtained has better cost function, then the resultant value replaces the least fit value in the population. The ssFPA/DE algorithm terminates if the maximum iteration is exceeded. The flow chart for the clustering technique using the variant of DE is given in Fig. 4.11.

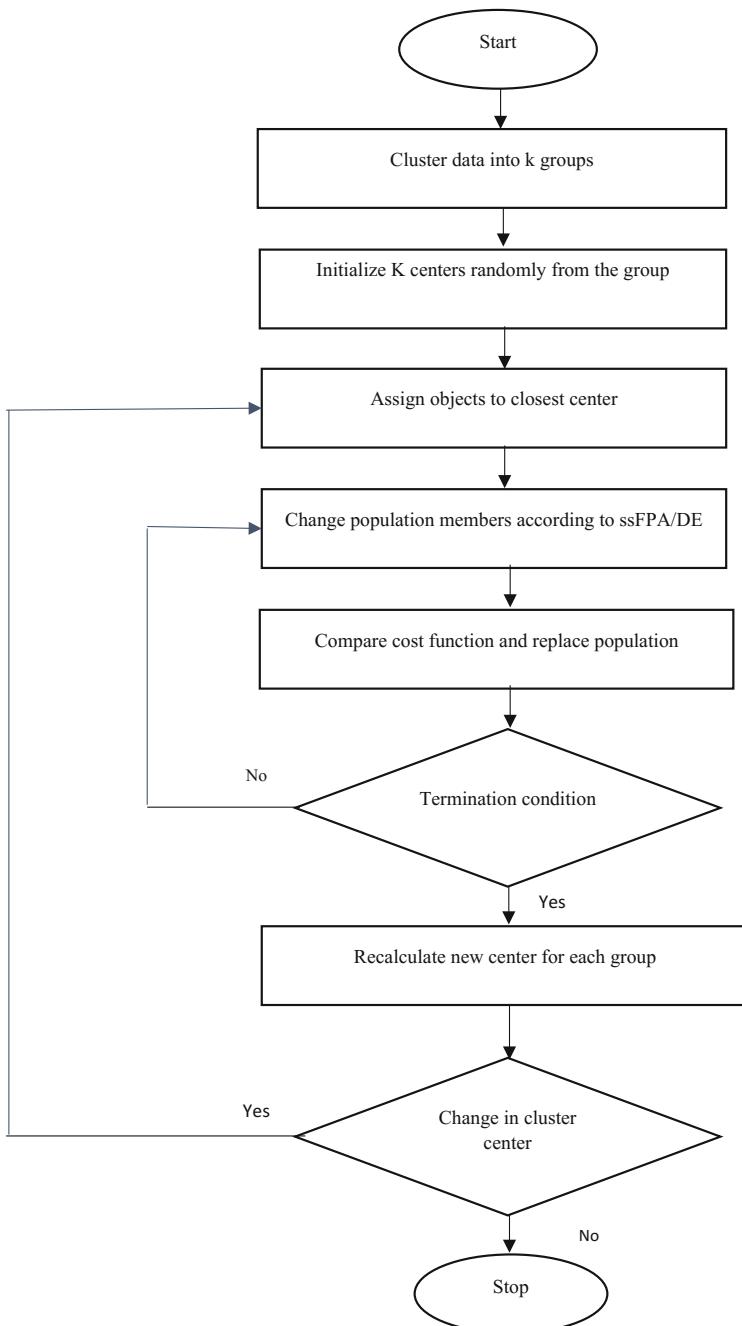


Fig. 4.11 Working of ssFPA/DE in k-means clustering

4.6 Experimental Results on Clustering

Here, the experiment is conducted on fifteen standard datasets with numeric data to compare the performance of the k-means algorithm, Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and classical DE with ssFPA/DE in clustering. The k-means algorithm of clustering is incorporated with GA, PSO, classical DE and ssFPA/DE for performing the data clustering. The clustering algorithms are implemented on i7 core processor, 64 bit operating system with 12 GB RAM using MATLABR2008b and a comparative result is obtained for the various algorithms. The corresponding cluster graph and curve graph for each dataset is obtained. The cluster quality of the clusters obtained is compared. The datasets used are described below:

- Fisher Iris dataset ($n = 150$, $d = 4$, $k = 3$): This is a standard multivariate dataset with 150 inputs for 3 different flower types: *sentosa*, *virginica* and *versicolour*. Here 4 different features of a flower is measured: type, petal width, sepal width and sepal length.
- Morse dataset ($n = 1296$, $d = 5$, $k = 4$): This dataset consist of 36 rows and 36 columns representing the morse code for letters from A to Z and numbers from 0 to 9. Each letter or number is represented with a dash or dot. Morse code consists of 5 elements: short mark, longer mark, intra character gap, short gap and medium gap.
- Hogg dataset ($n = 30$, $d = 6$, $k = 4$): This dataset is a lab based report of bacteria counts in different shipment of milk. Here, 6 sets of bacteria count are taken from 5 different shipment of milk.
- Weather dataset ($n = 60$, $d = 5$, $k = 4$): This is an unsupervised dataset obtained from lab test. It describes main characteristics of weather database for 5 different attributes: outlook, temperature, humidity, windy and play.
- Stock dataset ($n = 950$, $d = 10$, $k = 4$): This is a real dataset obtained from simulated stock returns. Here the stock returns for 10 different companies are being considered in this dataset.
- Temperature dataset ($n = 500$, $d = 19$, $k = 2$): This data set consists of a set of SW-NIR spectra measured on an HP diode array spectrometer. The data set is used to trace the influence of the temperature on vibrational spectra and the consequences for the predictive ability of multivariate calibration models. The data set has been used by several people to test calibration model building. The spectra 19 mixtures of ethanol, water and isopropanol and the spectra of the pure compounds are recorded in a 1 cm cuvette at different temperatures (30, 40, 50, 60, 70 °C). The temperature of the samples is controlled (~0.2° variation). The measurements were set-up according to a mixture design. For measurements at a temperature of 50 °C and higher, the experimental set-up was changed so that the actual temperature in the cuvette could be recorded.

- Glucosinolate dataset ($n = 660$, $d = 11$, $k = 2$): The glucosinolate levels for 11 different glucosinolates were measured at 1, 3, 7 and 14 days after treatment. This measurement was destructive so that for each time-point different plants were analysed. Each measurement was repeated 5 times. Besides the two treatments there was also a control group that received no JA treatment. The data presented here includes the measurements of eleven different glucosinolates 1, 3, 7 and 14 days after treatment of the root induced, shoot induced and control group.
- Moore dataset ($n = 120$, $d = 5$, $k = 2$): This dataset is a lab based report of the biochemical demand on five predictors. The dataset has five predictor values and one response.
- Ovariancancer ($n = 4000$, $d = 95$, $k = 2$): This dataset gives the observation on 4000 patients. The ovarian cancer dataset consists of high-resolution ovarian cancer data generated using the WCX2 protein array. The sample set includes 95 controls and 4000 ovarian cancers.
- Gas ($n = 20$, $d = 2$, $k = 2$): This dataset gives the Gasoline prices around the state of Massachusetts in 1993. The file contains two random samples of prices for a gallon of gas around the state of Massachusetts in 1993. The first sample, price1, contains 20 random observations around the state on a single day in January. The second sample, price2, contains 20 random observations around the state one month later.
- Cities ($n = 2961$, $d = 9$, $k = 2$): This dataset gives the quality of life ratings for U.S. metropolitan areas. The data includes ratings for 9 different indicators of the quality of life in 329 U.S. cities. These are climate, housing, health, crime, transportation, education, arts, recreation, and economics. For each category, a higher rating is better.
- Kmeansdata ($n = 560$, $d = 4$, $k = 3$): It is a four dimensional grouped data provided for clustering with one field.
- Popcorn ($n = 18$, $d = 3$, $k = 2$): It is a dataset showing the popcorn yield by popper type and brand. The columns of the matrix popcorn are brands (Gourmet, National, and Generic). The rows are popper type (Oil and Air.) The study popped a batch of each brand three times with each popper. The values are the yield in cups of popped popcorn.
- Examgrades ($n = 649$, $d = 33$, $k = 3$): This data approach student achievement in secondary education of two Portuguese schools. The data attributes include student grades, demographic, social and school related features and it was collected by using school reports and questionnaires. Two datasets are provided regarding the performance in two distinct subjects: Mathematics (mat) and Portuguese language (por).
- Parts ($n = 144$, $d = 4$, $k = 2$): This dataset shows the measurements on newly machined parts taken at 1 h intervals for 36 h. Each row of the runout matrix contains the measurements for four parts chosen at random. The values indicate, in thousandths of an inch, the amount the part radius differs from the target radius.

Fig. 4.12 Clusters after applying genetic algorithm

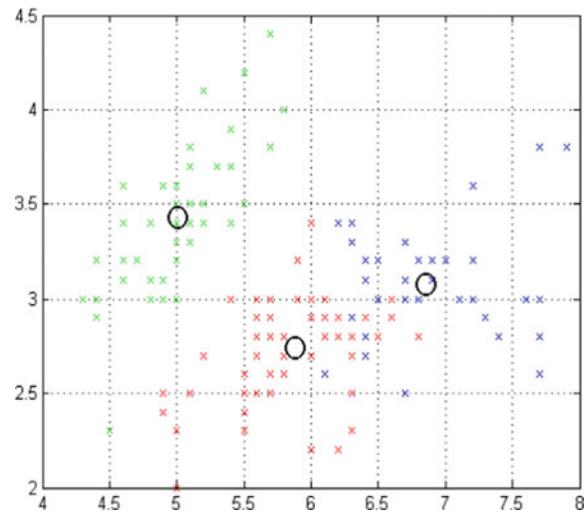
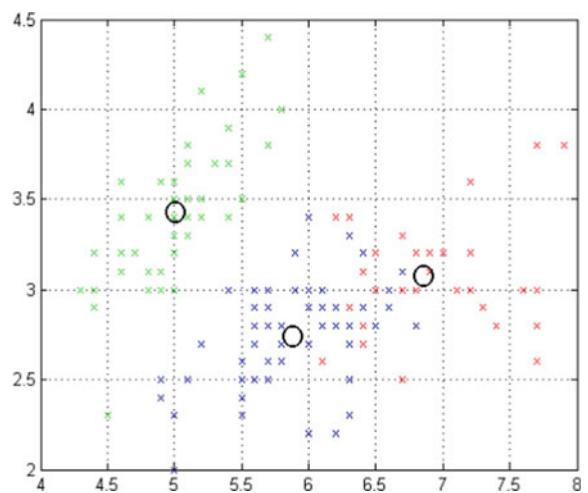


Fig. 4.13 Clusters after applying PSO



These datasets are used as input for clustering and results are obtained for various algorithms under consideration. Figures 4.12 and 4.19 gives the cluster graph and curve graph for the iris data set. The x-axis shows the position of the data and y-axis shows the resultant feature of the iris data set (Figs. 4.13, 4.14, 4.15, 4.16, 4.17 and 4.18).

Fig. 4.14 Clusters after applying classical DE

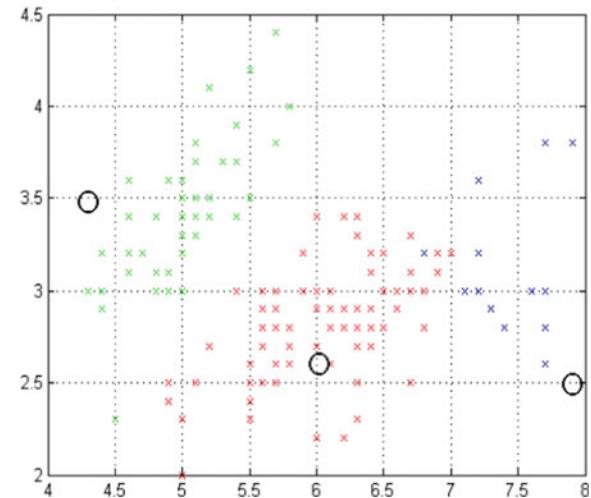
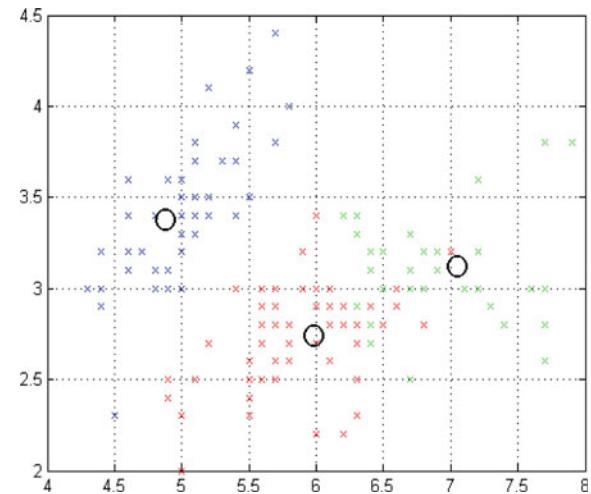


Fig. 4.15 Clusters after applying ssFPA/DE algorithm



4.6.1 Cluster Quality

Quality of the clusters obtained can be compared based on the following parameters:

Intra Cluster Distance

Intra cluster distance calculates the distance of the members in the cluster. It is the sum of squares of distance from the members of each cluster to its centroid. Minimum intra cluster distance gives good clusters. The formula for intra cluster distance is:

Fig. 4.16 Curve graph for genetic algorithm

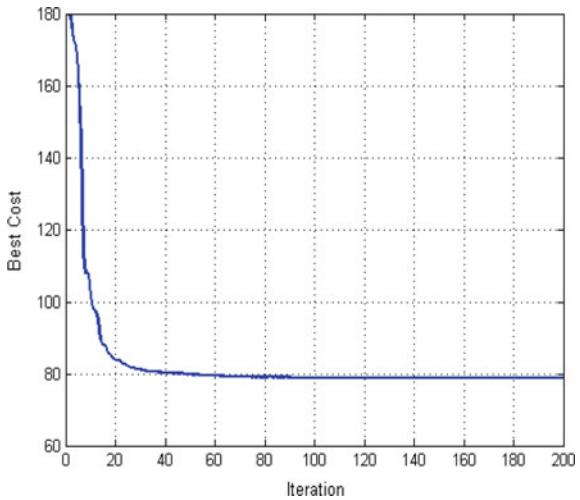
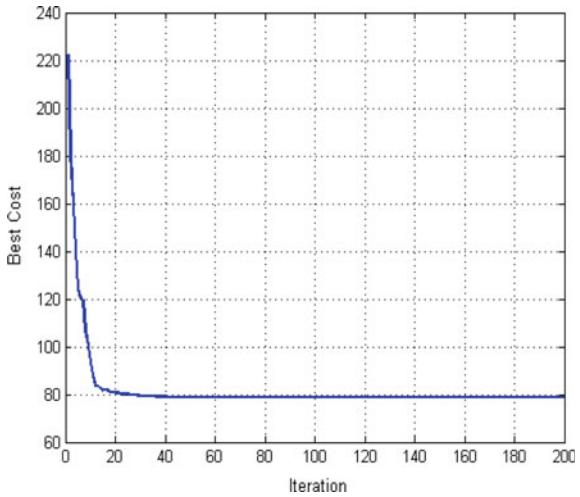


Fig. 4.17 Curve graph for PSO



$$intra = \sum_{j=1}^k \|x_i^j - c_j\|^2 \quad (4.2)$$

where $\|x_i^j - c_j\|$ is the distance between particles and centroid. Lower the intra cluster distance value, better the cluster formed. The comparative results obtained for mean of intra cluster distance is given in Table 4.6.

Fig. 4.18 Curve graph for ssFPA/DE

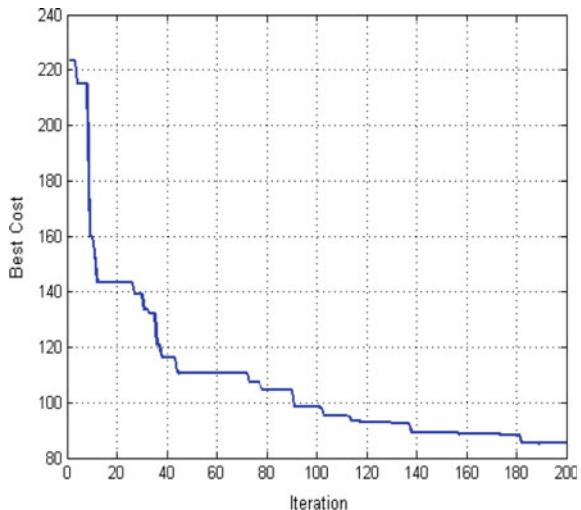
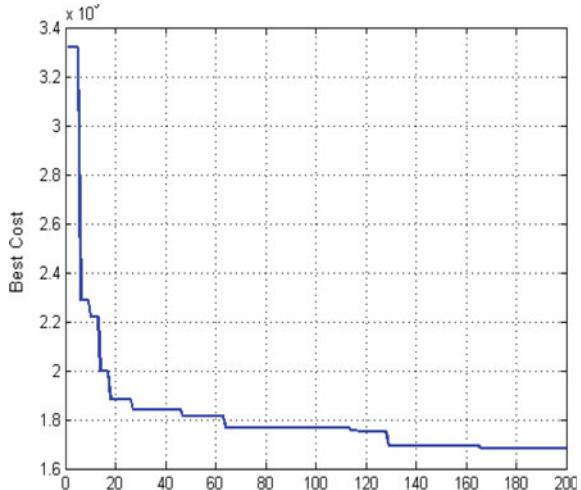


Fig. 4.19 Curve graph for classical DE



Inter Cluster Distance

Inter cluster distance calculates the distance of all the pairs of centroid of different clusters. It is the sum of squares of distance between each cluster centroid. The formula is given as:

$$inter = \min(\|c_i - c_j\|)^2 \quad (4.3)$$

Table 4.6 Mean intra cluster distance for K-means, GA, PSO, DE and ssFPA/DE

Data sets	Mean intra cluster distance				
	K-means	GA	PSO	Classical DE	FSDE
Morse (k = 4)	16.01	16.17	19.467	18.803	16.07
Iris (k = 3)	43.7	48.24	45.78	22.24	20.34
Hogg (k = 4)	120.1	122.3	146.66	107.08	99.04
Weather (k = 4)	1.6×10^5	1.7×10^5	1.8×10^5	1.7×10^5	1.6×10^5
Stockreturns (k = 4)	2011	2134	2345.7	2256.8	1693.03
Temperature (k = 2)	4.2	3.1	2.9	2.82	2.79
Glucosinolate (k = 2)	240.8	232.4	228.6	229.2	215.5
Moore (k = 2)	3.4×10^7	3.42×10^7	3.41×10^7	3.37×10^7	3.3×10^7
Ovariancancer (k = 2)	92,800	91,189	91,200	90,306.1	90,200.2
Gas (k = 2)	32.4	31.2	30.3	27.12	27.23
Cities (k = 2)	6.6×10^{10}	6.61×10^{10}	6.66×10^{10}	6.54×10^{10}	5.4×10^{10}
Kmeansdata (k = 3)	3812	3729	3716	3720.6	3715.4
Popcorn (k = 2)	1.2	1.15	1.12	1.02	0.89
Examgrades (k = 3)	1678.1	16,558.2	16,550.6	16,456.0	16,440.8
Parts (k = 2)	2.34	2.02	1.78	1.565	1.51

here c_i and c_j are the centroids of cluster i and j. Maximum inter cluster distance, better the cluster formed. The comparative results obtained for mean of inter cluster distance of the various algorithms are given in Table 4.7.

Quantization Error

Vector quantization divides large set of data into clusters having almost same number of points closest to them. Goal of vector quantization is to reduce the average quantization error. The formula for quantization error Q_e is given as:

$$Q_e = \sum_{j=1}^k \left[\sum_{i=1}^n \|x_i^j - c^j\|^2 / N_j \right] / k \quad (4.4)$$

where,

c^j is centroid of cluster j

N_j is number of particles of cluster j

k is the number of clusters

$\|x_i^j - c^j\|$ is the distance between particles and centroid.

Lower the quantization error, better is the cluster formed. The result obtained for quantization error is tabulated in Table 4.8.

Table 4.7 Mean inter cluster distance for K-means, GA, PSO, DE and ssFPA/DE

Data sets	Mean inter cluster distance				
	K-means	GA	PSO	Classical DE	FSDE
Morse (k = 4)	230.12	224.14	159.12	241.9	277.55
Iris (k = 3)	3851.3	4015.72	4037.6	4144.86	5781.3
Hogg (k = 4)	4245.2	4542.57	3158.7	4527.9	4549.71
Weather (k = 4)	3.4×10^8	3.6×10^8	3.0×10^8	3.6×10^8	3.6×10^8
Stockreturns (k = 4)	12,876.22	11,660.32	18,330.7	13,206	26,889
Temperature (k = 2)	21.3	20.45	23.56	38.38	39.2
Glucosinolate (k = 2)	1300.2	1289.7	1225.6	1306.6	1312.4
Moore (k = 2)	3.7×10^8	3.8×10^8	4.03×10^8	4.23×10^8	4.35×10^8
Ovariancancer (k = 2)	6.6×10^5	6.4×10^5	6.4×10^5	6.7×10^5	6.7×10^5
Gas (k = 2)	2314.5	2435.4	2432.8	2652.3	2772.1
Cities (k = 2)	4.12×10^{10}	4.4×10^{10}	4.6×10^{10}	4.9×10^{10}	5.2×10^{10}
Kmeansdata (k = 3)	16,453.4	14,325.6	14,001.6	15,789.9	16,645.5
Popcorn (k = 2)	30.2	31.6	33.8	36.3	38.9
Examgrades (k = 3)	1.5×10^4	1.6×10^4	1.6×10^4	1.6×10^4	1.7×10^4
Parts (k = 2)	17.67	15.23	15.05	18.95	21.45

Table 4.8 Quantization Error for K-means, GA, PSO, DE and ssFPA/DE

Data sets	Quantization error				
	K-means	GA	PSO	Classical DE	FSDE
Morse (k = 4)	4.03	4.04	4.86	4.7	4.01
Iris (k = 3)	14.03	16.08	15.25	7.41	6.78
Hogg (k = 4)	28.7	30.57	36.65	26.7	24.76
Weather (k = 4)	4.2×10^4	4.2×10^4	4.5×10^4	4.2×10^4	4.2×10^4
Stockreturns (k = 4)	530.3	533.5	586.4	564.2	423.2
Temperature (k = 2)	1.35	1.54	1.87	1.41	1.32
Glucosinolate (k = 2)	115.9	117.3	115.8	114.6	114.1
Moore (k = 2)	1.5×10^7	1.6×10^7	1.7×10^7	1.6×10^7	1.5×10^7
Ovariancancer (k = 2)	22,789.5	23,112	22,890	22,576.5	22,500.1
Gas (k = 2)	6.67	6.61	6.54	6.78	6.73
Cities (k = 2)	3.2×10^7	3.3×10^7	3.3×10^7	3.2×10^7	3.1×10^7
Kmeansdata (k = 3)	925.3	928.6	927.6	930.1	930
Popcorn (k = 2)	0.37	0.38	0.43	0.33	0.30
Examgrades (k = 3)	5527	5531	5520	5485.3	5476
Parts (k = 2)	0.50	0.61	0.55	0.52	0.43

Table 4.9 Execution time for K-means, GA, PSO, DE and ssFPA/DE

Data sets	Execution time				
	K-means	GA	PSO	Classical DE	FSDE
Morse (k = 4)	14.8	25.1	24.3	17.3	15.46
Iris (k = 3)	12.4	25.018	15.34	25.03	14.33
Hogg (k = 4)	24.1	25.54	26.34	26.77	24.7
Weather (k = 4)	46.3	55.34	56.67	60.65	48.7
Stockreturns (k = 4)	14.32	16.45	15.56	36.53	15.32
Temperature (k = 2)	130.4	128.4	132.2	127.1	125.4
Glucosinolate (k = 2)	8.03	8.12	8.8	8.9	7.4
Moore (k = 2)	8.14	8.87	8.34	9.99	9.77
Ovariancancer (k = 2)	104.5	105.2	105.6	103.8	101.4
Gas (k = 2)	8.5	9.1	9.4	8.82	8.43
Cities (k = 2)	37.1	37.6	36.7	37.24	35.4
Kmeansdata (k = 3)	111.7	113.4	114.3	112.4	110.4
Popcorn (k = 2)	7.2	6.7	7.0	6.52	6.12
Examgrades (k = 3)	11.3	10.67	12.3	11.23	10.34
Parts (k = 2)	7.6	7.1	6.5	6.7	6.1

Execution Time

Total time taken for the execution of task is called execution time. Lower the execution time, better the cluster. Execution time for the various algorithms is given in Table 4.9.

