Journal of Information Hiding and Multimedia Signal Processing ©2011 ISSN 2073-4212
Ubiquitous International Volume **2**, Number **4**, October **2011**

Genetic Algorithm with Histogram Construction Technique

Rahila Patel

Research scholar, G.H.R.C.E., nagpur (m.s.), India
rahila.patel@gmail.com

M.M.Raghuwanshi

NYSS College of Engg. and Research, Nagpur (m.s.), India
m_raghuwanshi@rediffmail.com

Urmila N Shrawankar

Deptt. Of CSE,GHRCE, Nagpur (m.s.), India
urmilas@rediffmail.com

Received April 2010; revised June 2011

Abstract. *This paper proposes a new niching technique for Distributed Genetic Algorithm (DGA) . In DGA quick evolution takes place because of independent evolution of subpopulation. To improve performance of DGA care need to be taken for distribution of population in to subpopulations. A proper distribution leads to efficient evaluation and good solutions. The new technique uses the traditional histogram construction technique for subpopulation formation. Experiments show that the proposed algorithm, GAHisto (Genetic Algorithm with Histogram technique) when tested with unimodal and multimodal test function has given good results. This paper is submitted for the special issue on Novel Intelligent Approach to Multimedia Signal Processing.*

Keywords: Distributed Genetic Algorithm (DGA), clustering algorithms, optimization, niching Genetic Algorithm, histogram construction

1. **Introduction.** Genetic algorithms are robust search and optimization tools and are widely applied to challenging engineering problems today. Genetic algorithm also needs evolution because of some of its undesirable properties. The first one is the lack of a quality guarantee of genetic search. For example, genetic algorithms are usually sensitive to the population size in terms of their search capability. It is difficult to estimate the required population size. Too large a population size leads to low efficiency, and one that is too small may simply fail to achieve satisfactory results. The second undesirable property is that once a genetic algorithm stagnates during a search, it usually loses most of its search capability, and there is no good way to rejuvenate the run in an efficient manner. Simple restart or strong mutations may waste the computations spent before by destroying the building blocks in the population. Weak mutations may perturb the solutions a little bit, but they cannot incur significant move in search space once the framework of the individual is established. The third problem of current genetic algorithms is the lack of robustness such as large variation of the performance of several runs due to the opportunistic and convergent nature of current genetic algorithms.

Distributed GAs (DGAs) is one of the most important representatives of methods based on spatial separation and tries to overcome some of the problems that exits with sequential GA. The basic idea of DGAs lies in the partition of the population into several subpopulations whose sizes are relatively small, each one of them being processed by a GA, independently from the others. Furthermore, an operator, called migration, produces a chromosome exchange between the sub-populations. Its principal role is to promote genetic diversity, and in essence, to allow the sharing of possible solutions. DGAs show two determinant advantages: 1) the preservation of the diversity due to the semi-isolation of the subpopulations, which may stand up to the premature convergence problem, and 2) they may be easily implemented on parallel hardware, obtaining, in this way, substantial improvements on computational time. [9][11]

In the past few decades, many niching techniques have been proposed, which have greatly improved the scalability and robustness of genetic search for difficult multi-modal functions. However, due to the convergent nature of the current genetic algorithm framework, these niching approaches still meet difficulty in many hard problems.

In this paper we propose a new distributed genetic algorithm GAHisto (Genetic Algorithm with Histogram construction technique), which takes care of population distribution. The algorithm makes use of histogram construction technique i.e. it forms initial clusters of population by counting all the nearest individuals of an individual. It uses four plans of co-evolution there by maintaining diversity in population and avoids premature convergence of the algorithm. This algorithm has given good response to multi-peak multimodal optimization functions.

The rest of the paper is organized as follows. In section 2, existing commonly used niching methods will be discussed. In section 3 new niching technique and the proposed algorithm will be discussed. Section 4 discusses experimental setup used during experimentation. Section 5 compares results of proposed algorithm with GAS3 algorithm. Finally some conclusion has been drawn in 6 and some comments on future work.

