Multimodal Optimization using Self-Adaptive Real Coded Genetic Algorithm with K-means & Fuzzy C-means Clustering

Vrushali K. Bongirwar
IVth semester M.Tech., CSE,
RCERT., Chandrapur(m.s.) India

Rahila Patel
RCERT, Chandrapur (M.S.)
India Pin: 442403

***Abstract*— Many engineering optimization tasks involve finding more than one optimum solution. These problems are considered as Multimodal Function Optimization Problems. Genetic Algorithm can be used to search Multiple optimas, but some special mechanism is required to search all optimum points. Different genetic algorithms are proposed, designed and implemented for the multimodal Function Optimization.**

In this paper, we proposed an innovative approach for Multimodal Function Optimization. Proposed Genetic algorithm is a Self Adaptive Genetic Algorithm and uses Clustering Algorithm for finding Multiple Optimas. Experiments have been performed on various Multimodal Optimization Functions. The Results has shown that the Proposed Algorithm given better performance on some Multimodal Functions.

Keywords*—*Genetic Algorithm (GA); self-adaptation; Multimodal Function Optimization; K-Means Clustering; Fuzzy C-Means Clustering.

I. INTRODUCTION

Optimization means solving problems in which one seeks to minimize or maximize a real function by systematically choosing the values of real or integer variables from within an allowed set.[5][12]

In Genetic algorithm (GA), we not only need to choose the algorithm, representation and operators for the problem, but we also need to choose parameter values and operator probabilities so that it will find the solution efficiently. This process of finding appropriate parameter values and operator probabilities is a time consuming task and considerable effort has gone into automating this process. Literature contains various studies on setting of strategy parameters [1], [2], [3]. There are various types of Optimization functions as unimodal, multimodal, etc. and GA is used to optimize them.

In real-coded GA (RCGA) chromosome is a real-parameter decision-variable vector. The recombination operation is a method of sharing information among chromosomes [8][9].Traditional real-coded genetic algorithm (RCGA) has parameters that must be specified before the RCGA is run on a problem. The most important parameters are the strategy or control parameters: population size (N), mutation rate (pm), crossover rate (pc), and probability distribution index (η). [4][11]

Multi-modal functions have multiple optimum solutions, of which many are local optimal solutions. Multi-modality in a search and optimization problem usually causes difficulty to any optimization algorithm in terms of finding the global optimum solutions. Therefore, when dealing with multi-modal functions, some modification is necessary to the standard GAs to permit stable subpopulations at all peaks in the search space.

In this paper, we have described a framework of GA for multimodal optimization which use clustering algorithm to identify different region of attractors, then search each attractors independently. For generating subpopulations we have used two different clustering techniques K-means and Fuzzy C-means.

Paper is organized as: section 2 introduces multi-modal optimization & Literature for solving them. In section 3 we introduced Clustering, K-means and Fuzzy C-means clustering. In section 4 Proposed Algorithm i.e. Self Adaptive GA with Clustering is described. In section 5 experimental setup is discussed. Section 6 presents empirical results of Proposed Algorithm over few multi-modal test functions. Finally in section 6 we draw some conclusion.

II. INTRODUCTION TO MULTI-MODAL OPTIMIZATION & LITERATURE

As the name suggests, multi-modal functions have multiple optimum solutions, of which many are local optimal solutions. Multi-modality in a search and optimization problem usually causes difficulty to any optimization algorithm in terms of finding the global optimum solutions. This is because in these problems there exist many attractors for which finding a global optimum can become a challenge to any optimization algorithm. In the case of peaks of equal value (height), the convergence to every peak is desirable, whereas with peaks of unequal value, in addition to knowing the best solutions, one may also be interested in knowing other optimal solutions. Therefore, when dealing with multi-modal functions, some modification is necessary to the standard GAs to permit stable subpopulations at all peaks in the search space. [13]

If a classical point-by-point approach is used for this purpose, the method must have to be used many times, every time hoping to find one of the optima.