4.6.2 Validation Index

There are various quantitative evaluation techniques available to test the cluster quality and these are known as validation index. It is used as a tool by researchers to test the cluster result. Internal quality compares different set of clusters without reference to external knowledge. A good clustering technique has high within cluster similarity and low inter cluster similarity. Here, two validation indexes are calculated namely Davies Bouldin (DB) index and Calinski Harabasz (CH) index.

Davies Bouldin (DB) Index

It is a matrix for evaluating the cluster algorithm. It is a function of ratio of sum of intra-distances to inter distances [2]. If $R_{i,j}$ be the measure of clustering scheme, $M_{i,j}$ is the separation between i and j cluster and S_i is the within cluster scatter for cluster I, then the DB index is defined as the ratio of S_i and $M_{i,j}$ which follows the following properties:

Table 4.10 DB Index for K-means, GA, PSO, DE and ssFPA/DE

Data sets	DB index				
	K-means	GA	PSO	Classical DE	FSDE
Morse (k = 4)	1.20	1.22	1.19	1.23	1.12
Iris (k = 3)	0.63	0.78	0.68	0.64	0.61
Hogg (k = 4)	0.432	0.43	0.45	0.41	0.40
Weather (k = 4)	0.41	0.45	0.38	0.35	0.34
Stockreturns (k = 4)	3.24	4.34	4.23	4.54	2.16
Temperature (k = 2)	1.23	1.43	1.20	1.35	1.03
Glucosinolate (k = 2)	3.7	3.12	3.2	3.4	2.8
Moore (k = 2)	0.7	0.56	0.67	0.8	0.43
Ovariancancer (k = 2)	1.46	1.55	1.65	1.45	1.34
Gas (k = 2)	1.21	1.12	1.10	1.06	1.03
Cities (k = 2)	7.1	7.54	7.6	7.5	6.3
Kmeansdata (k = 3)	0.81	0.43	0.76	0.89	0.44
Popcorn (k = 2)	1.20	1.34	1.32	1.21	1.12
Examgrades (k = 3)	4.5	5.1	5.5	4.3	4.1
Parts (k = 2)	1.89	2.01	2.4	2.5	1.8

1. $R_{i,j} \geq 0$.
2. $R_{i,j} = R_{j,i}$.
3. When $S_i \geq S_j$ and $M_{i,j} = M_{i,k}$ then $R_{i,j} > R_{i,k}$.
4. When $S_i = S_j$ and $M_{i,j} \leq M_{i,k}$ then $R_{i,j} > R_{i,k}$.

Here, lower the value of DB index, better the separation and closeness of the data inside the cluster. The formula for DB index is given as:

$$DB = \frac{1}{k} \sum_{i=1}^k \max_{i \neq j} \left\{ \frac{d(x_i) - d(x_j)}{d(c_i, c_j)} \right\} \quad (4.5)$$

where,

k number of clusters

i, j cluster labels

$d(x_i)$ and $d(x_j)$ all samples in cluster I and j to respective cluster centroids and $d(c_i, c_j)$ is the distance between these centroids.

The comparative results obtained for the DB index for the various algorithms are tabulated in Table 4.10.

Table 4.11 CH index for K-means, GA, PSO, DE and ssFPA/DE

Data sets	CH Index				
	K-means	GA	PSO	Classical DE	FSDE
Morse (k = 4)	0.78	0.67	0.67	0.64	1.065
Iris (k = 3)	0.32	0.22	0.301	0.21	0.34
Hogg (k = 4)	0.17	0.168	0.23	0.20	0.26
Weather (k = 4)	0.14	0.146	0.133	0.143	0.148
Stockreturns (k = 4)	2.12	2.048	2.01	1.075	2.48
Temperature (k = 2)	1.3	0.89	0.95	0.71	1.12
Glucosinolate (k = 2)	1.89	2.04	2.13	2.21	2.24
Moore (k = 2)	1.2	0.23	0.45	0.536	1.25
Ovariancancer (k = 2)	0.32	0.55	0.43	0.69	1.3
Gas (k = 2)	0.4	0.38	0.24	0.53	0.8
Cities (k = 2)	5.6	4.3	4.44	5.61	6.1
Kmeansdata (k = 3)	0.6	0.47	0.45	0.56	0.73
Popcorn (k = 2)	0.12	0.43	0.32	0.23	0.8
Examgrades (k = 3)	2.21	1.45	2.78	2.95	3.1
Parts (k = 2)	1.12	1.5	1.34	1.8	2.1

Calinski Harabasz (CH) Index

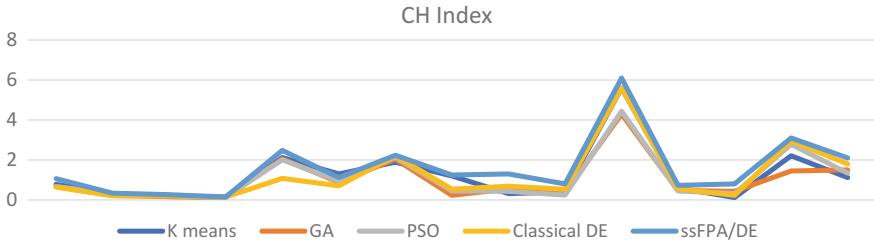
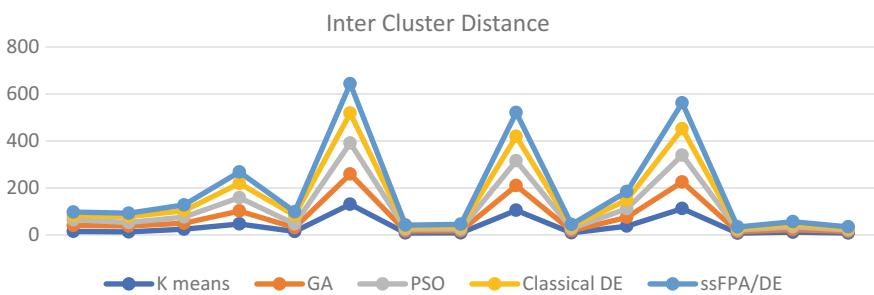
It is another method for calculating cluster quality. It is used to evaluate the optimal number of clusters [3]. Higher the value of CH index, the better the cluster that is formed. The formula for CH index is given as:

$$CH = \frac{\text{trace}(S_B)}{\text{trace}(S_W)} \cdot \frac{n_p - 1}{n_p - k} \quad (4.6)$$

where S_B is between cluster scatter matrix, S_W is the internal scatter matrix, n_p is the number of clustered samples and k is the number of clusters. The comparative results obtained for the CH index is given in Table 4.11.

4.6.3 Graphical Representation

The above tabulated values of Cluster Quality and Validation index is depicted graphically. Figures 4.20 and 4.21 shows the performance curve for CH index in validation index and inter cluster distance in cluster quality. The x-axis represents the different

**Fig. 4.20** Curve for CH index**Fig. 4.21** Curve for mean inter cluster distance

datasets used and y- axis represents the value obtained. The line graph shows that the values obtained for the ssFPA/DE is significantly better than the values obtained from classical evolutionary approaches. The values are recorded for fifteen different datasets.

4.7 Summary

In the proposed work, a hybrid algorithm is suggested using DE and Flower Pollination Algorithm. The proposed approach show that ssFPA/DE is outperforming the classical DE algorithm. Graphical results also show better performance for the new technique. Further work is done to apply this hybrid algorithm in the field of clustering. The results of various algorithms are collected and the results are compared. The results obtained from ssFPA/DE in data clustering is found to be more efficient than other evolutionary approaches.

References

1. Storn, R., Price, K.: Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J. Global Optim.* **11**(4), 341–359 (1997)
2. Davies, D.L., Bouldin, D.W.: A cluster separation measure. *IEEE Trans. Pattern Anal. Mach. Intell.* **2**, 224–227 (1979)
3. Caliński, T., Harabasz, J.: A dendrite method for cluster analysis. *Commun. Stat.-Theor. Methods* **3**(1), 1–27 (1974)
4. Yang, X.S.: Flower pollination algorithm for global optimization. In: International conference on unconventional computing and natural computation, pp. 240–249. Springer, Berlin, Heidelberg (2012)

Chapter 5

Forced Strategy Differential Evolution Used for Data Clustering



This chapter introduces another approach of Differential Evolution algorithm named as FSDE—Forced Strategy Differential Evolution. FSDE uses two control parameters: a constant parameter and a varying parameter. By using two control parameters, the efficiency of FSDE improves greatly. This variant is then applied on clustering of data.

5.1 Introduction

With the ever evolving and rapid progress in the area of research, the amount of data that needs to be processed or produced has increased multi fold. With this increase in the data collection, there arise major challenges in the field of data mining. Data mining or knowledge discovery is the technique of identifying the data from various aspects and then proper categorization of these data information. The data mining software analyse the relationships and patterns that exist within the data. Traditionally, data mining technique has been classified under two types namely supervised learning and unsupervised learning. Supervised learning technique is used for classifications and predictions, whereas unsupervised learning techniques are those in which no predictions or classifications are possible.

Clustering of data is an unsupervised learning technique. Cluster Analysis is an emerging area of interest to many researchers in the field of data mining. Clustering divides data into useful and meaningful groups called clusters. The main aim of clustering is that the objects or data in a group should be similar to one another and dissimilar to the objects in the other group. Clustering determines the intrinsic grouping within a collection of unlabelled data. It is a major method for statistical

data analysis to be used in the field of machine learning, pattern recognition, image processing, data compression and information retrieval. Jain [1] has summarized clustering technique and the various methods used. Clustering of data plays a vital role in efficient data mining, voice recognition, web mining, market analysis etc. Fast and accurate clustering of data plays an important role in the field of automatic information retrieval system. It is considered as a multi objective optimization problem. It involves an iterative task of trial and error. Clustering can be classified as hard clustering and soft clustering. In hard clustering, each object belong to one cluster or do not belong to any cluster. In soft clustering or fuzzy clustering, each object may belong to more than one cluster [2]. Clustering algorithm can be categorized into hierarchical and partitional algorithms. In hierarchical clustering, a hierarchy of partitions is constructed and a dendrogram representation is created. In this technique, each partition is grouped within the partition of next level in the hierarchy. In partitional clustering, a single partition is constructed with a given number of non-overlapping clusters. The main disadvantage of partitional clustering is to find partition of data with a specified number of clusters which minimizes within cluster differences. Partitional algorithms are iterative and usually converge to local minima.

The simplest and most popular partitional clustering algorithm is the k-means technique which was coined by MacQueen [3]. Here a given set of N data are partitioned into k different clusters. Grouping of data is done by minimizing the Euclidian distance between the data and centroid. It is one of the simplest unsupervised learning algorithms used for clustering. This algorithm is significantly sensitive to initial randomly selected centroid. Result of k-means algorithm depends on the initial mean values and so frequently suboptimal partitions are found. As k-means algorithm may converge to suboptimal partitions, some stochastic optimization approach is used to avoid this situation and thereby to find a globally optimum solution. Such problems can be solved using the evolutionary algorithm. Coello et al. [4] stated that evolutionary algorithm work in a robust and efficient manner for clustering. Evolutionary algorithm, which is a part of evolutionary computing, uses biological methods of reproduction, recombination, mutation and selection. Storn and Price [5] introduced Differential Evolution (DE) algorithm which follows the concepts of the evolutionary algorithm. DE is simple, stochastic algorithm based on the population that help to solve optimization problems. The effectiveness and performance of DE is determined by the control parameters and test vector generation strategy. Many variants are designed by changing these strategies and control of test vector parameters.

The aim of this work is to present a variant of Differential Evolution approach. This approach is then combined with k-means to be applied on a clustering problem. Here, a variant of DE algorithm named as Forced Strategy Differential Evolution (FSDE) is introduced. The experimental results obtained from FSDE are explained in the successive sections. Then a detailing is done on the implementation of FSDE and other evolutionary algorithms on k-means approach for application on data clustering.

5.2 Proposed Variant of Differential Evolution

A proposed new strategy for mutation called FSDE involves the best solution vector $X_{best,G}$. It coincides faster as compared to the traditional strategies which have only random vectors. Here, two control parameters are being used. The parameter F known as mutation factor takes a constant value between (0,2) while the new parameter N takes a varying value which lies between (0,1). As two different control parameters are considered, the value of donor vector is improved greatly, and hence, the efficiency of DE algorithm is enhanced profoundly. The proposed strategy is given as:

$$V_{i,G} = X_{r1,G} + N \cdot ((X_{best,G} - X_{r2,G}) - F \cdot (X_{best,G} - X_{r3,G})) \quad (5.1)$$

Here, the random selection of base vector prevents the strategy from becoming greedy in nature.

5.3 Test Problem

The above stated hybrid algorithm FSDE is implemented on i7 core processor, 64 bit operating system with 12 GB RAM using MATLAB2008b and a comparative result is obtained with the original DE algorithm. The traditional mutation strategies are replaced with the proposed mutation strategy and FSDE is composed. Fifteen different functions are considered and the results are calculated by fixing the value to reach and number of iterations. The maximum number of iteration is fixed as 5000 and the maximum number of evaluations as 5,000,000. Results are tabulated by varying the dimension and value to reach. The results are tabulated for comparison with the existing algorithms. The column labelled significance show if the result obtained for FSDE is better than other strategies. Few of the results obtained and their corresponding graphs are given in Tables 5.1, 5.2 and 5.3.

A comparative analysis is performed and study is done on DE/best/1, DE/rand/1, DE/best-to-rand/1, DE/best/2, DE/rand/2 and FSDE. By setting the value-to-reach (VTR) and dimensions, the best value, number of function evaluation (NFE) and the CPU time of different function strategies of Booth, Ackley, Rastrigin etc. are calculated. In some functions, the results are good for both classical DE and proposed algorithm. The overall result obtained shows that the FSDE approach is performing better than the classical DE approach. The Friedman statistical test run is conducted on FSDE algorithms to validate the results. Based on the values from Table 5.1, the Friedman test was applied and the results are tabulated in Table 5.4. The ranks obtained after the Friedman test is tabulated in Table 5.5.

The above tables show that the new mutation strategy has significant performance in comparison to the existing mutation strategies. The rank obtained on the basis of NFE is the best for FSDE. The rank obtained on the basis of best value is comparatively better for best value case. These rankings obtained on the basis of Freidman's

Table 5.1 Comparative results of Best Value for VTR = 1.0e-015 for different strategies of DE with FSDE

Function	D	VTR	DE	FSDE				Significance
				DE/best/1	DE/rand/1	DE/best-to-rand/1	DE/best/2	
Sphere	50	1.e-015	9.73e-016	6.9e-016	7.532e-016	9.655e-016	7.17e+0	6.04e-016 +
	25	1.e-015	9.34e-015	9.35e-015	9.539e-015	9.42e-015	6.92e+000	8.9e-016 +
	75	1.e-015	7.18e-013	6.77e-013	9.4e-013	7.5e-013	9.01e+00	6.69e-013 +
	100	1.e-015	9.5e-016	8.96e-016	9.07e-016	8.3e-016	5.4e-004	7.8e-016
	50	1.e-015	3.265e-016	2.318e-016	3.713e-016	7.587e-016	7.725e-016	5.95e-016 -
Beale	25	1.e-015	4.29e-015	7.72e-015	1.125e-015	1.36e-017	7.5e-015	3.6e-016 -
	75	1.e-015	4.4e-013	4.57e-013	7.3e-013	9.01e-013	5.8e-013	3.5e-013
	100	1.e-015	7.6e-016	3.012e-016	4.76e-017	3.18e-016	5.47e-016	2.23e-016
	50	1.e-015	3.497e-016	2.0514e-016	6.0738e-016	7.0792e-016	8.35e-016	3.28e-016 -
	25	1.e-015	1.807e-015	7.55e-016	1.95e-015	2.75e-015	6.47e-015	9.4e-016 -
Booth	75	1.e-015	2.13e-013	4.7e-014	8.77e-013	3.7e-013	8.3e-013	3.9e-013 -
	100	1.e-015	9.9e-016	8.25-016	8.17e-016	4.6e-017	5.78e-016	2.96e-016 +
	50	1.e-015	-1.8e+003	-2.253e+003	-7.8403e+001	-1.38e+003	-1.66e+003	-4.56e+003 NA
	25	1.e-015	-4.22e+002	-4.8e+002	-1.67e+003	-4.47e+003	-1.5e+003	-2.5e+003 NA
	75	1.e-015	-8.16e+003	-3.4e+003	-7.37e+003	-1.95e+003	-3.7e+003	-8.7e+002 NA
Schwefel	100	1.e-015	-1.09e+004	-9.52e+003	-1.2e+003	-1.4e+003	-5.94e+003	-1.93e+003 NA
	50	1.e-015	-7.6399e+00	-7.214e+00	-7.39e+00	-6.959e+00	-6.847e+00	-7.34e+00 NA
	25	1.e-015	-7.69e+00	-7.64e+00	-6.87e+00	-7.35e+00	-6.98e+00	-7.1e+00 NA
	75	1.e-015	-7.08e+00	-7.68e+00	-7.92e+00	-6.06e+00	-7.9e+00	-8.4e+00 NA
	100	1.e-015	-1.95e+001	-1.98e+001	-1.97e+001	-1.96e+001	-1.94e+001	NA

(continued)

Table 5.1 (continued)

Function	D	VTR	DE	FSDE				Significance
				DE/best/1	DE/rand/1	DE/best-to-rand/1	DE/best/2	
Schaffer N.2	50	1.e-015	6.6e-016	8.88e-016	4.43e-016	6.55e-016	8.87e-016	2.22e-016 +
	25	1.e-015	1.33e-015	1.33e-015	6.66e-016	5.3e-015	1.33e-015 0	-
	75	1.e-015	3.42e-014	6.3e-013	6.6e-013	8.7e-013	8.6e-013	4.4e-013 +
	100	1.e-015	4.44e-016	0	6.66e-016	4.44e-016	8.88e-016	0 -
Schaffer N.4	50	1.e-015	3.05e-015	2.9e-001	2.92e-001	2.93e-001	2.89e-001	2.82e-001 +
	25	1.e-015	2.92e-001	2.92e-001	2.92e-001	2.92e-001	2.92e-001	NA
	75	1.e-015	2.9e-001	2.9e-001	2.9e-001	2.9e-001	2.9e-001	NA
	100	1.e-015	2.92e-001	2.92e-001	2.92e-001	2.92e-001	2.92e-001	NA
HimmelBlau	50	1.e-015	1.6e-016	8.05e-016	3.83e-016	9.12e-016	1.46e-016	3.35e-016 -
	25	1.e-015	4.83e-015	4.42e-015	1.902e-015	3.95e-015	5.14e-015	1.59e-015 +
	75	1.e-015	9.6e-015	2.7e-013	6.9e-013	8.4e-013	3.4e-013	5.1e-013 -
	100	1.e-015	9.19e-016	6.71e016	9.91e-016	8.65e-016	9.13e016	5.12e-016 +
Bird	50	1.e-015	-1.035e+002	-1.067e+002	-1.05e+002	-1.065e+002	-1.03e+002	-1.029e+002 NA
	25	1.e-015	-9.303e+001	-1.04e+002	-1.066e+002	-1.034e+002	-1.04e+002	-1.066e+002 NA
	75	1.e-015	-9.4e+001	-1.06e+002	-1.0e+002	-1.06e+002	-1.05e+001	-1.07e+001 NA
	100	1.e-015	-1.06e+002	-1.02e+002	-1.05e+002	-1.04e+002	-1.058e+002	-1.02e+002 NA
Extended Cube	50	1.e-015	3.31e-015	4.98e-005	6.1e-008	1.93e-005	2.68e+00	5.40e-008 +
	25	1.e-015	5.701e-008	5.212e-005	7.1003e-008	1.73e-005	2.92e+009	5.86e-007 -
	75	1.e-015	8.06e-008	4.9e-008	6.55e-008	8.8e+001	1.29e-009	5.7e-007 +
	100	1.e-015	7.01e-008	1.4e-005	9.3e-008	4.6e-008	3.4e-006	1.78e-006 -

(continued)

Table 5.1 (continued)