2. **Related Work.** Niching methods have been developed to reduce the effect of genetic drift resulting from the selection operator in the standard GA. They maintain population diversity and permit the GA to investigate many peaks in parallel. On the other hand, they prevent the GA from being trapped in local optima of the search space. Niching GAs are based on the mechanics of natural ecosystems. In nature, animals compete to survive by hunting, feeding, grazing, breeding, etc., and different species evolve to fill each role. A niche can be viewed as a subspace in the environment that can support different types of life. A species is defined as a group of individuals with similar biological features capable of interbreeding among themselves but that are unable to breed with individuals outside their group. For each niche, the physical resources are finite and must be shared among the population of that niche. By analogy, niching methods tend to achieve a natural emergence of niches and species in the environment (search space). A niche is commonly referred to as an optimum of the domain, the fitness representing the resources of that niche. Species can be defined as similar individuals in terms of similarity metrics.

Niching is the name given to modifications to the genetic algorithm, which encourages the formation of niche populations, or species, representing different, locally optimized solutions. Several strategies exist to facilitate the formation of niches within a population, most notably crowding, mate selection, explicit fitness sharing and clearing.

Crowding V When the crowding method is used to promote offspring, when each new solution is created, it is compared to similar solutions present in the population, and probabilistically replaces one with a lower fitness. A variation on this method is deterministic crowding, where the new solution replaces one of the parent solutions.

Mate Selection V This method is based on the observation that in nature, organisms do not choose their mates randomly, but instead favor other individuals similar to them, so as not to propagate rival genes. In the genetic algorithm, selection of pairs of solutions to undergo crossover is biased so that solutions prefer to crossover with similar solutions. In this way, crossover can be used productively in the later stages of the genetic algorithm to converge niche populations to local optima without the overhead of producing solutions combining parents belonging to different niches.

Explicit Fitness Sharing V This method works by reducing the fitness of solutions as a function of the number of similar solutions currently represented in the population (i.e. belonging to the same niche). [2]

Clearing: The clearing method is very similar to fitness sharing but is based on the concept of limited resources of the environment. Instead of sharing the resources between all individuals of a single subpopulation as in fitness sharing, clearing attributes them only to the best members of the subpopulation. In practice, the capacity of a niche specifies the maximum number of elements that this niche can accept. Thus, clearing preserves the fitness of the best individuals (dominant individuals) of the niche and resets the fitness of the others that belong to the same subpopulation (dominated individuals). As in the sharing method, individuals belong to the same niche (or subpopulation) if their distance in the search space is less than a dissimilarity threshold (clearing radius). Clearing can be coupled with elitism strategies to preserve the best elements of the niches during the generations. The order of complexity of the basic clearing procedure is where is the number of niches maintained during the search. [7]

Paper [10] presents genetic algorithm with species and Sexual Selection (GAS3). GAS3 uses sex determination method, which divides the population in to males and females. Species formation takes place around females. Sexual selection is used and interspecies mating is restricted. Merging of subpopulations takes place in the later part.

In each species offspring is generated in every iteration and updates sub-population.

The pseudo-code for GAS3 is as follows

```
1. Create initial population
2. Use sex determination method to determine the sex of an individual.
3. Species evolution phase creates many species around niches
4. Repeat
   {
   4.1 For certain number of evolution with each species in population do
       {
           4.1.1 Selection Plan: Choose one female parent and g male parents from
                 species S (the set M).
           4.1.2 Generation Plan: Create the offspring set C from M
           4.1.3 Replacement Plan: Combine solutions in C and M to form set
                 R. Arrange members in R according to their fitness.
           4.1.4 Update Plan: if one of the offspring is better than the female parent
                 then replace that female parent with offspring. Otherwise compare
                 offspring with male
                 members and use replace worst strategy for updating.
           4.1.5 Increment the performance count of species if offspring replaces female
                 parent.
       }
    4.2 Examine the performance of each species and merge species.
    } Until (termination condition)
```

This is a generalized framework for GAS3, where one can use his own sex determination method, species evolution scheme, and four plans for evolution and scheme for merging of species. More detailed and updated information about niching methods can be found in [2]-[8].