Various techniques are adopted to optimize multimodal functions. Genetic Algorithm is used in combination with niching techniques. A niching method must be able to form and maintain multiple, diverse, final solutions, whether these solutions are of identical fitness or different fitnesses. A niching method must be able to maintain these solutions for a large enough iterations.[13]

Petrowski suggested the clearing procedure, which is a niching method inspired by the principle of sharing of limited resources within subpopulations of individuals characterized by some similarities. A clustering algorithm [13][17] is used to divide the population into niches. Crowding methods insert new elements in the population by replacing similar elements.[18] Harik [19] introduced a modified tournament algorithm that exhibited niching capabilities.

Species conservation techniques are used to identify species within a population and to conserve the identified species in the current generation..[10]

Multimodal Functions have multiple local as well as global optimas. Any standard GA is found to be less efficient for the various multimodal as well as multimodal multipeak functions, because during execution when the algorithm finds first optima it get terminated with the single optima as result. But for Multimodal as well as multimodal Multipeak functions, the aim is to search all the optimas present in the given function. For this reason, the Initial Population must be divided into clusters. We are assuming, in each cluster we will find an optima. So we can reach to all the optimas.

So we have used K-means as well as Fuzzy C-means Clustering for dividing the Population into clusters.

III. Clustering K-Means & Fuzzy C-Means Clustering

Clustering is the unsupervised classification of patterns (or data items) into groups (or clusters). A resulting partition should possess the following properties: (1) homogeneity within the clusters, i.e. data that belong to the same cluster should be as similar as possible, and (2) heterogeneity between clusters. i.e. data that belong to different clusters should be as different as possible. Several algorithms require certain parameters for clustering, such as the number of clusters and cluster shapes.

Let us describe the clustering problem formally. Assume that S is the given data set:

$S = \left\{ \vec{x}_{1,} \ldots\ldots \vec{x}_n \right\}$ where $\vec{x}_i \in R^n$ .The goal of clustering is to find K clusters $C_{1,}$ $C_{2,\ldots\ldots}$ C_k such

that $C_i \neq \phi$ for i = 1, . . .K. (1)

$$C_i \cap C_j = \phi \text{ for i, j = 1, . . .K; } i \neq j \quad (2)$$

$$\bigcup_{i=1}^{K} C_i = S \quad (3)$$

and the objects belonging into same cluster are similar in the sense of the given metric, while the objects belonging into different clusters are dissimilar in the same sense.

In other words, we seek a function $f : S \rightarrow \{1,\ldots\ldots,K\}$

such that for $i = 1,\ldots..,K : C_i = f^{-1}(i)$ where C_i satisfies the above conditions. The Euclidean metric can be used to measure the cluster quality.

Then the function f is sought such that

$$f \;= \arg\min Evq(c_1,\ldots\ldots.,c_k) \quad (4)$$

$$= \arg\min_i \sum_{i=1}^{k} \| \vec{x}_i - \vec{c}_{f(\vec{x}_i)} \|^2$$

Where

$$\vec{c}_k = \frac{1}{|C_k|} \sum_{\vec{x} \in c_{k_i}} \vec{x}_i \qquad \text{k=1,…,K} \quad (5)$$

Therefore instead of function f directly, one can search for the centers of the clusters, i.e. vectors $\vec{c}_i,\ldots\ldots.,\vec{c}_k$.

implement the function f as

$$f(\vec{x}) = \arg\min_i \| \vec{x}_i - \vec{c}_i \|^2 \quad (6)$$

That is, assign the point to the cluster corresponding to the nearest centre.[21]

Finally, complete content and organizational editing before formatting. Please take note of the following items when proofreading spelling and grammar:

A. K-means algorithm

This algorithm is based on an adaptive approach where a random set of cluster base is selected from the original dataset, and each element update the nearest element of the base with average of its attributes.[7]

K-means is one of the simplest unsupervised learning algorithms that solve the well-known clustering problem. This algorithm aims at minimizing an objective function; in this case a squared error function.

The objective function.