Function	D	VTR	DE	FSDE				Significance
				DE/best/1	DE/rand/1	DE/best-to-rand/1	DE/best/2	
Ackley	50	1.e-015	7.19e-015	6.46e-012	7.99e-015	3.63e-013	3.09e+00	7.99e-015
	25	1.e-015	7.99e-015	5.02e-015	7.99e-015	3.59e-013	3.213e+00	4.4e-015
	75	1.e-015	9.81e-013	6.24e-012	9.5e-013	9.8e-013	3.11e+00	9.95e-013
Gold	100	1.e-015	1.51e-014	2.24e-013	7.99-015	2.93e-014	1.4e-014	4.44e-015
	50	1.e-015	3.00e+00	3.00e+00	3.00e+00	3.00e+00	3.00e+00	NA
	25	1.e-015	3.00e+00	3.00e+00	3.00e+00	3.00e+00	3.00e+00	NA
Griewank	75	1.e-015	3.0e+00	3.0e+00	3.0e+00	3.0e+00	3.0e+00	NA
	100	1.e-015	3.0e+00	3.0e+00	3.0e+00	3.0e+00	3.0e+00	NA
	50	1.e-015	9.99e-016	9.99e-016	1.6e-013	6.56e-013	1.07e+00	7.77e-015
Rastrigin	25	1.e-015	1.477e-002	9.214e-015	7.88e-015	5.07e-009	1.06e+00	9.9e-016
	75	1.e-015	7.1e-013	9.8e-013	9.5e-013	9.8e-003	1.05e+00	9.7e-013
	100	1.e-015	4.67e-002	8.88e-016	9.98e-016	9.9e-016	7.41e-002	8.88e-016
Rosenbrock	50	1.e-015	1.79e+001	1.23e+002	7.47e+001	1.28e+002	1.52e+002	2.98e+001
	25	1.e-015	3.61e+001	1.181e+002	8.17e+001	1.727e+002	1.674e+002	0
	75	1.e-015	2.48e+001	1.16e+002	7.75e+001	1.26e+002	1.611e+002	8.24e+001
	100	1.e-015	1.09e+001	6.71e+001	5.43e+001	1.08e+001	2.34e+001	1.23e+001
	50	1.e-015	9.6e-016	1.07e-008	7.88e-016	3.9e-009	1.07e+005	1.107e+001
	25	1.e-015	3.98e+000	1.403e-008	6.9e-015	1.56e-011	7.15e+004	8.5e-016
	75	1.e-015	6.12e-013	1.28e-008	7.04e-013	6.2e-011	1.02-014	9.7e-012
	100	1.e-015	7.68e-016	1.77e-014	8.86e-006	7.67e-016	5.8e+001	8.83e-016

Table 5.2 Comparative results of NHE for $VTR = 1.0e-014$ for different strategies of DE with FSDE

Function	D	VTR	DE			FSDE	Significance
			DE/best/1	DE/rand/1	DE/best-to-rand/1		
Sphere	75	1.e-014	2,510,000	3,139,000	3,030,000	296,000	5,000,000
	100	1.e-014	2,500,000	3,190,000	3,010,000	2,960,000	5,000,000
Beale	75	1.e-014	42,000	71,000	52,000	75,000	124,000
	100	1.e-014	41,000	87,000	64,000	77,000	117,000
Booth	75	1.e-014	44,000	84,000	56,000	75,000	115,000
	100	1.e-014	44,000	87,000	57,000	74,000	119,000
Schwefel	75	1.e-014	9000	15,000	23,000	5100	7000
	100	1.e-014	6000	13,000	23,000	6000	6000
Michlewicz	75	1.e-014	1000	1000	1000	1000	1000
	100	1.e-014	1000	1000	1000	1000	1000
Schaffer N.2	75	1.e-014	61,000	192,000	122,000	132,000	224,000
	100	1.e-014	62,000	139,000	120,000	135,000	225,000
Schaffer N.4	75	1.e-014	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000
	100	1.e-014	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000
HimmelBlau	75	1.e-014	45,000	99,000	92,000	81,000	317,000
	100	1.e-014	46,000	99,000	62,000	79,000	296,000
Bird	75	1.e-014	1000	1000	1000	1000	1000
	100	1.e-014	1000	1000	1000	1000	NA

(continued)

Table 5.2 (continued)

Function	D	VTR	DE DE/best/1	DE/rand/1	DE/best-to- rand/1	DE/best/2	DE/rand/2	FSDE	Significance
Extended cube	75	1.e-014	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000	NA
	100	1.e-014	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000	NA
Ackley	75	1.e-014	428,000	5,000,000	519,000	5,000,000	5,000,000	207,000	+
	100	1.e-014	4,270,000	5,000,000	514,000	5,000,000	5,000,000	205,000	-
Gold	75	1.e-014	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000	NA
	100	1.e-014	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000	NA
Griewank	75	1.e-014	5,000,000	3,578,000	3,380,000	5,000,000	5,000,000	1,299,000	+
	100	1.e-014	5,000,000	3,520,000	4,007,000	3,766,000	5,000,000	1,345,000	+
Rastrigin	75	1.e-014	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000	2,350,000	+
	100	1.e-014	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000	5,000,000	NA
Rosenbrock	75	1.e-014	663,000	870,000	778,000	4,608,000	5,000,000	1,743,000	-
	100	1.e-014	4,080,000	5,000,000	917,000	5,000,000	4,650,000	1,212,000	-

Table 5.3 Comparative results of CPU time for VTR = 1.0e-016 for different strategies of DE with FSDE

Function	D	VTR	DE			FSDE	Significance
			DE/best/1	DE/rand/1	DE/best-to-rand/1		
Sphere	75	1.e-014	14.64	58.66	11.07	55.22	91.76
	100	1.e-014	9.62	66.13	11.53	63.21	103.56
Beale	75	1.e-014	7.65	9.68	10.56	8.4	6.8
	100	1.e-014	10.2	14.1	6.8	8.83	7.84
Booth	75	1.e-014	7.48	12.4	42.1	7.91	10.82
	100	1.e-014	6.62	8.78	6.68	5.55	6.34
Schwefel	75	1.e-014	7.78	7.37	10.8	13.3	7.87
	100	1.e-014	6.49	6.22	5.86	5.17	5.27
Michalewicz	75	1.e-014	18.16	3.9	11.3	9.1	5.2
	100	1.e-014	8.35	10.85	8.1	9.04	6.7
Schaffer N.2	75	1.e-014	14.28	15.75	18.3	21.4	9.78
	100	1.e-014	14.6	13.35	8.35	9.5	9.1
Schaffer N.4	75	1.e-014	130.2	134.6	145.2	132.2	165.2
	100	1.e-014	132.23	124.96	134.5	140.2	121.3
HimmelBlau	75	1.e-014	10.2	7.67	7.33	7.1	15.2
	100	1.e-014	7.51	6.67	5.82	6.4	11.23
Bird	75	1.e-014	8.4	4.5	5.6	6.7	4.3
	100	1.e-014	8.5	5.05	3.9	4.8	4.3

(continued)

Table 5.3 (continued)

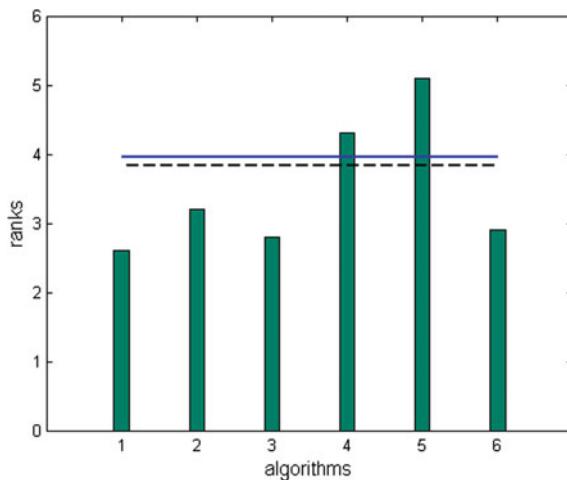
Function	D	VTR	DE DE/best/1	DE/rand/1	DE/best-to- rand/1	DE/best/2	DE/rand/2	FSDE	Significance
Extended cube	75	1.e-014	163.03	153.2	156.3	150.7	148.38	153.3	-
	100	1.e-014	176.37	169.1	176.33	182.1	178.7	167.2	+
Ackley	75	1.e-014	12.6	17.37	18.07	108.7	103.2	48.6	-
	100	1.e-014	14.65	102.3	18.13	133.2	135.3	60.21	-
Gold	75	1.e-014	97.9	98.7	99.3	98.6	104.2	101.2	-
	100	1.e-014	104.7	107.3	110.3	107.89	110.2	105.1	-
Griewank	75	1.e-014	140.3	81.7	35.4	109.3	111.4	31.2	+
	100	1.e-014	116.2	78.34	90.23	89.67	113.2	46.37	+
Rastrigin	75	1.e-014	105.8	96.2	97.6	100.8	89.6	54.3	+
	100	1.e-014	103.84	106.23	105.2	106.12	104.2	103.91	-
Rosenbrock	75	1.e-014	42.65	109.73	36.2	85.58	157.77	34.3	+
	100	1.e-014	136.6	104.1	56.2	99.01	103.2	40.06	+

Table 5.4 Test statistics using Friedman's test

N	25
Chi sq	23.6
Df	5
Asymptotic Significance	0.0003

Table 5.5 Ranks of the diverse strategies

Strategies	Mean rank on best value	Mean rank on NFE
DE/best/1	2.6	2.7
De/rand/1	3.2	4.0
DE/best-to-rand/1	2.8	3.3
De/best/2	4.3	3.6
DE/rand/2	5.1	4.7
FSDE	2.9	2.5

**Fig. 5.1** Bonferroni Dunn chart for best value

test justify the efficiency of FSDE strategy. Based on the ranks obtained, a graphical representation of the results is shown in Fig. 5.1. The x axis of the graph represents the various strategies used and the y axis shows the ranks obtained.

5.4 FSDE in Clustering

This section discusses the implementation of the mutation variant of DE in clustering using the k-means algorithm. Each record in a dataset is handled as random sample of population under consideration. Now, consider to cluster these datasets to k random groups. Partitions of the data set are carried out on the basis of certain objective functions. This is a feature that opts to an optimization problem to minimize or maximize the function from a set of given available alternatives. This function determines how well the chosen solution performs. The fitness of each solution is performed by evaluating the distance between the centroid and the entity point which is defined as:

$$Fitness(C) = \sum_{j=1}^k \sum_{i=1}^n \|x_i^j - c^j\|^2 \quad (5.2)$$

where x_i^j is the entity point, c^j is the centroid and $\|x_i^j - c^j\|$ gives the distance between the centroid and the entity point. Then for each k groups, centroid is selected arbitrarily. Using Euclidean distance method, distance is calculated for each data from its corresponding centroid in the group. K-means algorithm terminates if there is no change in centroid allocated. Result of the k-means algorithm is used as one of the elements of DE algorithm. FSDE algorithm performs the proposed mutation and crossover function. If the resultant value obtained has better cost function, then the resultant value replaces the least fit value in the population. The FSDE algorithm terminates if the maximum iteration is exceeded. Figure 5.2 shows the flow chart for the clustering technique using the variant of DE.

5.5 Experimental Results on Clustering

Here, the experiment is conducted on fifteen standard datasets with numeric data to compare the performance in data clustering by incorporating k-means algorithm with Genetic Algorithm (GA), Particle Swarm Optimization (PSO), classical DE and FSDE. These clustering results were compared with the results of simple k-means algorithm. The clustering algorithms were implemented on i7 core processor, 64 bit operating system with 12 GB RAM using MATLABr2008b and a comparative result was obtained for the various algorithms. The corresponding cluster graph and curve graph for each dataset was obtained. The cluster quality of the clusters obtained is compared. Fifteen real time datasets from UCI repository of Machine learning database is used. The datasets used are described below:

- Fisher Iris dataset ($n = 150$, $d = 4$, $k = 3$): This is a standard multivariate dataset with 150 inputs for 3 different flower types: sentosa, virginica and versicolour. Here 4 different features of a flower is measured: type, petal width, sepal width and sepal length.

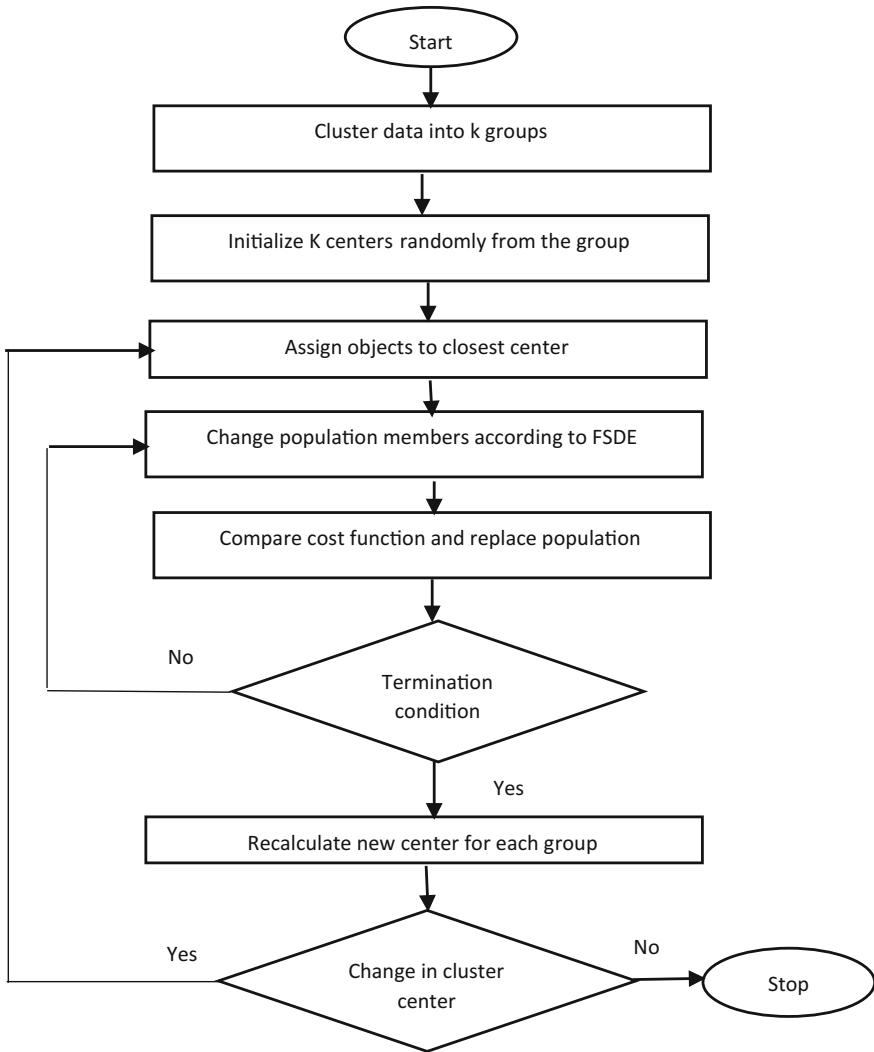


Fig. 5.2 Flow chart for FSDE in clustering

- Morse dataset ($n = 1296$, $d = 5$, $k = 4$): This dataset consist of 36 rows and 36 columns representing the morse code for letters from A to Z and numbers from 0 to 9. Each letter or number is represented with a dash or dot. Morse code consists of 5 elements: short mark, longer mark, intra character gap, short gap and medium gap.
- Hogg dataset ($n = 30$, $d = 6$, $k = 4$): This dataset is a lab based report of bacteria counts in different shipment of milk. Here, 6 sets of bacteria count are taken from 5 different shipment of milk.

- Weather dataset ($n = 60$, $d = 5$, $k = 4$): This is an unsupervised dataset obtained from lab test. It describes main characteristics of weather database for 5 different attributes: outlook, temperature, humidity, windy and play.
- Stock dataset ($n = 950$, $d = 10$, $k = 4$): This is a real dataset obtained from simulated stock returns. Here the stock returns for 10 different companies are being considered in this dataset.
- Temperature dataset ($n = 500$, $d = 19$, $k = 2$): This data set consists of a set of SW-NIR spectra measured on an HP diode array spectrometer. The data set is used to trace the influence of the temperature on vibrational spectra and the consequences for the predictive ability of multivariate calibration models. The dataset has been used by several people to test calibration model building. The spectra 19 mixtures of ethanol, water and isopropanol and the spectra of the pure compounds are recorded in a 1 cm cuvette at different temperatures (30, 40, 50, 60, 70 °C). The temperature of the samples is controlled (~0.2° variation). The measurements were set-up according to a mixture design. For measurements at a temperature of 50 °C and higher, the experimental set-up was changed so that the actual temperature in the cuvette could be recorded.
- Glucosinolate dataset ($n = 660$, $d = 11$, $k = 2$): The glucosinolate levels for 11 different glucosinolates were measured at 1, 3, 7 and 14 days after treatment. This measurement was destructive so that for each time-point different plants were analysed. Each measurement was repeated 5 times. Besides the two treatments there was also a control group that received no JA treatment. The data presented here includes the measurements of eleven different glucosinolates 1, 3, 7 and 14 days after treatment of the root induced, shoot induced and control group.
- Moore dataset ($n = 120$, $d = 5$, $k = 2$): This dataset is a lab based report of the biochemical demand on five predictors. The dataset has five predictor values and one response.
- Ovarian cancer ($n = 4000$, $d = 95$, $k = 2$): This dataset gives the observation on 4000 patients. The ovarian cancer dataset consists of high-resolution ovarian cancer data generated using the WCX2 protein array. The sample set includes 95 controls and 4000 ovarian cancers.
- Gas ($n = 20$, $d = 2$, $k = 2$): This dataset gives the Gasoline prices around the state of Massachusetts in 1993. The file contains two random samples of prices for a gallon of gas around the state of Massachusetts in 1993. The first sample, price1, contains 20 random observations around the state on a single day in January. The second sample, price2, contains 20 random observations around the state one month later.
- Cities ($n = 2961$, $d = 9$, $k = 2$): This dataset gives the quality of life ratings for U.S. metropolitan areas. The data includes ratings for 9 different indicators of the quality of life in 329 U.S. cities. These are climate, housing, health, crime, transportation, education, arts, recreation, and economics. For each category, a higher rating is better.
- K means data ($n = 560$, $d = 4$, $k = 3$): It is a four dimensional grouped data provided for clustering with one field.
- Popcorn ($n = 18$, $d = 3$, $k = 2$): It is a dataset showing the popcorn yield by popper type and brand. The columns of the matrix popcorn are brands (Gourmet, National,

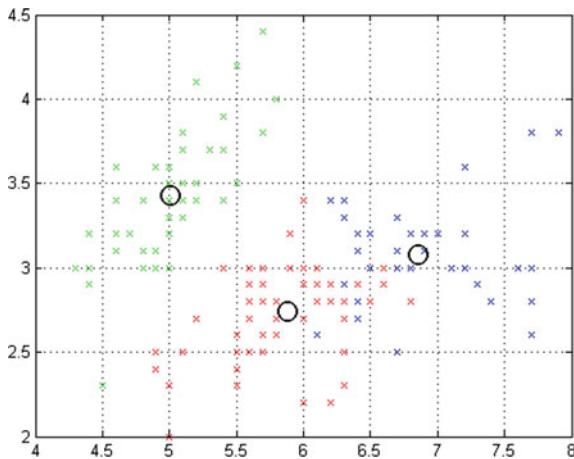


Fig. 5.3 Clusters after applying genetic algorithm

and Generic). The rows are popper type (Oil and Air.) The study popped a batch of each brand three times with each popper. The values are the yield in cups of popped popcorn.

- Exam grades ($n = 649$, $d = 33$, $k = 3$): This data approach student achievement in secondary education of two Portuguese schools. The data attributes include student grades, demographic, social and school related features and it was collected by using school reports and questionnaires. Two datasets are provided regarding the performance in two distinct subjects: Mathematics (mat) and Portuguese language (por).
- Parts ($n = 144$, $d = 4$, $k = 2$): This dataset shows the measurements on newly machined parts taken at 1 h intervals for 36 h. Each row of the runout matrix contains the measurements for four parts chosen at random. The values indicate, in thousandths of an inch, the amount the part radius differs from the target radius.

These datasets are used as input for clustering and results are obtained for various algorithms under consideration. The cluster graph and curve graph for the iris data set has been given below. Figures 5.3, 5.4, 5.5 and 5.6 shows the clusters obtained after performing genetic algorithm, PSO, DE and FSDE respectively for the iris dataset. The x-axis shows the position of the data and y-axis shows the petal width of the iris data set. Figures 5.7, 5.8, 5.9 and 5.10 shows the curve graphs obtained during clustering using genetic algorithm, PSO, DE and FSDE for the iris dataset. In these graphs, the x-axis shows the number of iterations and y-axis shows the best cost obtained at each iteration.

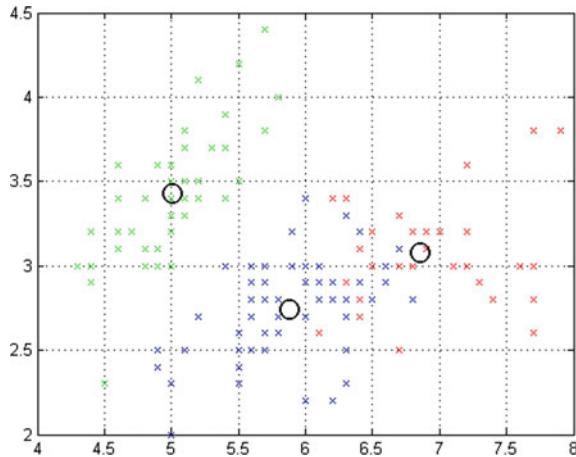


Fig. 5.4 Clusters after applying PSO

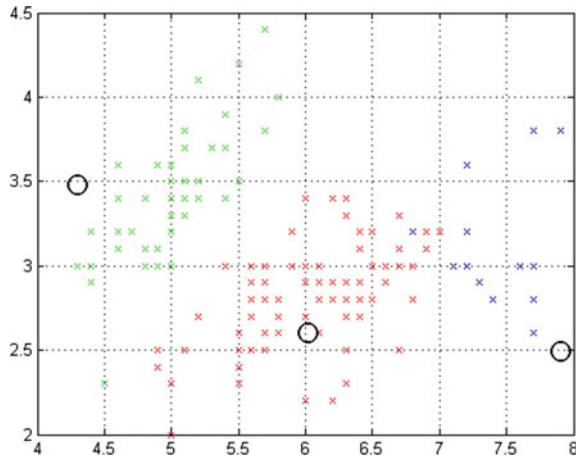


Fig. 5.5 Clusters after applying classical DE

5.5.1 Cluster Quality

Quality of the clusters obtained can be compared based on the following parameters:

Intra Cluster Distance

Intra cluster distance calculates the distance between the members within the cluster. It is calculated as the sum of squares of distance from the members of each cluster to its centroid. Minimum intra cluster distance gives good clusters. The formula for intra cluster distance is given as:

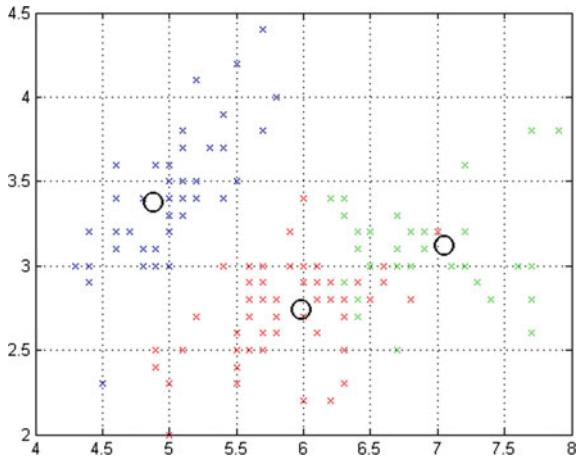


Fig. 5.6 Clusters after applying FSDE algorithm

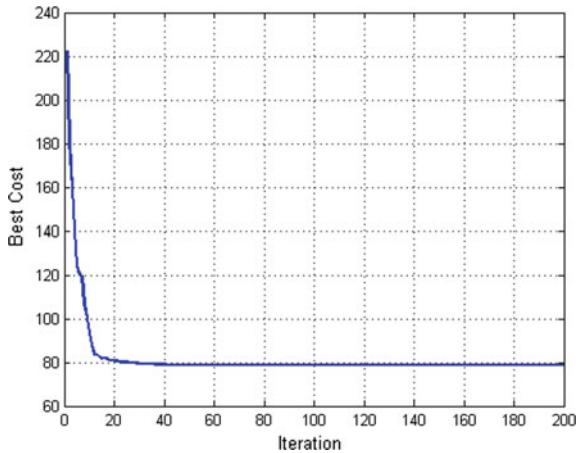


Fig. 5.7 Curve graph for genetic algorithm

$$intra = \sum_{j=1}^k \|x_i^j - c_j\|^2 \quad (5.3)$$

where $\|x_i^j - c_j\|$ is the distance between particles and centroid. Lower the intra cluster distance value, better the cluster formed. The comparative results obtained for mean of intra cluster distance is given in Table 5.6.