Drawback of GAS3 is that initial female selection does not guarantee that they are well distributed or well separated. GAS3KM [13] is an extension to GAS3 algorithm. In GAS3KM, K-means clustering algorithm is applied on each cluster to move female member at the centre of cluster. Considering all the pros and cons of K-means, it is used after species formation phase. K-means require initial cluster centres as input. All the females are considered as initial cluster centres. To overcome problem associated with random choice of initial cluster centres, fertile females are selected as initial cluster centres. If data set is large then it is very difficult to guess proper number of clusters a priori. This problem also tackled by providing number of females as number of clusters. The test results of GAS3KM for unimodal and multimodal test function show that it outperforms GAS3 algorithm in terms of number of function evaluation. K-means algorithm helps in producing well-distributed and well-separated clusters. These clusters evolve more quickly and require less number of function evaluations for finding optima. K-means also inject diversity in female population. K-means used with Sex Determination Method and Species Evolution Phase, behaves like self-adaptive algorithm, which automatically detects number of clusters and initial position of centroids.

3. **Proposed Algorithm.** This paper proposes an improved genetic algorithm GAHisto (Genetic Algorithm with Histogram construction technique). It makes use of traditional histogram construction technique for initial cluster formation of population. It simply counts nearest individuals of an individual in the population and forms a cluster.

A histogram is a bar graph that charts the relationship between value and frequency (number of occurrences). The word histogram is derived from the Greek histos 'anything set upright' (as the masts of a ship, the bar of a loom, or the vertical bars of a histogram); and gramma 'drawing, record, writing'. In statistics a histogram is a graphical display of tabulated frequencies, shown as bars. Histograms are typically used to approximate data distributions. Histograms and related synopsis structures have been successful in a wide variety of popular database applications including approximate querying, similarity searching and data mining. Histograms were a few of the earliest synopsis structures proposed and continue to be popular tools. Typically, the histograms are used as quick and easy estimates, and thus fast histogram construction algorithms can offset the slight loss of accuracy.

Let X = x1, $\cdots$, xn be a finite data sequence. The general problem of histogram construction is as follows: given some space constraint B, create and store a compact representation HB of the data sequence. HB uses at most B storage and is optimal under some notion of error. The representation collapses the values in a sequence of consecutive points xi where i belongs to [sr; er] (say sr < i < er) into a single value hr, thus forming a bucket br, that is, br = (sr; er; hr). Each bucket is represented by a single value. Thus a histogram defines a piecewise constant approximation of the data. The histogram HB is used to answer queries about the value at point i where 1 < i < n. The histogram uses at most B buckets which cover the entire interval [1; n], and saves space by storing only O(B) numbers instead of n. [1]

In this algorithm histogram technique counts niches in the population and clusters are formed around these niches. Each cluster contains at least two members. No cluster with single member or zero members is allowed, there by avoiding problem of outliers and empty clusters. The parent centric self-adaptive multi-parent recombination operators are used to explore the search space. Four plans of co evolution are applied on these clusters. The proposed algorithm is given below.

```
GENETIC ALGORITHM WITH HISTOGRAM CONSTRUCTION
  1. Initialize population.
  2. Formation of cluster by using histogram construction technique.
  3. For certain number of evolution with each cluster in population does
     {
     3.1 Selection Plan: Choose one cluster center parent and g parents from cluster S
         (the set M).
     3.2 Generation Plan: Create the offspring set C from M.
     3.3 Replacement Plan: Combine solutions in C and M to form set R. Arrange
          members in R according to their fitness.
     3.4 Update Plan: if one of the offspring is better than the parent then replace that
         parent with offspring.
     } Until (termination condition)
  The algorithm for cluster formation is given below.
```

```
CLUSTER FORMATION USING HISTOGRAM CONSTRUCTION TECHNIQUE

  1.Select one individual from population
  2.Calculate distance between the selected individual and the rest of the individuals in
    the population.
  3.The individual with minimum distance will be the Nearest Neighbour (NN) of the
    selected individual.
  4.Repeat step 1-3 for all individuals in the population.
  5.Form cluster of individual having common NN i.e. all the individuals near to one
individual (NN) form a cluster.
```

In general, sampling of more information from a population helps evolution process to bring better changes in the next generation. The studies on multi-parent recombination operators have given sufficient indication that they can enhance performance of GAs.