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \| x_i^{(j)} - c_j \|^2 \quad (7)$$

where $\left\| x_i^{(j)} - c_j \right\|^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster centre c_j , is an indicator of the distance of the *n* data points from their respective cluster centers. The algorithm is composed of the following steps:

1. Place K points into the space represented by the objects that are being clustered. These points represent *initial group centroids.*
2. *Assign each object to the group that has the closest centroid.*
3. *When all objects have been assigned, recalculate the positions of the K centroids.*
4. *Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.*

B. Fuzzy C-means algorithm.

Fuzzy c-means (FCM) is a method of clustering which allows one piece of data to belong to two or more clusters. This method is frequently used in pattern recognition. It is based on minimization of the following objective function:

$$J_m = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^m \left\| x_i - c_j \right\|^2 \qquad 1 \le m < \infty \tag{8}$$

where m is any real number greater than 1, u_{ij} is the degree of membership of x_i in the cluster j, x_i is the ith of d-dimensional measured data, c_j is the d-dimension center of the cluster, and ||*|| is any norm expressing the similarity between any measured data and the center. The algorithm is composed of the following steps:

1. *Initialize U=[uij] matrix, U(0)*
2. *At k-step: calculate the centers vectors C(k)=[cj] with U(k)*

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^m \cdot x_i}{\sum_{i=1}^{N} u_{ij}^m} \tag{9}$$

3. *Update U(k) , U(k+1)*

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{\left\| x_i - c_j \right\|}{\left\| x_i - c_k \right\|} \right)^{\frac{2}{m-1}}} \tag{10}$$

4. *If || U(k+1) - U(k)||<* ε *then STOP; otherwise return to step 2.*

IV. Proposed Algorithm : Self Adaptive GA With Clustering

Real Coded Self Adaptive Genetic Algorithm proposed in [4], was used for Multimodal Function Optimization. The algorithm has given good performance on Multimodal Functions. But fails to solve Multimodal Multipeak Functions. Here in this work, we propose Self Adaptive Genetic Algorithm which makes use of Clustering Algorithm.

Self Adaptation is important feature of Proposed Algorithm. MPX and MLX are two Multi-parent Crossover Operators. The MPX is exploitative and MLX is explorative in nature, a mechanism of utilizing suitable one is required to achieve efficient search. The basic idea of mechanism is that the probabilities of applying each operator are adapted based on the performance of the offspring generated by operator.[15]

In Self Adaptive GA with Clustering, K-Mean as well as Fuzzy C-means clustering algorithm are used. Performance of proposed Algorithm with K-means and with Fuzzy C-means is analyzed. Clustering is used after Population Initialization. The number of Clusters is a parameter of the Algorithm.

A. Algorithm: Self Adaptive GA with Clustering

In the terms of algorithm-generator, Self Adaptive GA with Clustering can be described as follows:

Step 1:- Generate Initial Population randomly.

Step 2:- Divide the Initial Population into Clusters by using Clustering Algorithm.(K-means or Fuzzy C-means.)

Step 3:-In each Cluster we have to apply the following steps.

Step 4:- The best parent and other (μ-1) random solutions are picked from *B* to form set P.

Step 5:- Create λ offspring solutions (the set C) from *P* using MPX or MLX operator. Choose recombination operator according to their probability. This probability depends upon the good offspring produced by the respective operator and is updated after every fixed number of function evaluations.

Step 6:- Replacement Plan: form a family of λ and μ solutions (the Set *R*).

Step 7:- Update Plan: From the *R*, choose *r* best solutions and put them in *R'*. Update *B* as

B := (*B**R*) U R'

Step 8:-Repeat the Steps 4 to 8 until Stopping Criteria met.

Step 9- Find best solution in each cluster.

Here set B denotes the population of solutions of size N. Operator adaptation methods is implemented as follows

1. Each member of population contains information about the operator (1 for MPX, 2 for MLX and 0 initially) along with decision variables and objectives. This information is placed into chromosomes at the time of generation.

2. In order to implement a 'learning-rule' to adapt operator probabilities two global counters; one for MPX operator and other for MLX operator are maintained. Increment these counters when offspring replaces parent. Operator is rewarded if offspring produced by it replaces best parent. Probability of operator is calculated as

$$P_{MPX} = (C_{MPX} / C_{MPX} + C_{MLX})$$
$$P_{MLX} = 1 - P_{MPX} \qquad (11)$$

Where P_{MPX} and P_{MLX} are probabilities of MPX and MLX respectively. C_{MPX} and C_{MLX} are global counters for MPX and MLX respectively. We limit the range of P_{MPX} to 0.05 to 0.95.