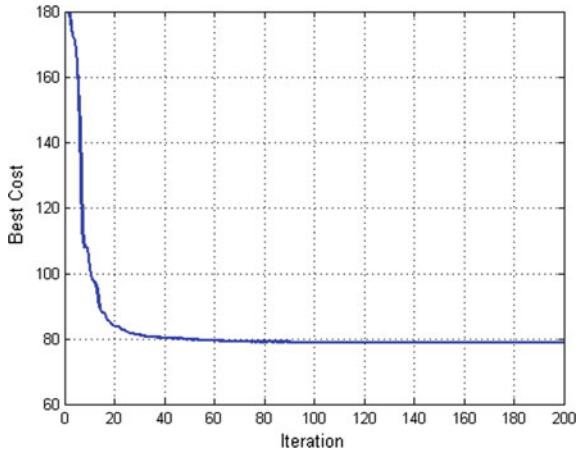


Fig. 5.8 Curve graph for PSO

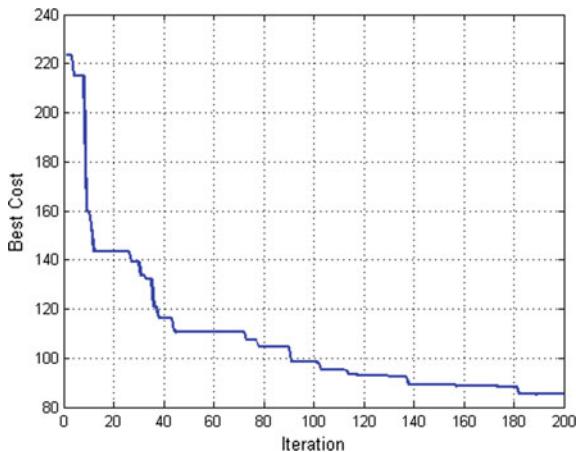


Fig. 5.9 Curve graph for FSDE

Inter Cluster Distance

Inter cluster distance calculates the distance of all the pairs of centroid of different clusters. It is the sum of squares of distance between each cluster centroid. The formula is given as:

$$inter = \min(\|c_i - c_j\|)^2 \quad (5.4)$$

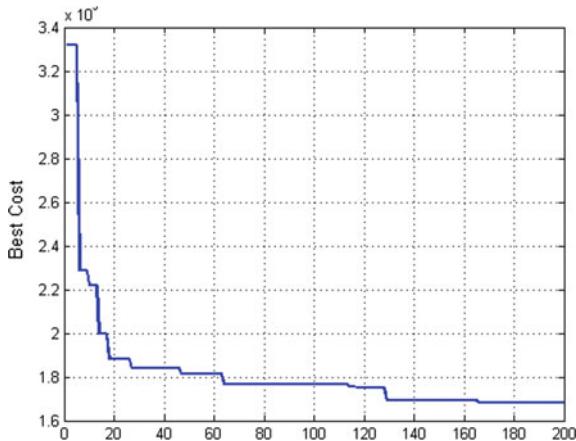


Fig. 5.10 Curve graph for classical DE

here c_i and c_j are the centroids of cluster i and j. Maximum the inter cluster distance, better shall be the cluster formed. The comparative results obtained for mean of inter cluster distance of the various algorithms are given in Table 5.7.

Quantization Error

Vector quantization divides large set of data into clusters having almost same number of points closest to the cluster. Goal of vector quantization is to reduce the average quantization error. The formula for quantization error Q_e is given as:

$$Q_e = \sum_{j=1}^k \left[\sum_{i=1}^n \|x_i^j - c^j\|^2 / N_j \right] / k \quad (5.5)$$

where,

c^j is centroid of cluster j

N_j is number of particles of cluster j

$\|x_i^j - c^j\|$ is the distance between particles and centroid

Lower the quantization error, better is the cluster formed.

The result obtained for quantization error is tabulated in Table 5.8.

Execution Time

Total time taken for the execution of task is called execution time. Lower the execution time, better is the cluster formed. Execution time for the various algorithms is shown in Table 5.9.

Table 5.6 Mean Intra cluster distance for K-means, GA, PSO, DE and FSDE

Data sets	Mean intra cluster distance				
	K-means	GA	PSO	Classical DE	FSDE
Morse (k = 4)	16.01	16.17	19.467	18.803	16.07
Iris (k = 3)	43.7	48.24	45.78	22.24	20.34
Hogg (k = 4)	120.1	122.3	146.66	107.08	99.04
Weather (k = 4)	1.6×10^5	1.7×10^5	1.8×10^5	1.7×10^5	1.6×10^5
Stock returns (k = 4)	2011	2134	2345.7	2256.8	1693.03
Temperature (k = 2)	4.2	3.1	2.9	2.82	2.01
Glucosinolate (k = 2)	240.8	232.4	228.6	229.2	221.3
Moore (k = 2)	3.45×10^7	3.42×10^7	3.41×10^7	372×10^7	3.13×10^7
Ovariancancer (k = 2)	92,800	91,189	91,200	90,306.17	90,010.4
Gas (k = 2)	32.4	31.2	30.3	27.12	25.12
Cities (k = 2)	6.67×10^{10}	6.61×10^{10}	6.66×10^{10}	6.54×10^{10}	6.51×10^{10}
Kmeans data (k = 3)	3812	3729	3716	3720.6	3715.3
Popcorn (k = 2)	1.2	1.15	1.12	1.02	1.01
Examgrades (k = 3)	1678.1	16,558.2	16,550.6	16,456.01	16,450
Parts (k = 2)	2.34	2.02	1.78	1.565	1.45

5.5.2 Validation Index

There are various quantitative evaluation techniques available to test the cluster quality and these are known as validation index. It is used as a tool by researchers to test the cluster result. Internal quality compares different set of clusters without any references to the external knowledge. A good clustering technique has high within cluster similarity and low inter cluster similarity. Here, two validation indexes are considered namely Davies Bouldin (DB) index and Calinski Harabasz (CH) index.

Davies Bouldin (DB) Index

It is a matrix for evaluating the cluster algorithm. It is a function of ratio of sum of intra-distances to inter distances [6]. If $R_{i,j}$ be the measure of clustering scheme, $M_{i,j}$ is the separation between i and j cluster and S_i is the within cluster scatter for cluster I, then the DB index is defined as the ratio of S_i and $M_{i,j}$ which follows the following properties:

1. $R_{i,j} \geq 0$.
2. $R_{i,j} = R_{j,i}$.
3. When $S_i \geq S_j$ and $M_{i,j} = M_{i,k}$ then $R_{i,j} > R_{i,k}$.
4. When $S_i = S_j$ and $M_{i,j} \leq M_{i,k}$ then $R_{i,j} > R_{i,k}$.

Here, lower the value of DB index, better the separation and closeness of the data inside the cluster. The formula for DB index is given as:

Table 5.7 Mean Inter cluster distance for K-means, GA, PSO, DE and FSDE

Data sets	Mean inter cluster distance				
	K-means	GA	PSO	Classical DE	FSDE
Morse (k = 4)	230.12	224.14	159.12	241.9	277.55
Iris (k = 3)	3851.3	4015.72	4037.6	4144.86	5781.3
Hogg (k = 4)	4245.2	4542.57	3158.7	4527.9	4549.71
Weather (k = 4)	3.4×10^7	3.6×10^7	3.0×10^7	3.6×10^7	3.6×10^7
Stock returns (k = 4)	12,876.22	11,660.32	18,330.7	13,206	26,889
Temperature (k = 2)	21.3	20.45	23.56	38.38	40.2
Glucosinolate (k = 2)	1300.2	1289.7	1225.6	1306.6	1320.4
Moore (k = 2)	3.78×10^8	3.8×10^8	4.03×10^8	4.23×10^8	4.25×10^8
Ovariancancer (k = 2)	663,425.4	643,251.9	645,678.12	675,868.25	685,100.2
Gas (k = 2)	2314.5	2435.4	2432.8	2652.3	2665.3
Cities (k = 2)	4.12×10^{10}	4.45×10^{10}	4.67×10^{10}	4.99×10^{10}	5.1×10^{10}
Kmeansdata (k = 3)	16,453.4	14,325.6	14,001.6	15,789.9	16,678.2
Popcorn (k = 2)	30.2	31.6	33.8	36.3	37.2
Examgrades (k = 3)	15,432.7	160,034.4	163,212.4	160,364.8	16,432.1
Parts (k = 2)	17.67	15.23	15.05	18.95	19.5

Table 5.8 Quantization Error for K-means, GA, PSO, DE and FSDE

Data sets	Quantization error				
	K-means	GA	PSO	Classical DE	FSDE
Morse (k = 4)	4.03	4.04	4.86	4.7	4.01
Iris (k = 3)	14.03	16.08	15.25	7.41	6.78
Hogg (k = 4)	28.7	30.57	36.65	26.7	24.76
Weather (k = 4)	42,567.2	42,625.15	45,300.17	42,600.2	42,443.2
Stock returns (k = 4)	530.3	533.5	586.4	564.2	423.2
Temperature (k = 2)	1.35	1.54	1.87	1.41	1.21
Glucosinolate (k = 2)	115.9	117.3	115.8	114.6	113.2
Moore (k = 2)	1.58×10^7	1.61×10^7	1.78×10^7	1.68×10^7	1.54×10^7
Ovariancancer (k = 2)	22,789.5	23,112	22,890	22,576.5	21,100
Gas (k = 2)	6.67	6.61	6.54	6.78	6.12
Cities (k = 2)	3.25×10^7	3.30×10^7	3.33×10^7	3.27×10^7	3.15×10^7
Kmeans data (k = 3)	925.3	928.6	927.6	930.1	910.7
Popcorn (k = 2)	0.37	0.38	0.43	0.33	0.23
Examgrades (k = 3)	5527	5531	5520	5485.3	5411.2
Parts (k = 2)	0.50	0.61	0.55	0.52	0.48

Table 5.9 Execution time for K-means, GA, PSO, DE and FSDE

Data sets	Execution time				
	K-means	GA	PSO	Classical DE	FSDE
Morse (k = 4)	15.8	25.1	24.3	17.3	15.6
Iris (k = 3)	15.4	25.018	15.34	25.03	14.13
Hogg (k = 4)	24.1	25.54	26.34	26.77	23.7
Weather (k = 4)	46.3	55.34	56.67	60.65	45.7
Stock returns (k = 4)	14.32	16.45	15.56	36.53	14.12
Temperature (k = 2)	130.4	128.4	132.2	127.1	125.4
Glucosinolate (k = 2)	8.03	8.12	8.8	8.9	8.01
Moore (k = 2)	8.14	8.87	8.34	9.99	8.12
Ovariancancer (k = 2)	104.5	105.2	105.6	103.8	101.2
Gas (k = 2)	8.5	9.1	9.4	8.82	8.25
Cities (k = 2)	37.1	37.6	36.7	37.24	36.9
Kmeans data (k = 3)	111.7	113.4	114.3	112.4	111.5
Popcorn (k = 2)	7.2	6.7	7.0	6.52	6.45
Examgrades (k = 3)	11.3	10.67	12.3	11.23	11.21
Parts (k = 2)	7.6	7.1	6.5	6.7	6.6

$$DB = \frac{1}{k} \sum_{i=1}^k \max_{i \neq j} \left\{ \frac{d(x_i) - d(x_j)}{d(c_i, c_j)} \right\} \quad (5.6)$$

where,

k is number of clusters

i, j are cluster labels

$d(x_i)$ and $d(x_j)$ are all samples in cluster i and j to respective cluster centroids

$d((c_i, c_j))$ is the distance between these centroids.

The comparative results obtained for the DB index for the various algorithms are tabulated in Table 5.10.

Calinski Harabasz (CH) Index

It is another method for calculating cluster quality. It is used to evaluate the optimal number of clusters [7]. Higher the value of CH index, the better the cluster that is formed. The formula for CH index is given as:

$$CH = \frac{\text{trace}(S_B)}{\text{trace}(S_W)} \cdot \frac{n_p - 1}{n_p - k} \quad (5.7)$$

where S_B is between cluster scatter matrix, S_W is the internal scatter matrix, n_p is the number of clustered samples and k is the number of clusters. The comparative results obtained for the CH index is given in Table 5.11.

Table 5.10 DB Index for K-means, GA, PSO, DE and FSDE

Data sets	DB index				
	K-means	GA	PSO	Classical DE	FSDE
Morse (k = 4)	1.20	1.22	1.19	1.23	1.12
Iris (k = 3)	0.63	0.78	0.68	0.64	0.61
Hogg (k = 4)	0.432	0.43	0.45	0.41	0.40
Weather (k = 4)	0.41	0.45	0.38	0.35	0.34
Stock returns (k = 4)	3.24	4.34	4.23	4.54	2.16
Temperature (k = 2)	1.23	1.43	1.20	1.35	1.32
Glucosinolate (k = 2)	3.7	3.12	3.2	3.4	3.1
Moore (k = 2)	0.7	0.56	0.67	0.8	0.75
Ovariancancer (k = 2)	1.46	1.55	1.65	1.45	1.35
Gas (k = 2)	1.21	1.12	1.10	1.06	1.04
Cities (k = 2)	7.1	7.54	7.6	7.5	7.43
Kmeans data (k = 3)	0.81	0.43	0.76	0.89	0.85
Popcorn (k = 2)	1.20	1.34	1.32	1.21	1.19
Examgrades (k = 3)	4.5	5.1	5.5	4.3	4.1
Parts (k = 2)	1.89	2.01	2.4	2.5	2.1

Table 5.11 CH index for K-means, GA, PSO, DE and FSDE

Data sets	CH index				
	K-means	GA	PSO	Classical DE	FSDE
Morse (k = 4)	0.78	0.67	0.67	0.64	1.065
Iris (k = 3)	0.32	0.22	0.301	0.21	0.34
Hogg (k = 4)	0.17	0.168	0.23	0.20	0.26
Weather (k = 4)	0.144	0.146	0.133	0.143	0.148
Stock returns (k = 4)	2.12	2.0484	2.01	1.075	2.48
Temperature (k = 2)	1.3	0.89	0.95	0.71	1.2
Glucosinolate (k = 2)	1.89	2.04	2.13	2.21	2.4
Moore (k = 2)	1.2	0.23	0.45	0.536	0.63
Ovariancancer (k = 2)	0.32	0.55	0.43	0.69	0.71
Gas (k = 2)	0.4	0.38	0.24	0.53	0.65
Cities (k = 2)	5.6	4.3	4.44	5.61	5.7
Kmeans data (k = 3)	0.6	0.47	0.45	0.56	0.62
Popcorn (k = 2)	0.12	0.43	0.32	0.23	0.232
Examgrades (k = 3)	2.21	1.45	2.78	2.95	3.4
Parts (k = 2)	1.12	1.5	1.34	1.8	1.85

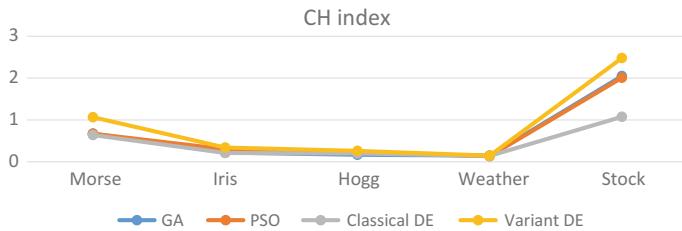


Fig. 5.11 Curve for CH index



Fig. 5.12 Curve for execution time

5.5.3 Graphical Representation

The above tabulated values of Cluster Quality and Validation index has been depicted graphically in Figs. 5.11 and 5.12. The graphs below show the performance curve for execution time in cluster quality and CH index in validation index. The x-axis represents the different datasets used and y-axis represents the value obtained. The line graph shows that the values obtained for the FSDE is significantly better than the values obtained from classical DE approach. The values have been recorded for five different datasets.

5.6 Summary

In this work, a variant of the mutation strategy of Differential Evolution named FSDE is proposed and is applied to k-means technique of data clustering. The results obtained show that the variant created is more efficient than the classical schemes of DE and the results are significantly good for clustering application. This method is used for clustering only numeric dataset. Further extension of the work can be

done in the field of image and text. Also, FSDE technique is applied only to k-means technique of clustering. Further extension of the work can be done in applying FSDE to other techniques of clustering like hierarchical agglomerative method, DBSCAN method etc.

References

1. Jain, A.K.: Data clustering: 50 years beyond K-means. *Pattern Recogn. Lett.* **31**(8), 651–666 (2010)
2. Bezdek J.C., Ehrlich R., Full W.:FCM: the fuzzy c-means clustering algorithm. *Comput. Geosci.* **10**(2–3), 191–203 (1984)
3. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, no. 14. pp. 281–297 (1967)
4. Coello, C.A.C., David, A.V.V., Gary, B.L.: Evolutionary algorithms for solving multi-objective problems, vol. 242. Kluwer Academic, New York (2002)
5. Storn, R., Price, K.: Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J. Global Optim.* **11**(4), 341–359 (1997)
6. Davies, D.L., Bouldin, D.W.: A cluster separation measure. *IEEE Trans. Pattern Anal. Mach. Intell.* **2**, 224–227 (1979)
7. Caliński, T., Harabasz, J.: A dendrite method for cluster analysis. *Commun. Stat. Theor. Meth.* **3**(1), 1–27 (1974)

Chapter 6

Reconstructed Mutation Strategy for Differential Evolution Algorithm



This chapter introduces a new variant of Differential Evolution named as RDE—Reconstructed Differential Evolution. RDE differs from classical DE in its mutation strategy. To improve efficiency and robustness, RDE strategy uses three control parameters unlike FSDE and ReDE. This variant is then applied on clustering of complex weather images.

6.1 Introduction

Weather surveillance radar is used to locate the motion, movement and type of precipitation. The weather radar images provide a map view of reflected particles for specific area around the radar. Generally, different color codes are used to represent the varying degree of precipitation around that area. With the help of these radar images, the climatic conditions of a region can be predicted. Weather describes the condition of our atmosphere during a specific period of time and climate represent a composite of day to day weather over longer period of time. Climatology attempts to analyze and explain the impact of climate so that the society can plan accordingly. These climatology analysis are often done on radar images representing various climatic conditions. These images do contain varying scale of severity for any specific climatic parameter of study. The climatologists often find it convenient to analyze climatic conditions, if tools are available, to segment the weather images based on the severity scale which is represented by different colors. Segmentation of the weather radar image is also used for automated analysis of weather condition. Satellite images with high resolution have become a good source of information for weather forecasting. So there is a need for efficient extraction and processing of useful data. By splitting the images according to color code, the specific hazard type can be viewed and depending on the intensity, the corresponding alerts can be given.

Thresholding is a basic technique of image segmentation where the image is converted to foreground and background images. Thresholding on colored images are done by manipulating the color components based on color spaces. Problems like thresholding are classified as unsupervised learning and such issues can be solved using evolutionary algorithm. Evolutionary algorithm being a portion of evolutionary computing uses the biological methods of reproduction, recombination, selection and mutation. In 1997, Storn and Price introduced the DE algorithm that follows the concepts of evolutionary algorithm. DE remains straightforward, stochastic, population based algorithm that help in solving optimization problem. Differential Evolution (DE) approach is used for fast selection of optimal threshold. The efficiency and performance of DE is determined based on the control parameters and the trial vector generation strategy that is being used. Numerous variants of DE are designed by changing these trial vector strategy and control parameters. In DE algorithm, population size, crossover constant and the mutation scale factor F are the three control parameters being used. Numerous work has also been done on fuzzy based image thresholding. Here, in this section, a new variant of DE named Reconstructed Differential Evolution strategy (RDE) is developed in which three different mutation scale factors are used. This strategy is compared with the other classical variants, by tabulation, to confirm better efficiency of RDE. This strategy is used in image thresholding for segmenting the weather images based on specific climatic condition. Based on fuzzy entropy, multilevel image thresholding is then used along with the differential evolution approach to perform the image segmentation. The paper explains the implementation of the variant in multilevel thresholding and the results that were obtained during the study.

6.2 RDE Mutation Strategy

In RDE, three control parameters are taken into consideration unlike traditional DE which takes only one control parameter into consideration. By involving the best solution vector, this strategy coincides faster as compared to the traditional strategies having random vectors only. By changing the mutation strategy of DE algorithm, the efficiency of the classical DE algorithm is improved profoundly. The variables $X_{r1,G}$, $X_{r2,G}$, $X_{r3,G}$ are chosen at random. The parameter F known as amplifying parameter takes a constant value between (0,2). The new parameter N1 takes a varying value which lies between (0,1) and N2 takes the complement of N1.