This algorithm makes use of Multi-parent Parent-Centric Crossover Operator (PCCO). One such operator is MPX (Multi-Parent Polynomial Distribution Crossover Operator) [12]. The operators based on polynomial distribution are more exploitative and exploitation range decreases with increase in distribution index of probability distribution (b). The MPX operator is a multi-parent extension of the SBX operator. The prototype algorithm for the MPX operator is as follows:

a) From population select best parent and pick other (g-1) solutions randomly.

b) For each gene (i=1,n) in real-parameter chromosome execute following steps

The fitness value of the evenly produced offspring is compared with the fitness value of its parents in order to decide whether or not the evenly produced offspring is accepted as a member of the next generation. The offspring is accepted as a candidate for the further evolutionary process if and only if the reproduction operator was able to produce an offspring that could outperform the fitness of its parents. This strategy guarantees that evolution is presumed mainly with crossover results that were able to mix the properties of their parents in an advantageous way. Hence, via these means we are in a position to attack one of the reasons for a premature convergence, namely the loss of relevant genetic information due to improper crossover operation. Furthermore, this strategy has proven to act as a precise mechanism for self-adaptive selection pressure steering. An offspring is

i) Choose u_i randomly from the interval [0, 1].

ii) Compute β_i using (4.1).

$$\beta_i = \begin{cases} (2u_i)^{1/(\eta+1)} & \text{if } u_i \le 0.5 \\ \left[1/(2(1-u_i))\right]^{1/(\eta+1)} & \text{Otherwise} \end{cases} \tag{4.1}$$

iii) Calculate

$$D = \left(\sum_{k=1}^{\mu} \left(\sum_{j=1}^{\mu} | x_i^j - x_i^k | \right) / \mu \right) / \mu \tag{4.2}$$

iv) Generate two genes around gene of best parent (say x^1) using (4.3)

$$y_i = x_{i1}^1 \pm (\beta_i * D) \tag{4.3}$$

better than the parents, if it surpasses the fitness of the parent in mating pool. In update plan, it replaces the parent whose fitness is just less than offspring.

4. **Experimental Setup.** The experimental setup used is given in Table 1. GAHisto algorithm uses histogram construction technique for cluster formation. Each cluster contains at least two members. Four plans of co evolution are applied on these clusters. For the experimentation, all the nearest members form a cluster and each cluster under go a sequence of operations like selection, crossover and mutation. Random selection has been used for selection of cluster members for recombination.

Table 1. GA Setup Used For Experimentation

Parameter	Values
Population size (N)	100
Crossover probability parameter (p_c)	0.3 - 0.8 in step of 0.1
Stopping criteria	Maximum 10^6 function evaluations or an objective value of 10^{-10}
Results average over	50 independent runs
Parameters for performance evaluation	1.Number of function evaluation for best run and worst run 2.Average number of function evaluation (AFES) 3.Best fitness, Average fitness and Worst fitness 4.Number of runs converged to global minima
Initialization of variables	Skewed initialization within [-10, -5]
Number of children (λ)	2

Multi-parent Parent-Centric Crossover Operator (PCCO) MPX (with $\eta = 1$) used with five parents (μ =5) to generate two offspring has been used for crossover. Since this is a real-coded genetic algorithm, real mutation operator has been used for mutation of offspring produced by crossover operator. If offspring generated is better than parents then the offspring replaces the parent. Replace-worst strategy has been used for parent replacement. All the clusters evolve independently and generate optima. 50 independent runs are executed for all possible combination of parameters N and pc.

4.1. **Test functions.** The minimization experiments will be performed on unconstrained unimodal and multi-modal functions with or without epistasis among n-variables as shown in appendix I. Using skewed initialization these functions will be evaluated for global minima at 0.