V. Experimental Setup

TABLE I. GA SETUP USED FOR EXPERIMENTATION

GA type	**Steady-state GA**
Population size (N)	150
Crossover probability parameter (p_c)	0.6 - 0.7 in step of 0.1
Distribution index (η)	For MPX=2 For MLX=4
Number of parents (μ)	10
Stopping criteria	Maximum 10^6 function evaluations or an objective value of Optima
Results average over	45 independent runs
Parameters for performance evaluation	1. Number of function evaluation for best run and worst run 2. Average number of function evaluation 3. Best fitness, Average fitness and Worst fitness 4. Number of runs converged to global Optima.
Number of children (λ)	2

When dealing with multi-modal functions, some modification is necessary to permit stable subpopulations at all peaks in the search space. So we have used two different techniques of Clustering K-means and Fuzzy C-means for dividing Population into subpopulations. Since we are comparing Self Adaptive GA with K-means Clustering and Self Adaptive GA with Fuzzy C-means Clustering it is mandatory to use the same setup for both the algorithms which is specified in table 1.

Optimization Functions f1, f2, f3, f4 and f5 are Minimization functions f6, f7 and f8 are Maximization functions.

VI. Results

We tested Self Adaptive GA with Clustering on some multimodal functions as listed in table 2. Results are shown in the table 3.Results obtained with RAGA in [4] have shown that, it has performed well for the Multimodal functions with single global peak i.e. functions f1 and f3. Algorithm searches global optima. But fails to solve Multimodal Multipeak Functions. Only single optima can be searched.

So clustering is introduced in the proposed Algorithm for forming subpopulations. Self Adaptive GA with KM have used K-means for Cluster formation. Each individual Cluster has evaluated to search the optima. Algorithm has given better performance while solving some multimodal multipeak functions with less number of Average Function evaluations. Figures 1 & 2 shows the Proper clusters formed by Self Adaptive GA with K-means. Each cluster contains a single optima. Figure 1(a) shows Initial Population, Figure 1(b) shows Clustered Population Figure 1(c) Optimas found for Five-Uneven-Peak Trap. Figure 2(a) shows Initial Population, Figure 2(b) shows Clustered Population Figure 2(c) Optimas found for Himmelblau's function. But for some multimodal multipeak functions all optimas have not searched due to improper clustering as illustrated in figure 3(b).

To overcome this difficulty we have replaced K-means by Fuzzy C-means for Cluster formation. Self Adaptive GA with FCM has given better performance for all the functions listed in table 2 with less number of Average function evaluations. Using Fuzzy C-means, Proper clusters have been formed as illustrated in figures 3(c).

VII. Conclusion

In this paper, Self Adaptive GA with clustering is proposed. Algorithm employs both MPX and MLX operators. To choose crossover operator it has self-adaptive mechanism. It also uses replace worst parent with best offspring strategy in update plan. Multimodal Functions have multiple local as well as global optimas. When dealing with multi-modal functions, some modification is necessary to the standard GAs to permit stable subpopulations at all peaks in the search space. For this reason, we have used K-means or Fuzzy C-means Clustering algorithm for creating subpopulations.

Self Adaptive GA with K-means performed well for functions having less than five global optimas. But Self Adaptive GA with Fuzzy C-means performed better for all functions stated in table 2. The average function evaluations for Self Adaptive GA with Fuzzy C-means are less as compared with Self Adaptive GA with K-means. Comparative results are shown in table 3.

Future research will follow two parallel directions. The first one will extend the study of the current algorithm on real world problems. The second one will experiment on the application of other clustering algorithms, especially ones which do not require a priori knowledge of the number of clusters.