The proposed strategy is given as:

$$X' = X_{r1,G} + F \cdot (N1 \cdot (X_{best,G} - X_{r2,G}) - N2 \cdot (X_{best,G} - X_{r3,G})) \quad (6.1)$$

The improved mutation strategy is combined with the crossover technique of traditional DE algorithm to form RDE algorithm. Results obtained from RDE is compared with the results obtained from traditional DE techniques.

6.3 Experimental Results and Statistical Analysis

The hybrid algorithm RDE referred above is implemented on i7 core processor, 64 bit operating system with 12 GB RAM using MATLABR2008b to obtain a comparative result to original DE algorithm. Fifteen different functions are considered and the results are calculated by fixing the value to reach and number of iterations. Results are taken by fixing various dimensions. The results are tabulated for comparison with the existing algorithms in Table 6.1.

A comparative analysis is performed and study is done on each technique. By setting the dimension and value-to-reach (VTR), the best value of different function strategies was calculated. The proposed hybrid algorithm provides best value for most of the standard functions. From the results tabulated, it is clearly shown that the RDE strategy gives good results for most of the functions used.

Based on the values from Table 6.1, Friedman test is applied and the results obtained were tabulated. Table 6.2 represents the values obtained from the test and Table 6.3 depicts the rank of the various mutation strategies used based on best value and NFE (Fig. 6.1).

Table 6.3 show that RDE strategy has significant performance in comparison to the existing mutation strategies. The rank obtained on the basis of best value is comparatively better for best value case. These rankings obtained on the basis of Freidman's test justify the efficiency of RDE strategy. Based on the ranks obtained, a graphical representation of the results is shown in Figs. 6.2, 6.3 and 6.4. The x axis of the graph represents the various strategies used and the y axis shows the ranks obtained.

6.4 Multi-level Image Thresholding

Thresholding is the simplest type of image segmentation done for gray scale images to create binary images. It can be applied to color images by designating separate RGB components of the image. The various techniques used for thresholding are: histogram shape, entropy, clustering and attribute based method, local method, spatial method etc. Thresholding sets all pixels that belong to a range of values to 1 (or other user defined value) and the rest to 0. For an 8-bit image, there are 256 threshold intensities possible ranging from 0 to 255.

Depending on number of image segments, thresholding are of two types: bi-level and multi-level. In the case of bi-level thresholding, the image segmentation is done in two different regions. One region holds the pixel having gray value larger than a defined value T and the other region holds gray values lesser than T . The first region is known as object pixels and the other is background pixels. In multi-level thresholding, the foreground and background objects are segregated into non overlapping sets which aids in segmenting the gray scale image into different sections. It is a type of clustering technique in which the gray level samples are grouped into two sections

Table 6.1 Comparative results of Best Value for different strategies of DE with RDE

Function	D	VTR	DE	DE/best/1	DE/rand/1	DE/best-to-rand/1	DE/best/2	DE/rand/2	RDE	Significance
Sphere	50	1.e-015	9.74e-016	6.91e-016	7.54e-016	9.66e-016	7.17e+0	6.62e-016	+	
	25	1.e-015	9.34e-015	9.35e-015	9.539e-015	9.42e-015	6.92e+000	8.46e-014	+	
	25	1.e-014	8.5e-016	9.99e-016	6.2e-016	9.7e-016	2.517e-001	8.26e-015	-	
Beale	50	1.e-015	3.20e-016	2.32e-016	3.71e-016	7.6e-016	7.73e-016	7.67e-016	-	
	25	1.e-015	4.20e-015	7.72e-015	1.125e-015	1.36e-017	7.5e-015	4.4e-015	-	
	25	1.e-014	5.9e-016	2.50e-016	3.8e-016	9.73e-016	3.9e-016	6.7e-015	+	
Booth	50	1.e-015	3.5e-016	2.051e-016	6.07e-016	7.07e-016	8.3e-016	9.21e-015	+	
	25	1.e-015	1.807e-015	7.553e-016	1.95e-015	2.75e-015	6.47e-015	2.6e-015	-	
	25	1.e-014	3.17e-016	5.87e-016	1.70e-016	5.9e-016	3.7-016	5.5e-015	+	
Schwefel	50	1.e-015	-1.82e+003	-2.25e+003	-7.84e+001	-1.381e+003	-1.67e+003	-3.32e+002	NA	
	25	1.e-015	-4.22e+002	-4.8e+002	-1.67e+003	-4.47e+003	-1.5e+003	-1.41e+003	NA	
	25	1.e-014	-5.6e+002	-2.48e+003	-6.4e+002	-4.93e+003	-2.4e+003	-1.7e+003	NA	
Michlewicz	50	1.e-015	-7.64e+00	-7.21e+00	-7.4e+00	-6.96e+00	-6.85e+00	-1.14e+01	NA	
	25	1.e-015	-7.69e+00	-7.64e+00	-6.87e+00	-7.35e+00	-6.98e+00	-6.6e+00	NA	
	25	1.e-014	-1.136e+001	-1.2e+001	-1.21e+001	-1.24e+001	-1.12e+001	-1.1e+001	NA	
Schaffer N.2	50	1.e-015	6.6e-016	8.88e-016	4.43e-016	6.55e-016	8.87e-016	6.66e-015	+	
	25	1.e-015	1.33e-015	1.33e-015	6.66e-016	5.3e-015	1.33e-015	0	-	
	25	1.e-014	6.66e-016	8.88e-016	0	8.88e-016	6.66e-016	1.55e-015	+	
Schaffer N.4	50	1.e-015	3.02e-015	2.9e-001	2.92e-001	2.93e-001	2.89e-001	2.85e-001	+	
	25	1.e-015	2.92e-001	2.92e-001	2.92e-001	2.92e-001	2.92e-001	2.92e-001	NA	
	25	1.e-014	2.92e-001	2.92e-001	2.92e-001	2.92e-001	2.92e-001	2.92e-001	NA	
HimmelBlau	50	1.e-015	1.6e-016	8.02e-016	3.83e-016	9.12e-016	1.46e-016	4.38e-016	-	
	25	1.e-015	4.83e-015	4.42e-015	1.902e-015	3.95e-015	5.14e-015	3.2e-016	-	

(continued)

(continued)

Table 6.1 (continued)

Function	D	VTR	DE	RDE				Significance
				DE/best/1	DE/rand/1	DE/best-to-rand/1	DE/best/2	
Bird	25	1.e-014	6.19e-016	7.18e-017	5.85e-017	7.17e-016	7.67e-016	9.6e-016
	50	1.e-015	-1.035e+002	-1.067e+002	-1.05e+002	-1.03e+002	-1.05e+002	NA
	25	1.e-015	-9.303e+001	-1.04e+002	-1.066e+002	-1.034e+002	-1.04e+002	-1.03e+002
	25	1.e-014	-1.039e+002	-1.006e+002	-1.057e+002	-1.055e+002	-1.008e+002	NA
Extended Cube	50	1.e-015	3.31e-015	4.98e-006	6.1e-008	1.93e-005	2.68e+00	2.04e-006
	25	1.e-015	5.701e-008	5.212e-005	7.1003e-008	1.73e-005	2.92e+009	2.16e-006
	25	1.e-014	1.50e-007	2.94e-010	1.36e-007	9.42e-011	2.609e-014	1.65e-006
	50	1.e-015	7.19e-015	6.40e-012	7.99e-015	3.63e-013	3.09e+00	4.4e-015
Ackley	25	1.e-015	7.99e-015	5.02e-015	7.99e-015	3.59e-015	3.213e+00	4.4e-014
	25	1.e-014	7.99e-015	6.33e-013	7.99e-015	9.32e-014	7.19e-014	-
	50	1.e-015	3.00e+00	3.00e+00	3.00e+00	3.00e+00	3.00e+00	NA
	25	1.e-015	3.00e+00	3.00e+00	3.00e+00	3.00e+00	3.00e+00	NA
Gold	25	1.e-014	3.00e+00	3.00e+00	3.00e+00	3.00e+00	3.00e+00	NA
	50	1.e-015	3.00e+00	3.00e+00	3.00e+00	3.00e+00	3.00e+00	NA
	25	1.e-014	3.00e+00	3.00e+00	3.00e+00	3.00e+00	3.00e+00	NA
	50	1.e-015	9.99e-016	9.99e-016	1.6e-013	6.56e-013	1.07e+00	9.99e-016
Griewank	25	1.e-015	1.477e-002	9.214e-015	7.88e-015	5.07e-009	1.06e+00	6.6e-016
	25	1.e-014	1.23e-002	8.88e-016	7.65e-016	8.881e-016	6.77e-011	9.65e-015
	50	1.e-015	1.79e+001	1.23e+002	7.47e+001	1.28e+002	1.52e+002	8.04e+001
	25	1.e-015	3.61e+001	1.181e+002	8.17e+001	1.727e+002	1.674e+002	6.61e+001
Rosenbrock	25	1.e-014	6.28e+001	9.72e+001	6.57e+001	1.047e+002	1.29e+001	7.17e+001
	50	1.e-015	9.6e-016	1.07e-008	7.88e-016	3.9e-009	1.07e+005	7.48e-016
	25	1.e-015	3.98e+000	1.403e-008	6.9e-015	1.56e-011	7.15e+004	8.52e-016
	25	1.e-014	9.5e-016	2.23e-012	3.9e+00	7.5e+005	3.51e+008	9.7e-014

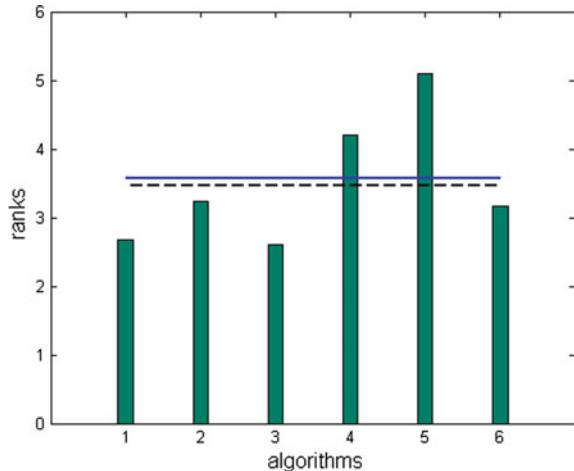
Table 6.2 Test statistics using Friedman's test

N	50
Chi sq	22.71
Df	5
Asymptotic significance	0.0004

Table 6.3 Ranks of the different strategies

Strategies	Mean rank on best value	Mean rank on NFE
DE/best/1	2.67	3.03
De/rand/1	3.23	4.3
DE/best-to-rand/1	2.60	3.0
De/best/2	4.23	3.4
DE/rand/2	5.1	4.2
RDE	3.16	3.1

Fig. 6.1 Bonferroni Dunn bar chart for best value



as background and foreground. In this method, different values are assigned between different ranges of threshold levels. The number of thresholds are stated in advance.

Texture of the input image can be characterized by using a statistical measure of randomness called Entropy.

Entropy of an image can be expressed as:

$$\text{Entropy} = - \sum_j^m p_j * \log_2(p_j) \quad (6.2)$$

Here, m is the number of gray levels and p_j is the probability of occurrence of pixel j . Low entropy has little contrast between pixels and large run of pixels. High

Fig. 6.2 Bonferroni Dunn bar chart for NFE

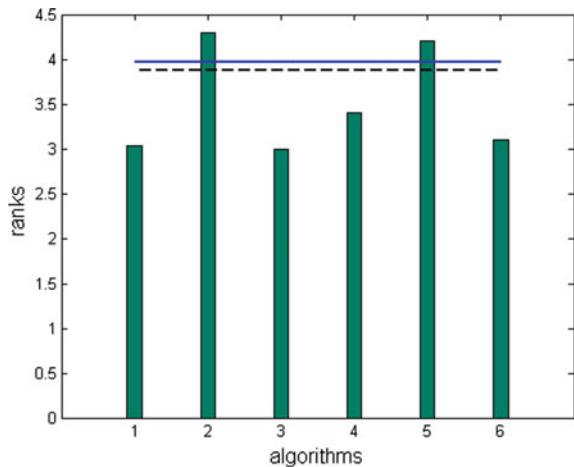
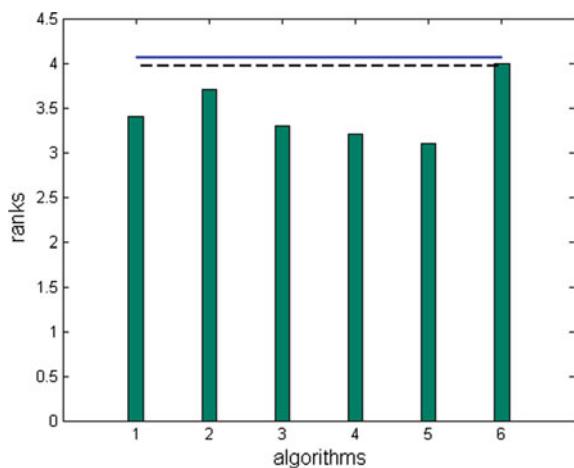


Fig. 6.3 Bonferroni Dunn bar chart for CPU time



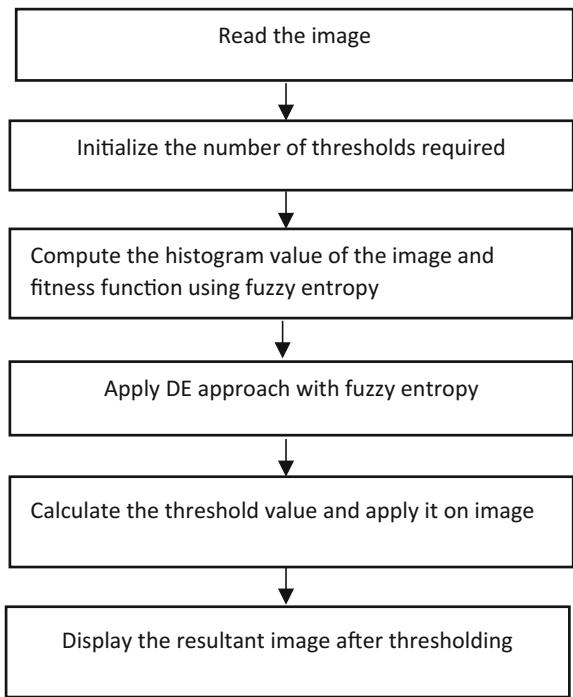
entropy has very high contrast between pixels and hence cannot be compressed as low entropy images.

The measure of uncertainty and entropy over a fuzzy set of an image is referred to as Fuzzy entropy. Consider an image Y of size $I \times J$. Fuzzy entropy of an image is denoted as:

$$\text{Fuzzy entropy} = \frac{1}{IJ \ln 2} \sum_{i=1}^I \sum_{j=1}^J S_n(\mu_Y(y_{ij})) \quad (6.3)$$

where $\mu_Y(y_{ij})$ represents the measure of a property of image like the pixel brightness and S_n symbolize the Shannon's function denoted as:

Fig. 6.4 Block diagram for multi-level thresholding using RDE



$$S_n(\mu_Y(y_{ij})) = -\mu_Y(y_{ij}) \ln \mu_Y(y_{ij}) - (1 - \mu_Y(y_{ij})) \ln (1 - \mu_Y(y_{ij})) \quad (6.4)$$

If the fuzzy entropy is larger, greater will be the information within that fuzzy set. The minimum fuzzy entropy value will be taken as the optimal threshold value. Y is then segmented into background and object pixels using the fuzzy threshold T:

$$y_{ij} = \begin{cases} b_0 & y_{ij} < T \\ b_b & y_{ij} \geq T \end{cases} \quad (6.5)$$

where b_o is the object pixels and b_b is the background pixels; $i = 1, 2, 3 \dots I$ and $j = 1, 2, 3 \dots J$. In this section, fuzzy entropy-based technique is considered for thresholding of the color image.

6.5 Multi-level Thresholding Using RDE Strategy

Here, DE approach is introduced in fuzzy entropy-based thresholding. In this approach, we have used the RDE mutation strategy for DE as given in Eq. 6.1. Using Eq. 6.3, the fuzzy entropy gives the fitness function meant for the DE algorithm. The block diagram for the proposed work is given in Fig. 6.4.

6.6 Test Results on Image Thresholding

The proposed RDE strategy is implemented on DE algorithm and this is combined with the fuzzy entropy approach to perform multi-level image thresholding. This method is applied on few sets of weather radar images. The values for entropy and CPU time is computed and tabulated in Table 6.4. The original image and the images

Table 6.4 Values obtained from images after thresholding

Image	Methods used	Entropy value	CPU time
Image 1	DE based thresholding	0.95	12.43
	RDE based thresholding	0.98	11.7
Image 2	DE based thresholding	0.818	17.02
	RDE based thresholding	0.830	14.3
Image 3	DE based thresholding	0.97	12.5
	RDE based thresholding	0.99	11.63
Image 4	DE based thresholding	0.95	13.4
	RDE based thresholding	0.97	12.35
Image 5	DE based thresholding	0.94	13.2
	RDE based thresholding	0.95	12.01
Image 6	DE based thresholding	0.82	13.4
	RDE based thresholding	0.83	11.2
Image 7	DE based thresholding	0.87	14.3
	RDE based thresholding	0.89	14.1

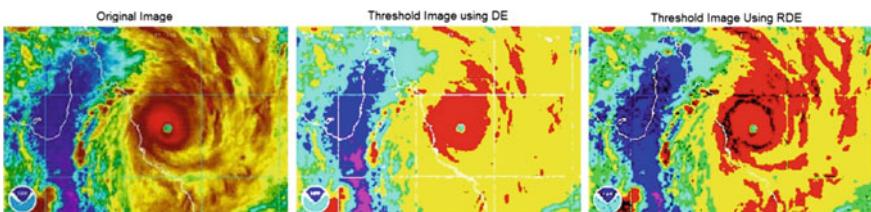


Fig. 6.5 Original and threshold image of sample image 1

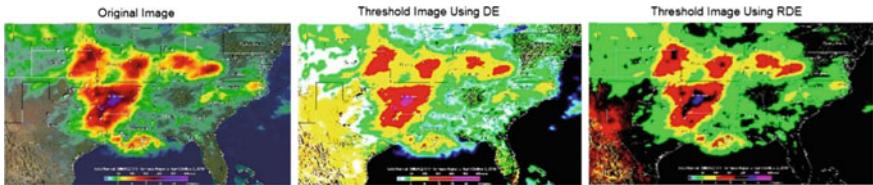


Fig. 6.6 Original and threshold image of sample image 2

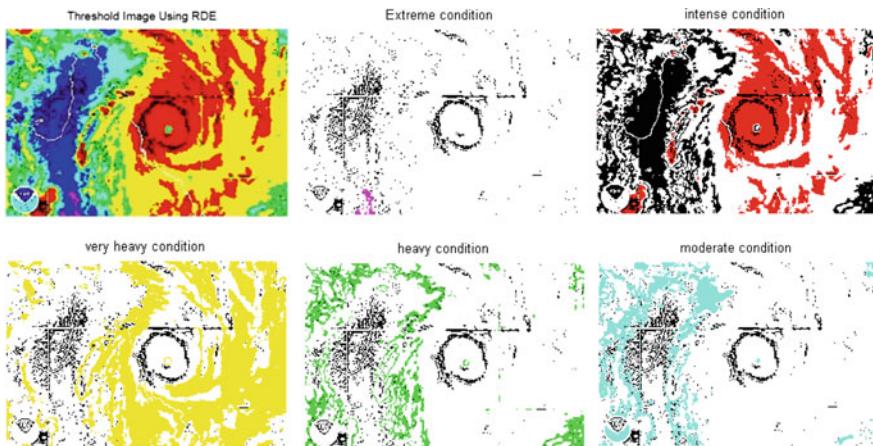


Fig. 6.7 Sample image 1 segmented based on hazard severity

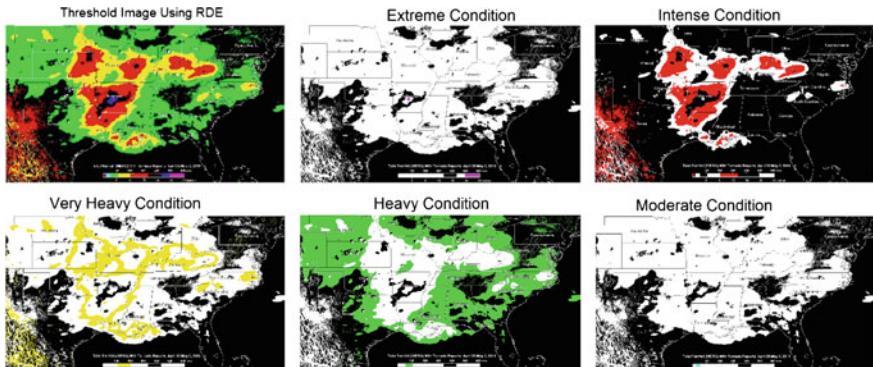


Fig. 6.8 Sample image 2 segmented based on hazard severity

after thresholding using DE and RDE technique are shown in Figs. 6.5 and 6.6. The results obtained demonstrate the proposed RDE strategy gives improved results for multilevel thresholding in comparison to classical DE approach.

The above tabulated results clearly indicate that the values obtained for RDE strategy based thresholding is much better compared to the results obtained using DE

strategy. Samples of weather radar images are taken from various weather websites like www.wunderground.com and the multilevel thresholding was performed on these images. The image after thresholding is segmented based on the specific hazard severity. A sample of rainfall image is shown in image 1 and for snowfall is shown in image 2. The results obtained are given in Figs. 6.5 and 6.6.

Color enhancements are seen in a radar image to help in interpretation. The specific colors are compared with the standard color bar to identify the hazard severity. Different colors on the image depict different intensities of the weather condition. The violet/pink color bar shows the extreme weather condition of snow, rain or hail in an area. The red color bar shows an intense weather condition. The yellow color bar shows a very heavy precipitation level and the green color bar shows heavy condition. Light blue color bar shows a moderate climatic condition. By clustering the images separately according to the specific hazard severity, it is easy to identify and predict the specific weather condition across the plains. The various weather hazards of image 1 and 2 is given in Figs. 6.7 and 6.8.

From the above resultant images, it is easy to identify the regions of extreme and intense weather conditions. In the original image, it is not very easy to see at a glance if a case of extreme condition is occurring on any specific land area. After separating the images based on color bar, the area for extreme and intense conditions are easy to identify. In Fig. 6.7 for example, the image for extreme condition shows only pink on the corresponding effected area whereas, for moderate condition the image easily depicts that no region on the map is having that specific climatic condition. Similarly, the segmented images for various weather images are given in Fig. 6.9.