5. **Discussion on Results.** GAHisto algorithm is tested on different unimodal and multimodal test functions. Best Run, Average Run, Worst Run, Best Fitness, Average Fitness, Worst Fitness and success of GAHisto for 50 independent run is shown in Table 2. The experimental results of GAHisto are good on unimodal and multi-modal test function. GAHisto has given 100% success for functions like ellipsoidal, schwefel, cigar, sphere and Ackelys. Table 3 shows performance comparison of GAHisto with GAS3 and GAS3KM algorithm. For population size =100 and pc = 0.5 GAHisto tested on Unimodal Test Function. AFES (Average number of Function EvaluationS) for all unimodal function has reduced to a greater extent i.e. it is more or less 50% as compared to Average number of Function EvaluationS required with GAS3 and GAS3KM algorithm. GAHisto has given 100% success in solving all unimodal functions.

Table 2. Performance of GAHisto on unimodal and multimodal test functions (population size =100)

Function	pcross	BestRun	AvgRun	WrstRun	BestFit	AvgFit	WrstFit	Success
Ellipsoidal	0.3	4061	5592.33	8085	5.80E-11	8.43E-11	9.79E-11	50/50
	0.4	4045	5814.11	8937	6.71E-11	8.41E-11	9.91E-11	50/50
	0.5	4145	5842.78	8793	5.65E-11	8.11E-11	9.99E-11	50/50
Schwefel	0.3	13853	24385	42665	7.65E-11	9.42E-11	1.00E-10	50/50
	0.4	12669	20593.2	30629	7.60E-11	9.03E-11	9.93E-11	50/50
	0.5	15297	21414.3	34441	7.26E-11	9.37E-11	9.99E-11	50/50
Sphere	0.3	3805	5445.67	8713	6.74E-11	8.65E-11	9.94E-11	50/50
	0.4	3801	5615.67	8157	5.66E-11	8.32E-11	9.55E-11	50/50
	0.5	3653	5721.44	8525	4.38E-11	7.66E-11	9.90E-11	50/50
Tablet	0.3	3669	5420.56	8489	5.19E-11	8.06E-11	9.91E-11	50/50
	0.4	3753	5641.89	8677	3.05E-11	7.78E-11	9.99E-11	50/50
	0.5	3749	5721.44	8345	3.62E-11	7.51E-11	9.99E-11	50/50
Cigar	0.3	4593	6286.33	8941	4.95E-11	8.18E-11	9.78E-11	50/50
	0.4	5033	6687.22	9509	6.47E-11	8.54E-11	9.99E-11	50/50
	0.5	4689	6700.33	8797	3.05E-11	7.61E-11	9.44E-11	50/50
Rastrigin	0.6	6405	198172	1000001	3.26E-11	0.608066	10.9452	46/50
	0.7	14505	189966	697269	1.45E-12	5.94E-11	9.80E-11	50/50
	0.8	9641	297417	1000001	1.52E-11	1.49253	17.9103	45/50
Griewangk	0.6	4177	281013	1000001	3.66E-11	0.002734	0.02459	43/50
	0.7	4549	398803	1000001	7.84E-12	0.001916	0.014772	43/50
	0.8	4145	286717	1000001	4.21E-11	0.003006	0.027017	43/50
Ackleys	0.6	5897	7722.11	10437	7.28E-11	9.12E-11	9.90E-11	50/50
	0.7	6041	7807.44	10337	8.04E-11	9.30E-11	9.92E-11	50/50
	0.8	6141	8131.22	11265	6.54E-11	9.08E-11	1.00E-11	50/50

Table 4 shows performance comparison of GAHisto with GAS3 and GAS3KM algorithm on multimodal test function for population size = 100 and probability of crossover pc = 0.6. The Ackleys function has an exponential term that covers its surface with numerous local minima. The complexity of this function is moderate. An algorithm that only uses the gradient steepest descent will be trapped in local optima, but any search strategy that analyzes a wider region will be able to cross the valley among the optima and achieve better results. GAHisto has given much better results for the Ackelys function.

The Rastrigins function has a contour made-up by a large number of local minima whose value increases with the distance to the global minimum. It is a parabolic function with a superposed cosine term of high amplitude. The number of local optima in the search region is approximately 10n. GAHistos performance when tested on the Rastrigins function was found to be moderate as compared to GAS3.