TABLE 2 MULTIMODAL TEST FUNCTIONS

Name	Test Function	Range	No. of Global Peaks
Michalewicz's function [16]	$f_1(x) = -\sum_{i=1}^{n} \sin(x_i)\cdot\left(\sin\left(\frac{i.x_i^2}{\Pi}\right)\right)^{2\cdot m}$	$0 \le x_i \le \prod$	1
Branins's rcos function [16]	$f_2(x_1,x_2) = a\cdot(x_2 - b\cdot x_1^2 + c\cdot x_1 - d)^2 + e\cdot(1-f)\cdot\cos(x_1) + e$	$-5 \le x_1 \le 10$ $0 \le x_2 \le 15$	3
Goldstein-Price's function [16]	$f_3(x_1,x_2) = (1+(x_1+x_2+1)^2\cdot(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2))$ $\cdot(30+(2x_1-3x_2)^2\cdot(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2))$	$-2 \le x_1x_2 \le 2$	1
Six-hump camel back function[16]	$f_4(x_1,x_2) = (4-2.1x_1^2 + x_1^{4/3})\cdot x_1^2 + x_1x_2 + (-4+4x_2^2)\cdot x_2^2$	$-3 \le x_1 \le 3$ $-2 \le x_2 \le 2$	2
Himmelblau's function [20]	$f_5(x_1,x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2$	$-6 \le x_1x_2 \le 6$	4
Equal Maxima[20]	$f_6(x) = \sin^6(5\prod x)$	$0 \le x \le 1$	5
Uneven Maxima[20]	$f_7(x) = \sin^6(5\prod(x^{\frac{3}{4}} - 0.05))$	$0 \le x \le 1$	5
Five-Uneven-Peak Trap [20]	$f_8(x) = \begin{cases} 80(2.5-x) \text{ for } 0 \le x < 2.5 \\ 64(x-2.5) \text{ for } 2.5 \le x < 5.0 \\ 64(7.5-x) \text{ for } 5.0 \le x < 7.5 \\ 28(x-7.5) \text{ for } 7.5 \le x < 12.5 \\ 28(17.5-x) \text{ for } 12.5 \le x < 17.5 \\ 32(x-17.5) \text{ for } 17.5 \le x < 22.5 \\ 32(27.5-x) \text{ for } 22.5 \le x < 27.5 \\ 80(x-27.5) \text{ for } 27.5 \le x \le 30 \end{cases}$	$0 \le x \le 30$	2

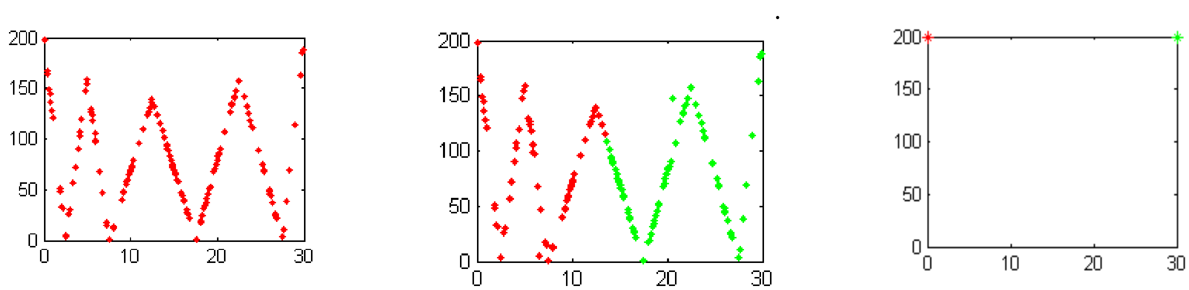

Figure 1.a) Initial Population b) Clustered Population using K-means c) Optima's found for Five-Uneven-Peak Trap

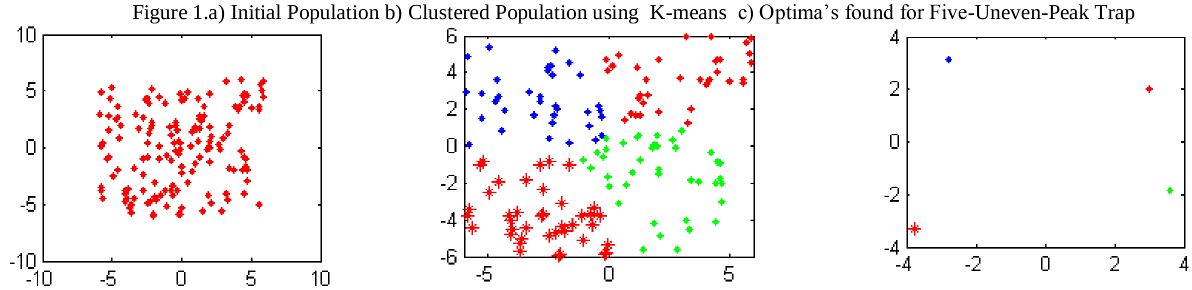

Figure 2.a) Initial Population b) Clustered Population using K-means c) Optima's found for Himmelblau's function