6.7 Summary

Herein the proposed work, reconstructed mutation strategy was implemented in DE algorithm and results being compared to prevailing mutation strategies. The comparative study shows better results for RDE. RDE was then applied to multilevel thresholding based on fuzzy entropy. The thresholding results were found to be better for RDE strategy in comparison to classical DE approach. The threshold image was further segregated based on the specific hazard level. From this study, it is seen that RDE strategy performs much better in comparison to other strategies. Currently the strategy has been applied for weather forecasting. An extension of the work can be done to apply this technique for medical image processing, land topology, image enhancement and other image processing areas.

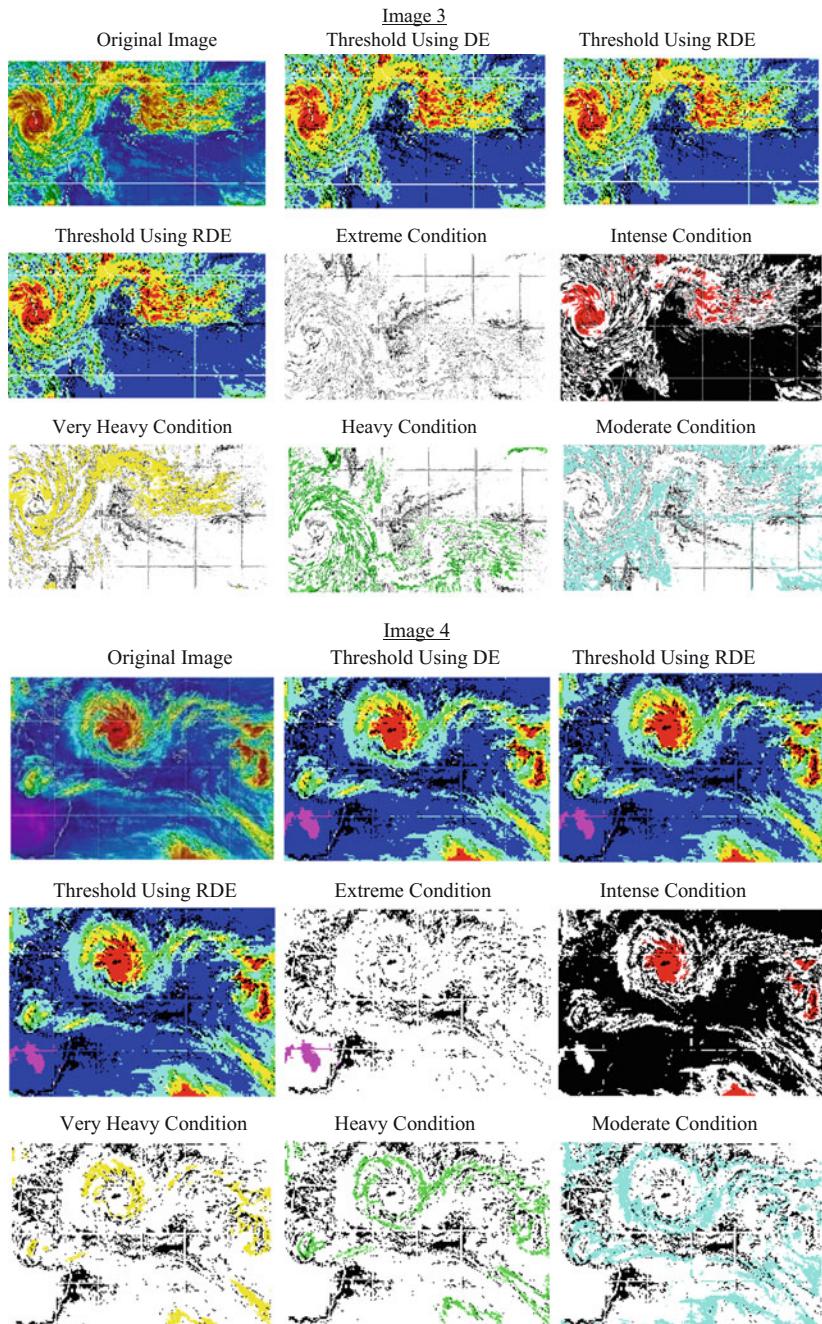


Fig. 6.9 Original images with threshold images using DE and RDE. Images segmented based on weather hazards

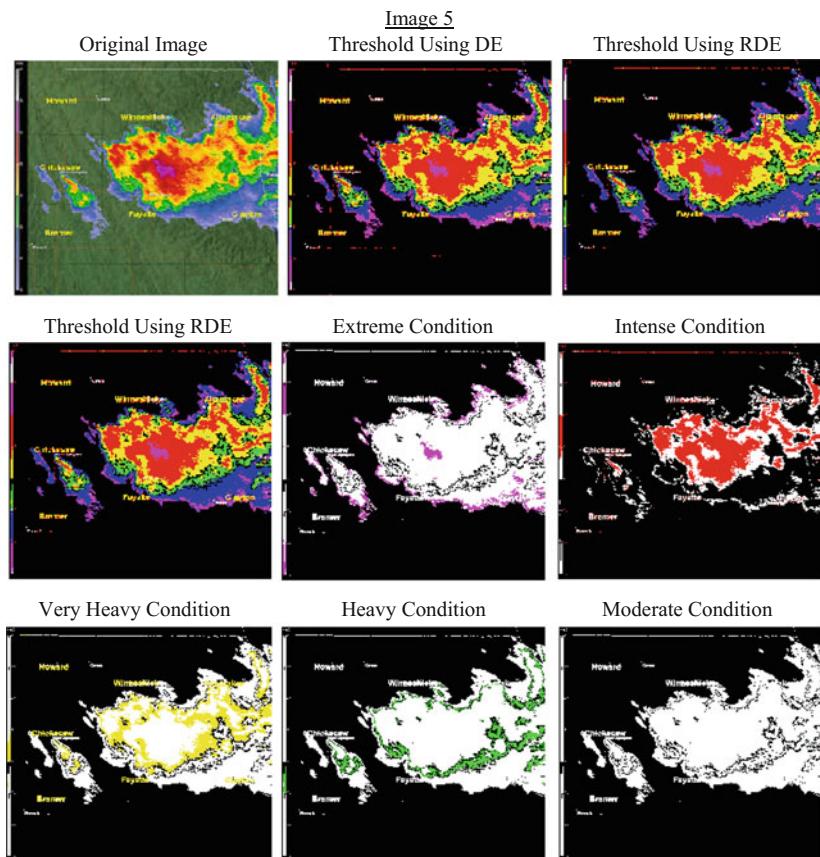


Fig. 6.9 (continued)

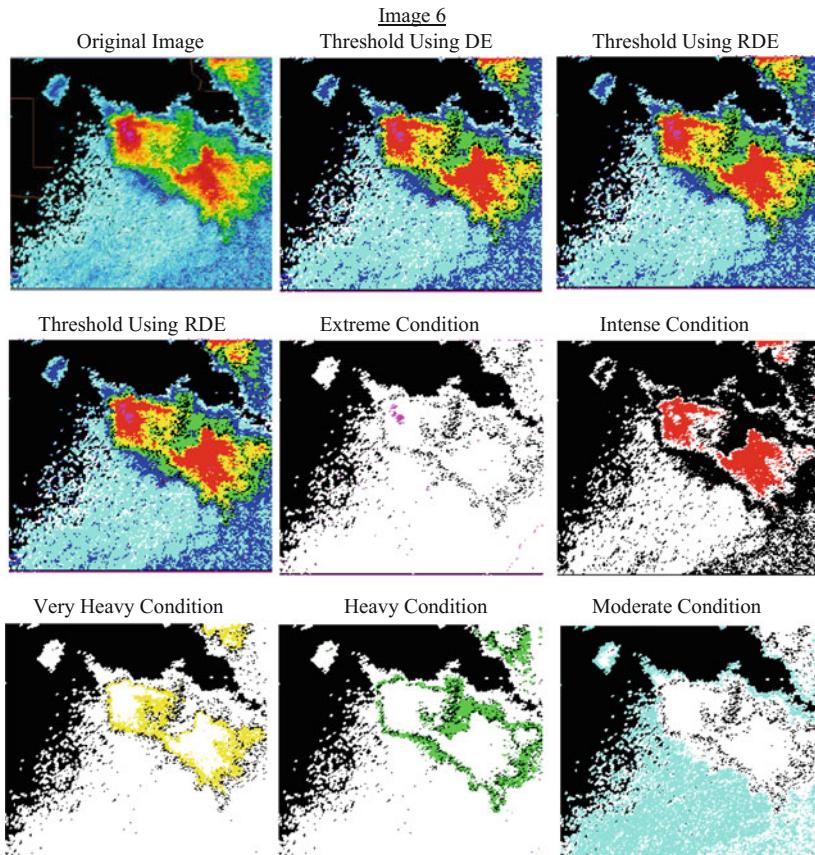


Fig. 6.9 (continued)

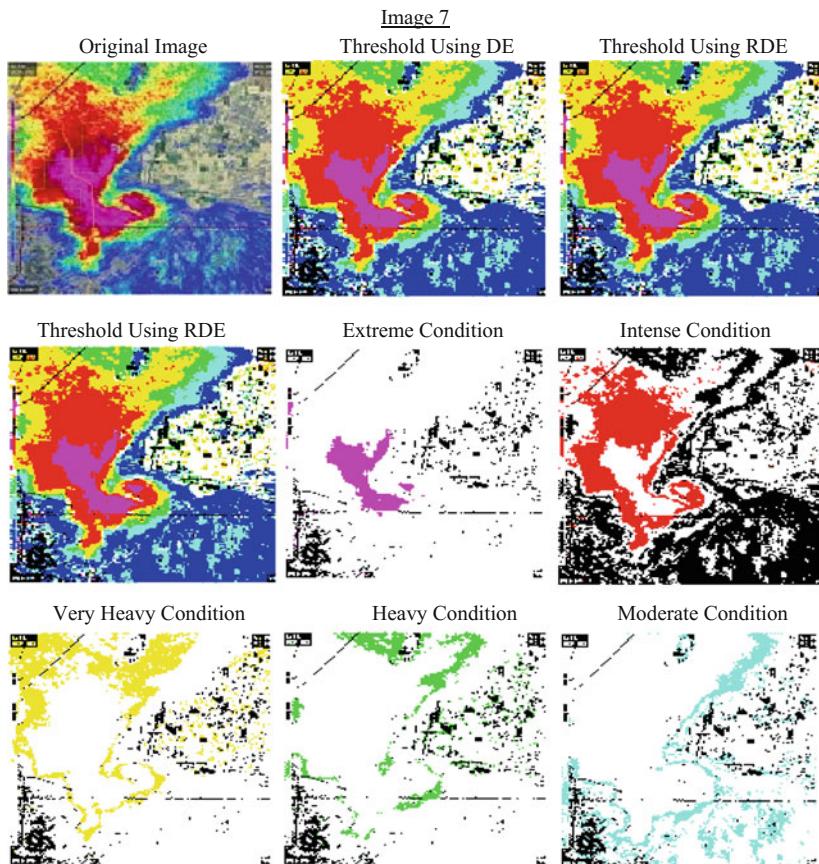


Fig. 6.9 (continued)

Chapter 7

Enhanced Differential Evolution with Fuzzy c-Means Technique



This chapter introduces another new variant of Evolutionary Algorithm named enhanced Differential Evolution (eDE). eDE is incorporated with fuzzy c-means technique to perform clustering of data. In this approach, the search strategy of eDE algorithm is combined with the fuzzy c-means technique and this technique is then applied on clustering of dataset.

7.1 Introduction

With the improvements in technology, the requirement for attaining, storage and handling a vast volume of data is ever growing. Clustering is the process of segregating a collection of data into a group of sub-classes, called clusters where the objects within one sub-class are similar to each other than objects within the other sub-class. It is the process of splitting larger population into minor groups that are comparable. Distinctions and resemblances are assessed on the basis of attribute properties describing the objects. The similarity measure, its way of implementation and the methods capability to identify as many hidden patterns possible, will define the quality of the clustering results. Organizing the data into clusters shall reflect greater intra cluster similarity and less of inter cluster similarity. In clustering, the characteristics of the data become the key variable of the problem and the choice of their selection within the clustering algorithm will greatly affect their results. Hence, the substance for analysis shall always be focused on these characteristics.

Clustering is based upon three characteristics: nesting, exclusiveness and completeness. In the nested type, separation is built on characteristics of nesting groups. Hierarchical clustering is nested meaning it gathers to exist within the bigger clusters. Exclusive separation is the characteristics in which the data object is allowed to exist in one or more than one cluster. Completeness is a sort of separation which needs the entire data objects to be collected. In complete clustering, every item is assigned

to a group. Several stages are included in data clustering namely data gathering, initial selection, depiction, clustering tendency, clustering scheme, justification and analysis.

Partitional algorithm certifies to decompose the data into groups of unrelated clusters using resemblance measure. It is an optimization problem as it diminishes the cluster assigning in terms of probability density. Clustering is used in two manners namely hard (crisp) and soft (fuzzy). In hard clustering, every element is allotted to one group only. The clusters will be separate but not coinciding. K-means algorithm is a type of crisp algorithm. In soft clustering, patterns will be allotted to all the groups based on a fuzzy membership pattern. Fuzzy c means is a type of fuzzy based clustering. When executing cluster analysis on a data set, the data is partitioned into groups depending on their similarity. Each of this group will be assigned a label. Such a group of data is referred to as a cluster. The benefit of performing clustering rather than classification is the flexibility to change and to help differentiate unique features within the group. Clustering is applied in regions like the search engine, web mining, information retrieval and topological analysis. Since no categorized documents are supplied in clustering, they are also referred to as unsupervised learning. To solve unsupervised learning, various evolutionary algorithms are used. It was proved that Evolutionary algorithms work in a robust and efficient manner for clustering according to Coello et al. [1]. These approaches can find global or near global optimal partition over dataset when the number of clusters are given. Differential Evolution (DE) is a simple, stochastic, population based, easy to implement function. It deals with non-differential, multi modal base, non-linear objective functions. Though DE has numerous advantages like ease of use and global exploration, it has few disadvantages too. It suffers from premature convergence. Also, the performance of DE decreases when the size of search space increases. The efficacy and performance of DE is decided by the control parameters and test vector generation strategy. Variants of evolutionary algorithm are created by changing the control parameters to improve the optimization function and also to improve the convergence rate. In this work, we have applied a variant of DE namely eDE (enhanced Differential Evolution) to the field of data clustering. In this variant, three control parameters are used: a constant between (0,2), a random variable between (0,1) and the complement of the random variable used. This variant was applied to cluster standard numerical dataset. eDE gave good results when it was applied on data clustering.

7.2 Fuzzy c-Means Clustering

In few cases of clustering, the clusters formed are not well separated. In fuzzy set theory, objects fit in a cluster with a membership degree between 0 and 1. Fuzzy c-means (FCM) is related to k-means technique. FCM was developed by Dunn [2]. In FCM, the dataset is clustered into n groups where each data in the dataset belong to a particular cluster with a certain degree. It is a type of soft clustering where a data can belong to multiple clusters. Each data has a membership grade which

indicates the degree to which the data belong to a particular cluster. Point on the edge of the cluster will have lower membership degree compared to other data in the cluster. Assume a set of n objects as $x_i = \{x_1, x_2, \dots, x_n\}$. Here, a collection of k clusters are initialized as C_1, C_2, \dots, C_k and a partition matrix $W = w_{ij} \in [0, 1]$ for $i = 1, 2 \dots n$ and $j = 1, 2 \dots k$. w_{ij} denotes the membership degree of the objects.

For a cluster C_j , the centroid c_j is denoted as:

$$c_j = \frac{\sum_{i=1}^n w_{ij}^p x_i}{\sum_{i=1}^n w_{ij}^p} \quad (7.1)$$

where, p is the fuzzifier which denotes the level of fuzziness of the cluster. Larger p means lower membership degree. The membership degree is computed as:

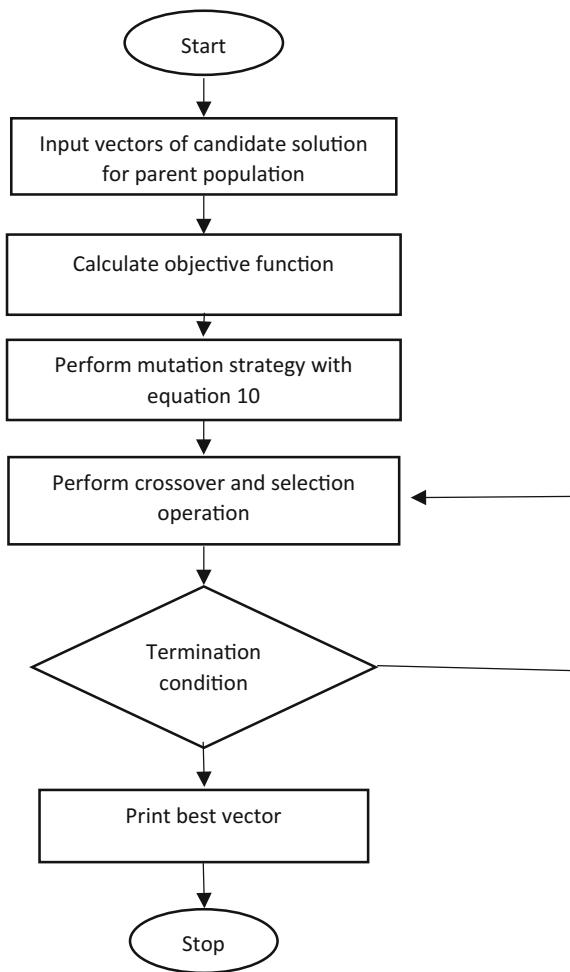
$$w_{ij} = \frac{1}{\sum_{k=1}^c \left(\left\| \frac{x_i - c_j}{x_i - c_k} \right\|^{\frac{1}{p-1}} \right)} \quad (7.2)$$

7.3 Enhanced Differential Evolution

A new strategy has been proposed for mutation called eDE (Enhanced Differential Evolution). This strategy uses three control parameters. The parameter F takes a constant value between (0,2). The parameter $F1$ takes a varying value which lies between (0,1) and $F2$ takes the complement of $F1$. As $F1$ and $F2$ has random values in each iteration, the convergence behavior is enhanced profoundly. As three different control parameters are considered, the value of donor vector is enhanced significantly, and hence, the effectiveness of eDE algorithm is heightened profoundly. Here, two sets of difference vectors are used. So, the preferred perturbation is attained faster. By randomly choosing the vectors, the algorithm is prevented from being greedy. The mutation strategy for the proposed technique is given as:

$$X' = F \times (X_{r1,G}) + F1 \times (X_{best,G} - X_{r2,G}) - F2 \times (X_{best,G} - X_{r3,G}) \quad (7.3)$$

By utilizing the best vector value, the algorithm converges faster in comparison to the traditional strategies. Using more than one difference vectors increases the diversity of population considered. The flow diagram for the proposed technique is shown in Fig. 7.1. The crossover and selection for eDE is same as the classical DE technique.

Fig. 7.1 Flow chart for eDE

7.4 Experimental Setting

eDE was executed using MATLABr2008b and a relative analysis was acquired with five diverse mutation strategy of classical DE algorithm and eDE. Fifteen diverse functions were considered and the results were computed. An optimal value of control parameter F is set as 0.6. As larger value of CR speeds up the convergence rate, for this experiment, CR is set to 0.8. The value to reach (VTR) is the global minimum or maximum of the function to stop the optimization if it is reached. The results are formulated for comparison with the existing algorithms in Table 7.1. By fixing the dimension as 50 and value-to-reach (VTR) as e-015, the best value, and the CPU time of diverse function strategies were considered. The best values for each function is indicated in bold in Table 7.1. The experiment substantiated that eDE

produced better results in case of numerous standard functions. The experiment was also validated by changing the dimension and VTR value.

7.4.1 Statistical Analysis

Friedman test was implemented on the results given in Table 7.1, and the results attained were formulated in Table 7.2. The statistical analysis verifies the efficiency of eDE algorithm. Table 7.3 shows the rank of various mutation strategies used based on best value and CPU time. Tables show that eDE has significantly better performance in comparison to the existing mutation strategies. The rank obtained on the basis of CPU time is the best for eDE and the rank attained on the basis of best value is better for eDE in comparison to the traditional mutation strategies considered. These rankings obtained on the basis of Freidman's test validates the efficiency of eDE technique. The rank obtained on the basis of CPU time taken is depicted in Fig. 7.2.

7.5 eDE in FCM Clustering

eDE algorithm is used in clustering using the FCM technique. Fitness of every variable is accomplished by estimating the distance in-between the centroid and the entity point It is represented as:

$$Fitness(C) = \sum_{j=1}^k \sum_{i=1}^n w_{ij} \|x_i^j - c^j\|^2 \quad (7.4)$$

where x_i^j is the entity point, c^j is the centroid, w_{ij} is the membership degree and $\|x_i^j - c^j\|^2$ provides the distance between the centroid and the entity point. The flow chart for the clustering technique using the variant of DE is given in Fig. 7.3.

7.6 Experimental Results on Clustering

Here, the experiment was shown on six standard datasets with numeric data to compare the execution of the k means algorithm, Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and classical DE with eDE in clustering. The FCM technique of clustering was incorporated with GA, PSO, classical DE and eDE for execution of the data clustering. The resultant cluster graph and curve graph for each dataset was attained. The cluster quality of the clusters acquired were compared. Eight real time datasets from Matlab repository is used. The datasets used are described below:

Table 7.1 Best Value obtained after 25 Runs for diverse functions

Function	DE				eDE
	DE/best/1	DE/rand/1	DE/rand-to-best/1	DE/best/2	
Sphere	9.73×10^{-16}	6.90×10^{-16}	7.53×10^{-16}	9.66×10^{-16}	7.17×10^{-16}
Beale	3.27×10^{-16}	2.32×10^{-16}	3.71×10^{-16}	7.59×10^{-16}	7.73×10^{-16}
Booth	3.50×10^{-16}	2.05×10^{-16}	6.07×10^{-16}	7.08×10^{-16}	8.35×10^{-16}
Schwefel	-1.80×10^3	-2.25×10^3	-7.84×10^1	-1.38×10^3	-1.66×10^3
Michalewicz	-7.64	-7.21	-7.39	-6.95	-6.84
Schaffer N.2	6.60×10^{-16}	8.88×10^{-16}	4.43×10^{-16}	6.55×10^{-16}	8.87×10^{-16}
Schaffer N.4	3.05×10^{-15}	2.90×10^{-1}	2.92×10^{-1}	2.93×10^{-1}	2.89×10^{-1}
HimmelBlau	1.60×10^{-16}	8.05×10^{-16}	3.83×10^{-16}	9.12×10^{-16}	1.46×10^{-16}
Bird	-1.04×10^{-02}	-1.07×10^{-2}	-1.05×10^{-2}	-1.07×10^{-2}	-1.03×10^{-2}
Extended Cube	3.31×10^{-15}	4.98×10^{-5}	6.10×10^{-8}	1.93×10^{-5}	2.68
Ackley	7.19×10^{-15}	6.46×10^{-12}	7.99×10^{-15}	3.63×10^{-13}	3.09
Gold	3.00	3.00	3.00	3.00	3.00
Griewank	9.99×10^{-16}	9.99×10^{-16}	1.60×10^{-13}	6.56×10^{-13}	1.07
Rastrigin	1.79×10^1	1.23×10^2	7.47×10^1	1.28×10^2	1.52×10^2
Rosenbrock	9.60×10^{-16}	1.07×10^{-8}	7.88×10^{-16}	3.90×10^{-9}	1.07×10^1

Table 7.2 Test statistics using Friedman's test

N	50
Chi sq	14.54
Df	5
Asymptotic significance	0.002

Table 7.3 Ranks of the different strategies

Strategies	Mean rank on best Value	Mean rank on CPU time
DE/best/1	3.4	4.7
DE/rand/1	3.7	3.33
DE/best-to-rand/1	2.7	3.67
DE/best/2	4.06	3.7
DE/rand/2	3.9	4.1
eDE	3.1	1.6

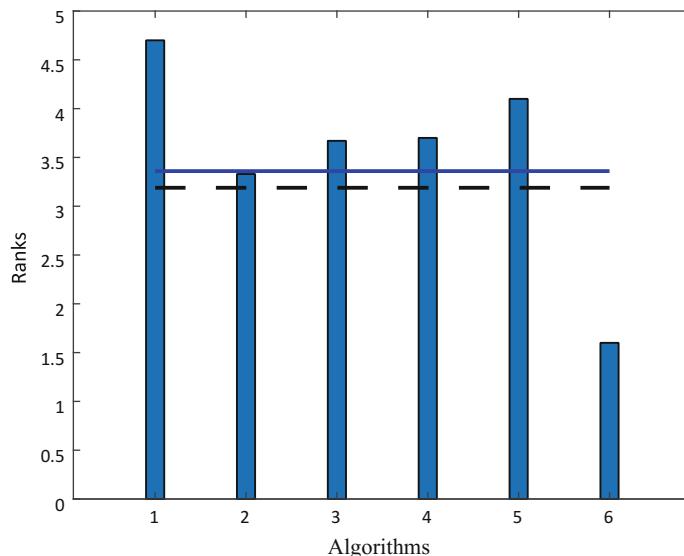


Fig. 7.2 Bonferroni Dunn bar chart for ranks based on CPU time for various strategies

- Moore dataset ($n = 120$, $d = 6$, $k = 2$): This dataset is a lab based report of the biochemical demand on five predictors.
- Cities ($n = 2961$, $d = 9$, $k = 2$): This dataset gives the quality of life ratings for U.S. metropolitan areas.
- Kmeansdata ($n = 100$, $d = 4$, $k = 3$): It is a four-dimensional data provided for clustering.

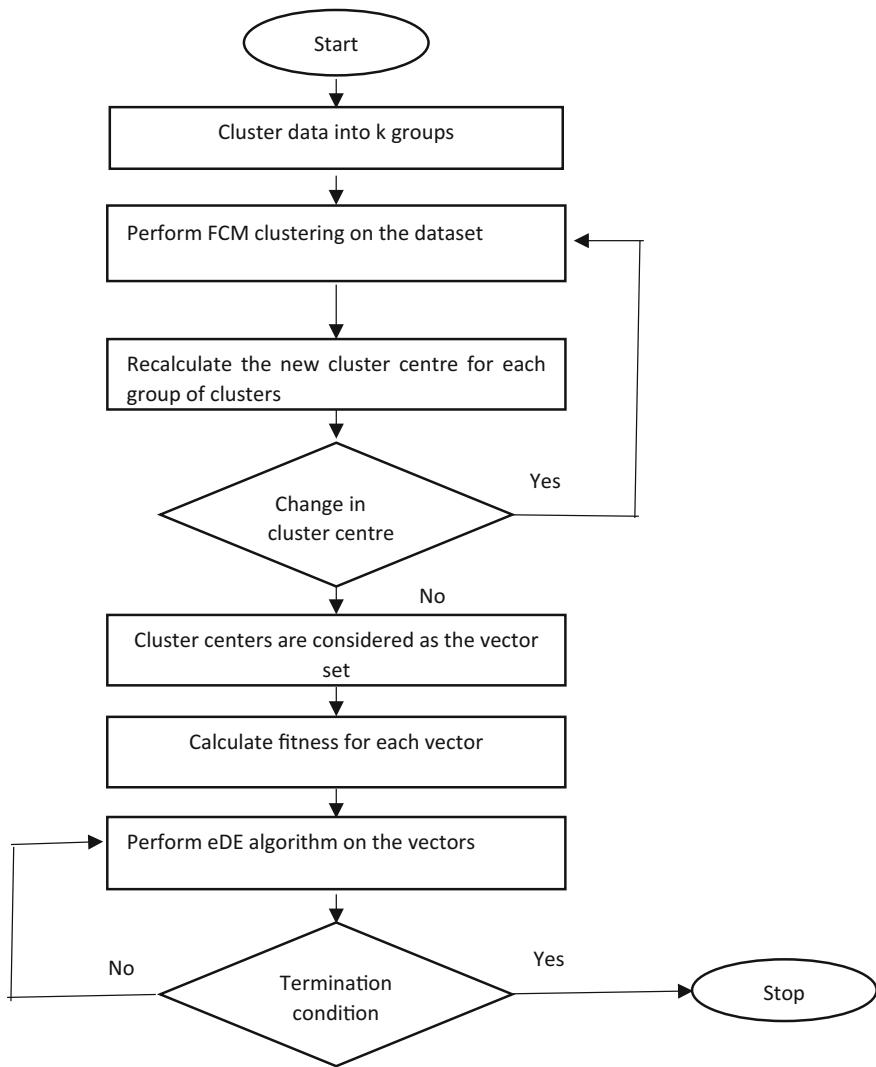


Fig. 7.3 Flow Chart for eDE in FCM clustering

- Reactions ($n = 75, d = 3, k = 3$): It is a reaction kinetic which is a function of three chemical reactions: hydrogen, isopentane and n-pentane. It uses the Hougan-Watson model.
- Fisher Iris dataset ($n = 150, d = 4, k = 3$): This is a standard dataset with 150 inputs for 3 different flower types: sentosa, virginica and versicolour. Here 4 different features of flower are measured: type, petal width, sepal width and sepal length.
- Examgrades ($n = 649, d = 33, k = 3$): The data set represent student performance from two Portuguese schools. Grades of student, demographic and social features

of the school were gathered from school reports and standard questionnaire and used as data attributes. Performance of students in two distinct subjects namely Mathematics (mat) and Portuguese language (por) were given as two datasets.

- Topography ($n = 64,800$, $d = 256$, $k = 3$): The dataset contains numerous representation of earth's topography. The data was supplied by National Geographical Data Centre, NOAA US Department of Commerce under data announcement 88-MGG-02.
- Gatlin ($n = 307,200$, $d = 640$, $k = 3$): The dataset contains numeric data with 640 different attributes.

These datasets were used as input for clustering and results were acquired for different algorithms under consideration. The cluster graph and curve graph for the kmeansdata data set has been given in Fig. 7.4. In the cluster graph, the x-axis shows the position and y-axis shows the distance of data of the kmeansdata set. The curve graphs obtained during various iterations of clustering using DE and eDE algorithm with k means for the kmeansdata dataset is depicted where the x-axis shows the number of iterations and y-axis shows the best cost obtained at each iteration.

7.6.1 Validation Index

There are various quantitative evaluation techniques available to test the cluster quality and these are known as validation index. Numerous validation index is used for testing the quality of cluster which was obtained using FCM technique. It is used as a tool by researchers to test the cluster result. The following are the various validation index considered.

Partition Coefficient (PC)

It defines the extent of overlapping between clusters [3]. Higher values of partition coefficient give good clusters. The formula for partition coefficient is given as:

$$PC = \frac{1}{N} \sum_{i=1}^c \sum_{j=1}^N (\mu_{ij})^2 \quad (7.5)$$

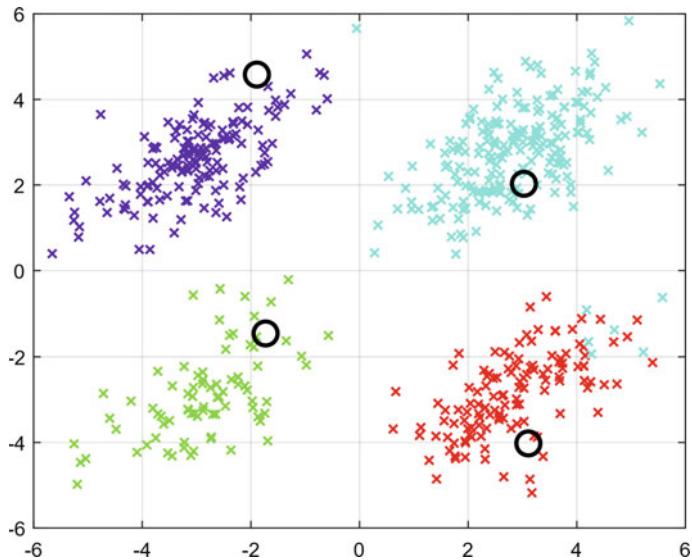
where μ_{ij} is the membership of data j in cluster i . The comparative results for PC is given in Table 7.4. The results obtained show that the result obtained for eDE is best in comparison to the other techniques.

Classification Entropy (CE)

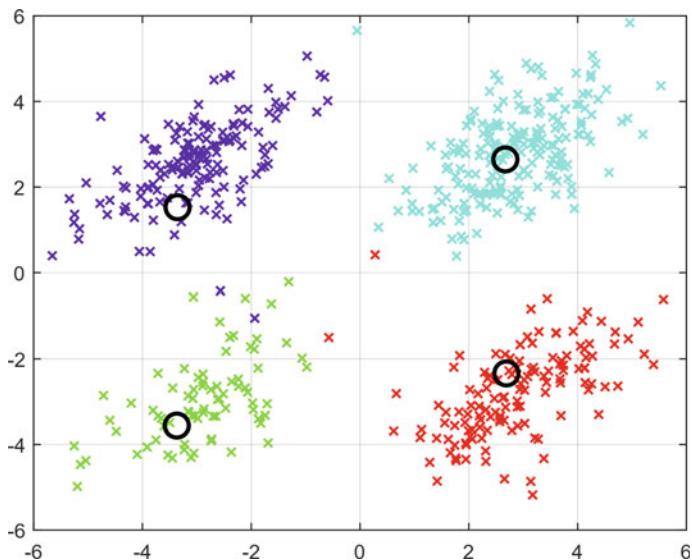
It calculates the fuzziness of the cluster partition. The formula is given as:

$$CE = -\frac{1}{N} \sum_{i=1}^c \sum_{j=1}^N \mu_{ij} \log(\mu_{ij}) \quad (7.6)$$

Cluster at initial stage of iteration using eDE



Cluster at final stage of iteration using eDE

**Fig. 7.4** Cluster diagram and Cluster graph of Kmeans dataset

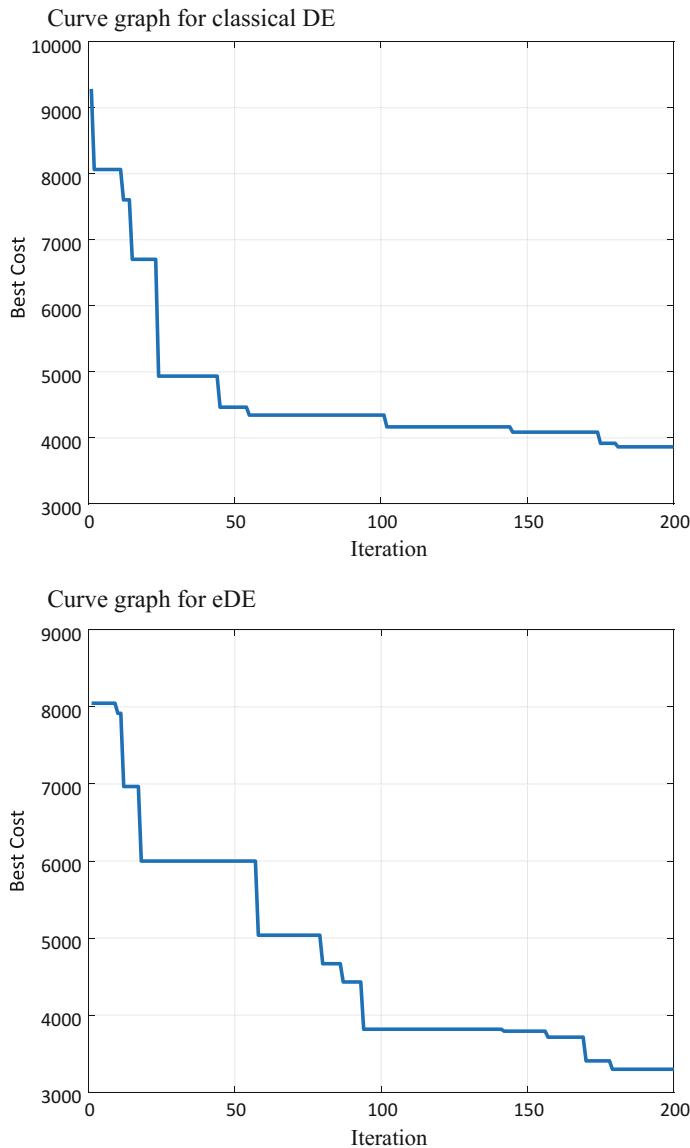


Fig. 7.4 (continued)

Table 7.4 Comparative table for PC

Data sets	Partition coefficient				
	K means	GA	PSO	Classical DE	eDE
Kmeansdata	0.765	0.767	0.761	0.764	0.7677
Iris	0.747	0.723	0.712	0.771	0.773
Examgrades	0.511	0.511	0.482	0.511	0.5112
Moore	0.656	0.657	0.654	0.661	0.6615
Cities	0.366	0.355	0.344	0.313	0.3666
Reactions	0.851	0.813	0.822	0.844	0.8524
Topography	0.534	0.507	0.511	0.531	0.545
Gatlin	0.572	0.54	0.516	0.5624	0.581

Table 7.5 Comparative table for CE

Data sets	Classification entropy				
	K means	GA	PSO	Classical DE	eDE
Kmeansdata	0.445	0.443	0.446	0.447	0.4474
Iris	0.467	0.477	0.473	0.477	0.489
Examgrades	0.832	0.812	0.824	0.837	0.84
Moore	0.612	0.622	0.625	0.621	0.633
Cities	1.473	1.412	1.474	1.472	1.477
Reactions	0.775	0.786	0.772	0.781	0.781
Topography	0.75	0.698	0.802	0.8113	0.82
Gatlin	0.65	0.71	0.69	0.747	0.748

Higher the value of CE, better will be the clusters. The comparative results obtained for CE for the various algorithms are given in Table 7.5. The results obtained show that the result obtained for eDE is best in comparison to the other techniques.

Partition Index (SC)

SC defines the ratio of sum of compactness to separation of the clusters. The formula for SC is given as:

$$SC = \sum_{i=1}^c \frac{\sum_{j=1}^N (\mu_{ij})^m \|x_j - v_i^2\|}{N_i \sum_{k=1}^c \|x_k - v_j^2\|} \quad (7.7)$$

where, v_i is the cluster centre of cluster i , N_i is the number of objects in the cluster i , m is weighting exponent and $\|x_j - v_i^2\|$ is the distance between particles and centroid. Lower the value of SC, better is the cluster formed. The result obtained for SC is tabulated in Table 7.6. The results obtained show that the result obtained for eDE is best in comparison to the other techniques.

Table 7.6 Comparative table for SC

Data sets	Partition index				
	K means	GA	PSO	Classical DE	eDE
Kmeansdata	0.707	0.712	0.699	0.682	0.665
Iris	0.623	0.633	0.612	0.615	0.611
Examgrades	2.012	2.12	2.02	1.89	1.88
Moore	0.7123	0.72	0.689	0.677	0.665
Cities	12.23	12.1	12.18	12.13	12.12
Reactions	2.627	2.58	2.63	2.54	2.54
Topography	0.168	0.168	0.166	0.1646	0.1644
Gatlin	0.079	0.0792	0.078	0.0809	0.08

Table 7.7 Comparative table for Separation Index

Data sets	Separation index				
	K means	GA	PSO	Classical DE	eDE
Kmeansdata	0.0015	0.0012	0.002	0.001	0.0008
Iris	0.0063	0.0068	0.006	0.005	0.0054
Examgrades	0.0285	0.0292	0.291	0.286	0.2852
Moore	0.0572	0.0587	0.058	0.057	0.0571
Cities	0.0915	0.0923	0.092	0.093	0.0898
Reactions	0.087	0.089	0.092	0.086	0.082
Topography	0.002	0.0021	0.0013	0.0012	0.001
Gatlin	2.18e-04	2.17e-04	2.2e-04	2.14e-04	2.14e-04

Separation Index (S)

This index deals with the minimum distance separation for cluster validity. The formula for Separation Index is given as:

$$S = \frac{\sum_{i=1}^c \sum_{j=1}^N (\mu_{ij})^2 \|x_j - v_i^2\|}{N \min_{i,k} \|v_k - v_i^2\|} \quad (7.8)$$

Lower the separation index, better is the cluster formed. The result obtained for separation index is tabulated in Table 7.7. The results obtained show that the result obtained for eDE is best in comparison to the other techniques.

Table 7.8 Comparative table for XB

Data sets	Xie Beni index				
	K means	GA	PSO	Classical DE	eDE
Kmeansdata	4.987	4.97	4.82	4.78	4.78
Iris	3.804	3.812	3.807	3.803	3.801
Examgrades	1.267	1.278	1.264	1.212	1.189
Moore	1.431	1.442	1.414	1.387	1.36
Cities	1.473	1.456	1.453	1.452	1.401
Reactions	28.53	28..12	28.78	27.89	27.76
Topography	0.821	0.821	0.818	0.8053	0.812
Gatlin	0.975	0.966	0.975	0.9623	0.9601

Table 7.9 Comparative table for Execution time

Data sets	Execution time				
	K means	GA	PSO	Classical DE	eDE
Kmeansdata	13.45	13.12	13.54	13.12	13.01
Iris	15.1	25.018	15.34	25.03	14.13
Examgrades	11.3	10.67	12.3	11.23	11.21
Moore	8.14	8.87	8.34	9.99	8.12
Cities	37.1	37.6	36.7	37.24	36.9
Reactions	15.43	16.12	15.67	17.23	15.12
Topography	0.813	0.8002	0.813	0.8053	0.8021
Gatlin	0.943	0.975	0.954	0.9623	0.9613

Xie Beni Index (XB)

XB was proposed by Xie and Beni [4]. It measures the ratio of total variation of within cluster to separation of clusters. XB is given as:

$$XB = \frac{\sum_{i=1}^c \sum_{j=1}^N (\mu_{ij})^m \|x_j - v_i^2\|}{N \min_{i,j} \|x_j - v_i^2\|} \quad (7.9)$$

Lower the XB value, better is the cluster formed. The result obtained for XB is given in Table 7.8. The results obtained show that the result obtained for eDE is best in comparison to the other techniques.

Execution Time

It is the total time taken for the execution of task. Lower the execution time, better the cluster. Execution time for the various algorithms are shown in Table 7.9. The results obtained show that the result obtained for eDE is best in comparison to the other techniques.

Table 7.10 Comparative table for DI Index

Data sets	Dunn index				
	K means	GA	PSO	Classical DE	eDE
Kmeansdata	0.0222	0.018	0.02	0.011	0.0255
Iris	0.028	0.03	0.03	0.034	0.034
Examgrades	0.128	0.11	0.114	0.1189	0.1289
Moore	0.221	0.22	0.234	0.255	0.255
Cities	0.045	0.04	0.067	0.056	0.0691
Reactions	0.443	0.33	0.333	0.441	0.443
Topography	0.131	0.134	0.121	0.135	0.139
Gatlin	0.087	0.0767	0.0862	0.0905	0.0909

Dunn Index (DI)

It is a matrix for evaluating the cluster quality. It is a function of ratio of sum of intra-distances to inter distances [2]. It tries to find a good intra cluster and inter cluster association. It is the ratio of inter cluster to intra cluster distance of the clusters. Here, larger the value of DI index, better the clusters formed. The formula for DI index is given as:

$$DI = \min \left\{ \frac{\min d((c_i, c_j))}{\max d'(l)} \right\} \quad (7.10)$$

where, i, j are cluster labels, $d'(k)$ is the average distance between cluster elements to center of cluster l , $d((c_i, c_j))$ is the distance between these centroids. The comparative results obtained for the DI index for the various algorithms are shown in Table 7.10. The results obtained show that the result obtained for eDE is best in comparison to the other techniques.

Alternative Dunn Index (ADI)

It is a variation of DI index. Larger the value of ADI index, the better the cluster that is formed. The formula for ADI index is given as:

$$ADI = \min \left\{ \frac{d(x_i, v_i) - d(x_j, v_j)}{\max d'(l)} \right\} \quad (7.11)$$

where v_i shows the center of cluster i . The relative comparison obtained for the ADI index is given in Table 7.11. The results obtained show that the result obtained for eDE is best in comparison to the other techniques.