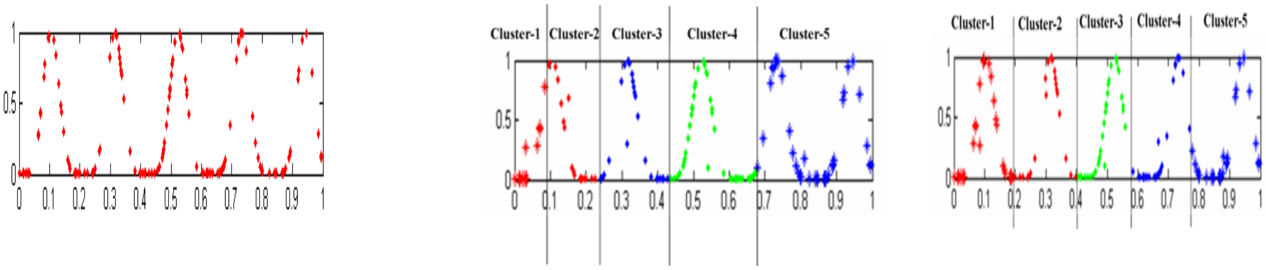

Figure 3.a) Initial Population b) Clustered Population using K-means c) Clustered Population using Fuzzy C-means for Equal Maxima Function.

TABLE 3 EXPERIMENTAL RESULTS OF RAGA WITH CLUSTERING

Sr.No.	**Function Name**	**Algorithm**	**No. of Function Evaluations**			**Fitness**			**Success**
			Best	**Avg**	**Worst**	**Best**	**Avg**	**Worst**	
1	f1 Michalewicz's function	RAGA	22226	107792	607670	-9.782	-9.68381	-9.66014	(45/45)
		SAGAKM	17044	157885	1000002	-9.76158	-9.68403	-9.6523	(41/45)
		SAGAFCM	20344	327997	1000002	-9.65959	-9.65091	-9.58876	(33/45)
2	F2 Branins's rcos function	RAGA	2094	3022.49	4358	0	0	0	(45/45)
		SAGAKM	560	1272.42	11048	0	0	0	(45/45)
		SAGAFCM	766	1224.82	2644	0	0	0	(45/45)
3	f3 Goldstein-Price's function	RAGA	2694	446178	1000002	3	34.2	84	(25/45)
		SAGAKM	2422	446169	1000002	3	31.8	84	(25/45)
		SAGAFCM	2590	446175	1000002	3	34.2	84	(25/45)
4	F4 Six-hump camel back function	RAGA	268	589.244	988	-1.03163	-1.03161	-1.0316	(45/45)
		SAGAKM	142	22598.4	1000002	-1.03163	-1.03162	-0.21546	(45/45)
		SAGAFCM	164	11528.8	1000002	-1.03163	-1.03162	-0.21546	(45/45)
5	f5 Himmelblau’s function	RAGA	3110	3661.11	4250	8.72E-24	4.95E-21	9.89E-21	(45/45)
		SAGAKM	428	212338	1000002	1.94E-23	3.23E-21	16228.8	(45/45)
		SAGAFCM	734	1157.89	1874	4.59E-23	2.59E-21	9.99E-21	(45/45)
6	f6 Equal Maxima	RAGA	594	34285.9	1000002	1	1	1	(44/45)
		SAGAKM	78	10113	872798	1	1	1	(45/45)
		SAGAFCM	40	6327.58	964032	1	1	1	(45/45)
7	F7 Uneven Maxima	RAGA	212	710.133	1008	1	1	1	(45/45)
		SAGAKM	54	1263.51	101348	1	1	1	(45/45)
		SAGAFCM	30	345.511	10458	1	1	1	(45/45)
8	f8 Five-Uneven-Peak Trap	RAGA	4	5.95556	10	200	200	200	(45/45)
		SAGAKM	730	34537.9	1000002	200	200	160	(45/45)
		SAGAFCM	838	34520.2	1000002	200	200	160	(45/45)