Table 7.11 Comparative table for ADI index

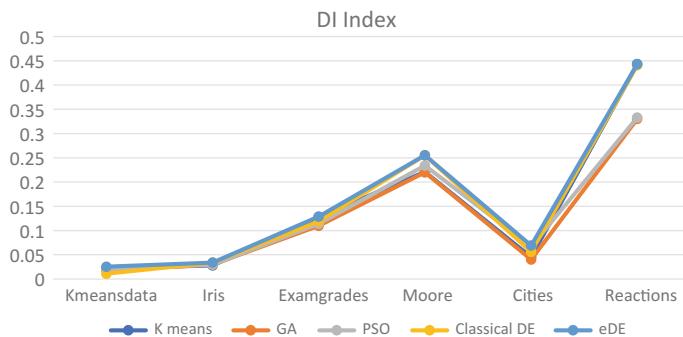
Data sets	ADI index				
	K means	GA	PSO	Classical DE	eDE
Kmeansdata	0.001	0.0013	0.002	0.001	0.0021
Iris	0.008	0.0068	0.009	0.009	0.0096
Examgrades	0.051	0.045	0.034	0.048	0.0531
Moore	0.045	0.036	0.043	0.046	0.0489
Cities	0.001	0.0010	0.001	0.002	0.0029
Reactions	0.114	0.113	0.111	0.013	0.115
Topography	9.23e-04	9.43e-04	9.28e-04	9.5e-04	9.5e-04
Gatlin	5.18e-05	5.23e-05	5.16e-05	5.15e-05	5.17e-05

7.6.2 Graphical Representation

The tabulated values of Validation index have been depicted graphically. Figure 7.5 show the performance curve for DI validation index. The x-axis represents the different datasets used and y-axis represents the value obtained. The line graph shows that the values obtained for the eDE is better than the values obtained from classical DE approach. The values have been recorded for eight different datasets.

7.7 Summary

In this work, a new variant of differential Evolution was proposed and named as eDE. eDE was compared with diverse mutation strategies of DE. The comparative study shows the superior performance of eDE. The ranks computed also justifies the efficiency of the strategy. eDE was applied with FCM technique for clustering

**Fig. 7.5** Curve for DI Index

standard numeric datasets. The cluster quality showed the efficiency of the variant developed. The work can be extended to the field of image thresholding, texture enhancement etc. for displaying the performance of the new mutation strategy in that area.

References

1. Coello, C.A.C.: Multi-objective evolutionary algorithms in real-world applications: some recent results and current challenges. Advances in Evolutionary and Deterministic Methods for Design, pp. 3–18. Springer International Publishing, Optimization and Control in Engineering and Sciences (2015)
2. Dunn, J.C.: A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *J. Cybern.* **3**, 32–57 (1973)
3. Bezdek, J.C.: Cluster validity with fuzzy sets. *J. Cybern.* **3**(3), 58–73 (1973)
4. Xie, X.L., Beni, G.: A validity measure for fuzzy clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* **8**, 841–847 (1991)

Chapter 8

Conclusion and Future Scope



This research work has in principle focused on creation of new variants of meta-heuristic algorithms like the DE and FPA. The work also highlights the various applications of these new variants, particularly in the field of clustering of data and images. The variant ReDE first explained is compared with 5 classical mutation strategies (DE/best/1, DE/rand/1, DE/best-to-rand/1, DE/best/2, DE/rand/2) and subsequently an analysis is performed based on fifteen standard functions. The statistical analysis is performed to analyse the performance of the variant ReDE. The ranks corresponding to various strategies are computed in Table 3.32 which reveal that ReDE performs better than the other mutation strategies. Bonferroni Dunn chart for the ranks obtained is also shown in Figs. 3.3–3.5. In the following chapter, another new variant ssFPA/DE is developed using DE and Flower Pollination Algorithm.

The next new variant FSDE is developed by changing the mutation strategy of DE. For both ssFPA/DE and FSDE, the variants created were compared with 5 classical mutation strategies (DE/best/1, DE/rand/1, DE/best-to-rand/1, DE/best/2, DE/rand/2) and analysis is done based on fifteen standard functions. The statistical analysis is tabulated in Tables 4.9 and 5.8 respectively. Also, both ssFPA/DE and FSDE are applied on real datasets from UCI repository and the cluster quality results obtained for the variants are compared with the results of k-means, genetic algorithm, PSO and classical DE. The results obtained show that ssFPA/DE and FSDE are better for clustering compared to the other evolutionary algorithms.

Another new variant, the RDE (Reconstructed Differential Evolution) is developed by changing the mutation strategy of DE. The variant is compared with 5 classical mutation strategies (DE/best/1, DE/rand/1, DE/best-to-rand/1, DE/best/2, DE/rand/2) and analysis is done based on fifteen standard functions. The statistical analysis is done for performance analysis and the results are tabulated in Table 6.2. The corresponding ranks for the various strategies computed in Table 6.3 show that RDE performs better than the other mutation strategies. Bonferroni Dunn chart for the ranks obtained is shown in Figs. 6.1–6.3. RDE is implemented on multi-level thresholding based on fuzzy entropy. After comparison with the classical DE approach, the thresholding results are found to be better for RDE strategy. The thresh-

old image is further segregated based on the specific hazard level. The result of this study demonstrate that the RDE strategy displays better efficiency in comparison to other strategies.

A new variant, the eDE (enhanced Differential Evolution) is developed by varying the mutation strategy of DE. The variant is compared with 5 classical mutation strategies (DE/best/1, DE/rand/1, DE/best-to-rand/1, DE/best/2, DE/rand/2) and analysis is done based on fifteen standard functions. The statistical analysis is done for performance analysis and the results are tabulated in Table 7.2. The corresponding ranks for the various strategies computed in Table 7.3 show that eDE performs better than the other mutation strategies. eDE is implemented on clustering data based on fuzzy entropy. After comparison with the classical DE approach, the clustering results are found to be better for eDE strategy.

8.1 Future Scope

A further extension of this research can be carried forward by employing these algorithms for clustering of documents and text information. As a part of this future extension, there is scope to modify the crossover strategy of DE. In other attempts, work can be extended to create more efficient variants of any other evolutionary algorithms like the Ant Colony Optimization, Firefly Algorithm etc. In future, research can also be done on less explored nature inspired algorithms like glow worm, bean optimization, amoeba based algorithm etc. The new variants of metaheuristics algorithms developed as a result can be used for solving other complex problems like image compression, clustering of networks, video segmentation, texture enhancement etc. Also these variants can be implemented in social awareness areas of application like medical diagnosis, remote sensing, geographical explorations etc.

Appendix

Benchmark functions used:

Global optimization approaches should be verified with the benchmark functions or problems. A wide range of test functions are designed to signify the different parts of global optimization algorithm. They can be extended to arbitrary dimensionality to allow scaled testing. Unimodal and multimodal functions have been used in testing the algorithm. The various benchmark functions are discussed in detail:

i. Sphere Function (f1):

The Sphere function has d local minima except for the global one. It is continuous, convex and unimodal. The maximum and minimum range is between (-5.12,5.12). The equation is given as:

$$f(x) = \sum_{i=1}^d x_i^2 \quad (\text{A.1})$$

ii. Beale Function (f2):

The Beale function is multimodal, with sharp peaks at the corners of the input domain. The maximum and minimum range is between (-4.5,4.5).

$$f(x) = (1.5 - x_1 + x_1 x_2)^2 + (2.25 - x_1 + x_1 x_2^2)^2 + (2.625 - x_1 + x_1 x_2^3)^2 \quad (\text{A.2})$$

iii. Booth Function (f3):

The function is usually evaluated on the square $x_i \in [-10,10]$, for all $i = 1, 2$. There are several local minima for this function. So it is a multimodal function.

$$f(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2 \quad (\text{A.3})$$

iv. Schwefel Function (f4):

The Schwefel function is complex, with many local minima. The plot shows the two-dimensional form of the function. The function is usually evaluated on the hypercube $x_i \in [-500, 500]$, for all $i = 1, \dots, d$.

$$f(x) = 418.9829 d - \sum_{i=1}^d x_i \sin\left(\sqrt{|x_i|}\right) \quad (\text{A.4})$$

v. Michalewicz Function (f5):

The Michalewicz function has local minima, and it is multimodal. The parameter m defines the steepness of t valleys and ridges; a larger m leads to a more difficult search. The recommended value of m is $m = 10$. The maximum and minimum range is between $(0, \pi)$.

$$f(x) = - \sum_{i=1}^d \sin(x_i) \sin^{2m}\left(\frac{ix_i^2}{\pi}\right) \quad (\text{A.5})$$

vi. Schaffner Function N.2 (f6):

It is shown on a smaller input domain in the second plot to show detail. The function is usually evaluated on the square $x_i \in [-100, 100]$, for all $i = 1, 2$.

$$f(x, y) = 0.5 + \frac{\sin^2(x^2 - y^2) - 0.5}{(1 + 0.001(x^2 + y^2))^2} \quad (\text{A.6})$$

vii. Schaffner Function N.4 (f7):

It is shown on a smaller input domain in the second plot to show detail. The function is usually evaluated on the square $x_i \in [-100, 100]$, for all $i = 1, 2, \dots, N$.

$$f(x, y) = 0.5 + \frac{\cos^2(\sin^2(|x^2 - y^2|)) - 0.5}{(1 + 0.001(x^2 + y^2))^2} \quad (\text{A.7})$$

viii. HimmelBlau Function (f8):

It is a multimodal function used to solve optimization problems. The function is evaluated on $x_i \in [-5,5]$, for all $i = 1, 2, \dots, d$.

$$f(x, y) = (x^2 + y - 11)^2 + (y^2 + x - 7)^2 \quad (\text{A.8})$$

ix. Bird Function (f9):

This is a bi-modal function with $f(x^*) = 106.764537$ in the search domain $[-2,2]$. The maximum and minimum range is given as $(-2\pi, 2\pi)$.

$$f(x, y) = \sin(x)e^{(1-\cos(y))^2} + \cos(y)e^{(1-\sin(x))^2} + (x - y)^2 \quad (\text{A.9})$$

x. Extended Cube Function (f10):

This is a multimodal minimization problem for global optimization. Here, n represents the number of dimensions and the maximum and minimum range is between $(-100, 100)$.

$$f(x) = \sum_{i=1}^n 100(x_{i+1} - x_i^3)^2 + (1 - x_i)^2 \quad (\text{A.10})$$

xi. Ackley Function (f11):

This function is used mainly to test optimization algorithms. The equation for this function is as given below:

$$f(x) = -a \exp \left(-b \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2} \right) - \exp \left(\frac{1}{d} \sum_{i=1}^d \cos(cx_i) \right) + a + \exp(1) \quad (\text{A.11})$$

The function is a risk for many optimization problems to get trapped in one of its many local minima. The maximum and minimum range is between $(-32, 32)$.

xii. Goldstein-Price Function (f12):

The Goldstein-Price function has several local minima. The function is usually evaluated on the square $x_i \in [-2, 2]$, for all $i = 1, 2$. The maximum and minimum range is between $(-2, 2)$.

$$f(x) = (1 + (x + y + 1)^2(19 - 14x + 3x^2 - 14y + 6xy + 3y^2))(30 + (2x - 3y)^2(18 - 32x + 12x^2 + 48y - 36xy + 27y^2)) \quad (\text{A.12})$$

xiii. Griewank Function (f13):

The Griewank function has many widespread local minima, which are regularly distributed. The maximum and the minimum range is between (-600,600).

$$f(x) = \sum_{i=1}^d \frac{x_i^2}{4000} - \prod_{i=1}^d \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad (\text{A.13})$$

xiv. Rastrigin function (f14):

The Rastrigin function has several local minima. It is highly multimodal, but locations of the minima are regularly distributed. The maximum and minimum range is between (-15,15).

$$f(x) = 10d + \sum_{i=1}^d [x_i^2 - 10 \cos(2\pi x_i)] \quad (\text{A.14})$$

xv. Rosenbrock Function (f15):

The Rosenbrock function, also referred to as the Valley or Banana function, is a popular test problem for gradient-based optimization algorithms. The maximum and minimum range is between (-15,15).

$$f(x) = \sum_{i=1}^{d-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right] \quad (\text{A.15})$$

The function is unimodal, and the global minimum lies in a narrow, parabolic valley. However, even though this valley is easy to find, convergence to the minimum is difficult.

Bibliography

1. Abdel-Raouf, O., Abdel-Baset, M., El-henawy, I.: An improved flower pollination algorithm with chaos. *Int. J. Educ. Manag. Eng. (IJEME)* **4**(2), 1–8 (2014)
2. Abraham, A., Das, S., Konar, A.: Automatic clustering using an improved differential evolution algorithm. *IEEE Trans. Syst. Man Cybern.-Part A: Syst. Hum.* **38**(1), 218–237 (2008)
3. Abraham, A., Das, S., Roy, S.: Swarm intelligence algorithms for data clustering. In: *Soft Computing for Knowledge Discovery and Data Mining*, pp. 279–313. Springer, US (2008)
4. Ali, M., Pant, M., Abraham, A.: Simplex differential evolution. *Acta Polytech. Hung.* **6**(5), 95–115 (2009)
5. Ali, M., Ahn, C.W.: An optimized watermarking technique employing SVD in DWT domain. In: *ICUIMC-2013*, p. 86 (2013)
6. Ashlock, D., McEachern, A.: Evolutionary nonlinear projection. *IEEE Trans. Evol. Comput.* **19**(6), 857–869 (2015)
7. Ayala, H.V.H., Marins dos Santos, F., Viviana, C.M., Leandro dos Santos, C.: Image thresholding segmentation based on a novel beta differential evolution approach. *Expert Syst. Appl.* **42**(4), 2136–2142 (2015)
8. Back, T., Hammel, U., Schwefel, H.P.: Evolutionary computation: comments on the history and current state. *IEEE Trans. Evol. Comput.* **1**(1), 3–17 (1997)
9. Bhandari, A.K., Kumar, A., Singh, G.K.: Tsallis entropy based multilevel thresholding for colored satellite image segmentation using evolutionary algorithms. *Expert Syst. Appl.* **42** (22), 8707–8730 (2015)
10. Bhandari, A.K., Kumar, A., Singh, G.K.: Modified artificial bee colony based computationally efficient multilevel thresholding for satellite image segmentation using Kapur's, Otsu and Tsallis functions. *Expert Syst. Appl.* **42**(3), 1573–1601 (2015)
11. Bhandari, A.K., Kumar, A., Chaudhary, S., Singh, G.K.: A novel color image multilevel thresholding based segmentation using nature inspired optimization algorithms. *Expert Syst. Appl.* **63**, 112–133 (2016)
12. Bhattacharya, A., Pranab, K.C.: Hybrid differential evolution with biogeography-based optimization for solution of economic load dispatch. *IEEE Trans. Power Syst.* **25**(4), 1955–1964 (2010)
13. Blum, C. and Roli, A.: Metaheuristics in combinatorial optimization: overview and conceptual comparison. *ACM Comput. Surv. (CSUR)* **35**(3), 268–308 (2003)
14. Boussaid, I., Lepagnot, J., Siarry, P.: A survey on optimization metaheuristics. *Inf. Sci.* **237**, 82–117 (2003)

15. Chakraborty, D., Sankhadip, S., Dutta, O.: DE-FPA: A hybrid differential evolution-flower pollination algorithm for function minimization. In: 2014 International Conference in High Performance Computing and Applications (ICHPCA), pp. 1–6. IEEE (2014)
16. Chao, Y., Min, D., Kai, C., Ping, C., Zhang, Z.: Fuzzy entropy based multilevel image thresholding using modified gravitational search algorithm. In: 2016 IEEE International Conference on Industrial Technology (ICIT), pp. 752–757 (2016)
17. Cortés A.P., Josue, R.G., Luis, A.V.V., Marco, A.R.S., Molina-Lozano, H., Ildar, B.: Design and implementation of differential evolution algorithm on FPGA for double-precision floating-point representation. *Acta Polytech. Hung.* **11**(4), 139–153 (2014)
18. Das, S., Konar, A., Chakraborty, U.K.: Two improved differential evolution schemes for faster global search. In: Proceedings of the 7th annual conference on Genetic and evolutionary computation, pp. 991–998. ACM (2005)
19. Das, S., Abraham, A., Konar, A.: Particle swarm optimization and differential evolution algorithms: technical analysis, applications and hybridization perspectives. *Advances of Computational Intelligence in Industrial Systems*, pp. 1–38. Springer, Berlin Heidelberg (2008)
20. Dorigo, M., Gambardella, L.M.: Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Trans. Evol. Comput.* **1**(1), 53–66 (1997)
21. El, A.A., Mbarek, N., Mohamed, M., Jamal, M.: Evolutionary algorithm for segmentation of medical images by region growing. In: 2016 13th International Conference on Computer Graphics, Imaging and Visualization (CGiV), pp. 119–124. IEEE (2016)
22. Fister Jr, X.S., Yang, X.S., Fister, I., Brest, J., Fister, D.: A brief review of nature-inspired algorithms for optimization. arXiv preprint arXiv:1307.4186 (2013)
23. Fong, S., Suash, D., Yang, X.-S., Zhuang, Y.: Towards enhancement of performance of K-means clustering using nature-inspired optimization algorithms. *Sci World J.*, 16 (2014)
24. Gandomi, A.H., Yang, X.-S., Talatahari, S., Deb, S.: Coupled eagle strategy and differential evolution for unconstrained and constrained global optimization. *Comput. Math Appl.* **63**(1), 191–200 (2012)
25. Guo, P.F., Bhattacharya, P., Nawwaf, K.: An efficient image pattern recognition system using an evolutionary search strategy. In: SMC 2009. IEEE International Conference In Systems, Man and Cybernetics, pp. 599–604. IEEE (2009)
26. Gutjahr, W.J.: Convergence analysis of metaheuristics. *Matheuristics*, pp. 159–187. Springer, US (2009)
27. Halder, U., Das, S., Dipankar, M.: A cluster-based differential evolution algorithm with external archive for optimization in dynamic environments. *IEEE Trans. Cybern.* **43**(3), 881–897 (2013)
28. Han, K.H., Kim, J.H.: Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. *IEEE Trans. Evol. Comput.* **6**(6), 580–593 (2002)
29. Islam, S.M., Das, S., Ghosh, S., Roy, S., Suganthan, P.N.: An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization. *IEEE Trans. Syst. Man Cybern. B Cybern.* **42**(2), 482–500 (2012)
30. Kumar, S., Kumar, P., Sharma, T.K., Pant, M.: Bi-level thresholding using PSO, artificial bee colony and MRLDE embedded with Otsu method. *Memetic Computing* **5**(4), 323–334 (2013)
31. Martínez E.A.C., César, H.M., Francisco, J.M.E., Nicolás, G.P.: Hybridization of evolutionary algorithms and local search by means of a clustering method. *IEEE Trans. Syst. Man Cybern. Part B (Cybernetics)* **36**(3), 534–545 (2005)
32. Maulik, U., Sanghamitra, B., Indrajit, S.: Integrating clustering and supervised learning for categorical data analysis. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **40**(4), 664–675 (2010)
33. Mezura, M.E., Jesús, V.R., Coello, C.A.C.: A comparative study of differential evolution variants for global optimization. In: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, pp. 485–492. ACM (2006)

34. Moh'd Alia, O., Mandava, R.: The variants of the harmony search algorithm: an overview. *Artif. Intell. Rev.* **36**(1), 49–68 (2011)
35. Nakib, A., Oulhadj, H., Siarry, P.: Image histogram thresholding based on multiobjective optimization. *Sig. Process.* **87**, 2516–2534 (2007)
36. Norman, N., Iba, H.: Accelerating differential evolution using an adaptive local search. *IEEE Trans. Evol. Comput.* **12**(1), 107–125 (2008)
37. Oh, J., Heesoo, H.: Feature enhancement of medical images using morphology-based homomorphic filter and differential evolution algorithm. *Int. J. Control Autom. Syst.* **8**(4), 857–861 (2010)
38. Osman, I.H., Kelly, J.P.: Meta-heuristics: an overview. *Meta-Heuristics*, pp. 1–21. Springer, US (1996)
39. Rahnamayan, S., Hamid, R.T., Magdy, M.A.S.: Opposition-based differential evolution. *IEEE Trans. Evol. Comput.* **12**(1), 64–79 (2008)
40. Raidl, G.R.: An efficient evolutionary algorithm for the degree-constrained minimum spanning tree problem. In: *Proceedings of the 2000 Congress on Evolutionary Computation*, vol. 1, pp. 104–111. IEEE (2000)
41. Ramadas, M., Pant, M., Abraham, A., Kumar, S.: ssFPA/DE: an efficient hybrid differential evolution–flower pollination algorithm based approach. *Int. J. Syst. Assur. Eng. Manag.*, 1–14 (2016)
42. Ramadas, M., Kumar, S.: An efficient hybrid approach using differential evolution and flower pollination algorithm. In: *2016 6th International Conference-Cloud System and Big Data Engineering (Confluence)*, pp. 59–64. IEEE (2016)
43. Saha, S., Bandyopadhyay, S.: A generalized automatic clustering algorithm in a multiobjective framework. *Appl. Soft Comput.* **13**(1), 89–108 (2013)
44. Sarkar, S., Gyana, R.P., Das, S.: A differential evolution based approach for multilevel image segmentation using minimum cross entropy thresholding. In: *International Conference on Swarm, Evolutionary, and Memetic Computing*, pp. 51–58 . Springer, Berlin Heidelberg (2011)
45. Shu, M.G., Chin-Chang, Y.: Enhancing differential evolution utilizing eigenvector-based crossover operator. *IEEE Trans. Evol. Comput.* **19**(1), 31–49 (2015)
46. Sun, J., Garibaldi, J.M., Hodgman, C.: Parameter estimation using metaheuristics in systems biology: a comprehensive review. *IEEE/ACM Trans. Comput. Biol. Bioinf.* **9**(1), 185–202 (2012)
47. Test_functions_for_optimization: In Wikipedia. Retrieved on 2 Jan 2016 from https://en.wikipedia.org/wiki/Test_functions_for_optimization (n.d.)
48. Wagner, G.P., Altenberg, L.: Perspective: complex adaptations and the evolution of evolvability. *Evolution*, 967–976 (1996)
49. Wang, H., Rahnamayan, S., Hui, S., Mahamed, G.H.O.: Gaussian bare-bones differential evolution. *IEEE Trans. Cybern.* **43**(2), 634–647 (2013)
50. Wu, G., Rammohan, M., Suganthan, P.N., Wang, R., Huangke, C.: Differential evolution with multi-population based ensemble of mutation strategies. *Inf. Sci.* **329**, 329–345 (2016)
51. Xie, X.L., Gerardo, B.: A validity measure for fuzzy clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* **8**, 841–847 (1991)
52. Yang, X.S.: Firefly algorithm, Levy flights and global optimization. *Research and development in intelligent systems XXVI*, pp. 209–218. Springer, London (2010)
53. Yogamangalam, R., Karthikeyan, B.: Segmentation techniques comparison in image processing. *Int. J. Eng. Technol. (IJET)* **5**(1), 307–313 (2013)
54. Yong, W., Zixing, C., Qingfu, Z.: Differential evolution with composite trial vector generation strategies and control parameters. *IEEE Trans. Evol. Comput.* **15**(1), 55–66 (2011)
55. Zhang, Q., Li, H.: MOEA/D: a multiobjective evolutionary algorithm based on decomposition. In: *IEEE Trans. Evol. Comput.* **11**(6), 712–731 (2007)