REFERENCES

[1] Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms, John Wiley & Sons, New York. (2001)

[2] Deb, K., Agrawal, S.: Understanding interactions among genetic algorithm parameters. Foundations of Genetic Algorithms V. Morgan Kaufmann(1999)

[3] Smith, J.E., Fogarty, T.C.: Operator and Parameter Adaptation in Genetic Algorithms. Soft Computing 1(2) (1997) 81–87

[4] M. M. Raghuwanshi and O. G. Kakde, "Real-Code Self-Adaptive Genetic Algorithm (RAGA)" Second Indian International Conference on Artificial Intelligence (IICAI-2005) 20-22 Dec. 2005 pp 3291-3302

[5] D. E. Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning", Pearson Education Asia, 1989.

[6] M. M. Raghuwanshi and O. G. Kakde, "Multi-parent Recombination operator with Polynomial or Lognormal Distribution for Real Coded Genetic Algorithm" *2nd Indian International Conference on Artificial Intelligence (IICAI)*, pp. 3274-3290, 2005.

[7] Rahila Patel,M.M.Raghuwanshi,Anil N. Jaiswal "Modifying Genetic Algorithm with Species and Sexual Selection by using K-means Algorithm" 2009 IEEE International.Advance Comnputing Conference (LCC 2009) Patiala, India 6-7 March 2009

[8] Qing Lu, Changyong Liang, Enqiao Zhang " Dynamic Sharing Scheme-based Multimodal Niche Genetic Algorithm" Proceedings of the 7th World Congress on Intelligent Control and Automation June 25 - 27, 2008, Chongqing, China

[9] Carlos M. Fonsecay and Peter J. Flemingz "Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization"

[10] Jian-Ping Li, Xiao-Dong Li and Alastair Wood "Species Based Evolutionary Algorithms for Multimodal Optimization: A Brief Review" WCCI 2010 IEEE World Congress on Computational Intelligence July, 18-23, 2010 - CCIB, Barcelona, Spain.

[11] Hartmut Pohlheim "GEATbx Introduction Evolutionary Algorithms: Overview, Methods and perators" (November 2005).

[12] Jingjun Zhang Yanmin Shang Ruizhen Gao Yuzhen Dong "An Improved Genetic Algorithm Based on K1 Triangulation with Variable Coefficient for Optimization of Multimodal Functions" ICIEA 2009

[13] Dr. K. Deb, Gulshan Singh: Comparison of multimodal optimization algorithms based on evolutionary algorithm. GECCO'06, Seattle, Washington USA.

[14] Deb, K., Agrawal, R.B.: Simulated binary crossover for continuous search space. Complex System 9 (1995) 115-148

[15] Raghuwanshi, M.M., Singru, P.M., Kale, U., Kakde, O.G.: Simulated Binary Crossover with Lognormal Distribution. In Proceedings of the 7th Asia-Pacific Conference on Complex Systems (Complex 2004)(2004)

[16] Hartmut Pohlheim: GEATbx: Example Functions (single and multi-objective functions) 2 Parametric Optimization.

[17] Rahila H. Sheikh, M. M.Raghuwanshi, Anil N. Jaiswal "Genetic Algorithm Based Clustering: A Survey" First International Conference on Emerging Trends in Engineering and Technology pp 314-319, 2008 IEEE

[18] Bruno Sareni and Laurent Kr¨ahenb¨uhl "Fitness Sharing and Niching Methods Revisited" IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 2, NO. 3, SEPTEMBER 1998-97

[19] G. Harick. Finding multi-modal solutions using restricted tournament selection. In *Proceedings of the Sixth International Conference on Genetic Algorithms(ICGA-95)*, pages 24–31, 1997.

[20] Xiaodong Li, *Senior Member, IEEE* "Niching Without Niching Parameters: Particle Swarm Optimization Using a Ring Topology" IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL 14, NO. 1, FEBRUARY 2010, pp 150-169

[21] Petra Kudov'a ,"Clustering Genetic Algorithm," IEEE, DOI 10.1 109/DEXA.2007.65, 2007