Table 3. Performance comparison of GAHisto with GAS3KM and GAS3 Algorithm with population size =100 and probability of crossover = 0. 5 on Unimodal Test functions

Function	**GAS3**			**GAS3KM**			**GAHisto**		
	AFES	Fitness	Success	AFES	Fitness	Success	AFES	Fitness	Success
Ellipsoidal	7331.84	2.57E-11	100%	7363.39	6.59E-11	100%	**5842.78**	5.65E-11	100%
Schwefel	36570.7	4.68E-11	100%	57777.1	4.50E-11	100%	**21414.3**	7.26E-11	100%
Sphere	7786.53	4.53E-11	100%	7449.5	5.70E-11	100%	**5627.67**	4.38E-11	100%
Tablet	13251.4	6.73E-11	100%	5600.7	6.04E-11	100%	**5721.44**	3.62E-11	100%
Cigar	11630.8	5.15E-11	100%	14282.4	7.13E-11	100%	**6700.33**	3.05E-11	100%

Table 4. Performance comparison of GAHisto with GAS3KM and GAS3 Algorithm with population size =100 and probability of crossover = 0. 6 on Multimodal Test functions

Function	**GAS3**			**GAS3KM**			**GAHisto**		
	AFES	Fitness	Succes	AFES	Fitness	Succes	AFES	Fitness	Succes
Rastrigin	187978	1.18E-11	100%	135082	7.64E-11	100%	**189966**	1.45E-12	100%
Griewangk	46323.2	3.78E-11	100%	45476.6	8.55E-11	100%	**281013**	3.66E-11	83.33 %
Ackleys	62702.8	5.92E-11	100%	50945.6	9.31E-11	100%	**7722.11**	7.28E-11	100%

The Griewangks function has a product term that introduces strong interdependence among the variables. This function is difficult to optimize because of its non-reparability and a search algorithm has to climb a hill to reach the next valley. Nevertheless, one undesirable property exhibited is that it becomes easier as the dimensionality is increased. GAHisto has not given desired results with the Griewangks function.

6. **Conclusion.** The work has proposed an improved distributed genetic algorithm with the new technique of niche formation. The proposed algorithm GAHisto, in a unique way used traditional histogram construction technique for formation of niches in population. Distribution of population into subpopulations by using this technique results in proper distribution there by enhancing performance of the algorithm. The test problems unimodal and multimodal, which simulate the various difficulty levels of real-world optimization problems, have been solved successfully. When compared with GAS3 algorithm it is found that the algorithm GAHisto has given better performance in most of the test function. The study has attempted to device a new way for population distribution in a distributed genetic algorithm.

Future study will be on different strategies for selection of members of cluster for crossover. Further GAHisto can be improved if cluster validity index like DB will be used. Also Merging /Migration schemes can be applied. Further we can solve multiobjective optimisation problems by modifying this algorithm.

REFERENCES

[1] Sudipto Guha, Nick Koudas, Kyuseok Shim, Approximation and streaming algorithms for histogram construction problems, *Proc. of IEEE International Conference on Data Engineering*, vol. 31, pp. 396-438, 2002.

[2] Dan Kennedy , *Incest, Cannibalism, Parochialism and the Genetic Algorithm*, Department of Computer Science and Electrical Engineering, University of Queensland, Brisbane, 1999.

[3] D. E. Goldberg, K. Deb, An investigation of niche and species formation in genetic function optimisation, *Proc. of the 3th International Conference on Genetic Algorithms*, pp. 42-50, 1989.

[4] Weiguo Sheng, Xiaohui Liu, and Michael Fairhurst, A niching memetic algorithm for simultaneous clustering and feature selection, *IEEE Trans. Knowledge and Data Engineering*, vol. 20, no. 7, pp. 868-879, 2008.

[5] Jianjun Hu1, Erik Goodman, *Robust and Efficient Genetic Algorithms with Hierarchical Niching and a Sustainable Evolutionary Computation Model*, Genetic Algorithm Research and Application Group Department of Computer Science and Engineering, Michigan State University, East Lansing, MI, 2003.

[6] S. M. Mahfoud, A comparison of parallel and sequential niching methods, *Proc. of IEEE International Conference on Genetic and Evolutionary Computation* , pp. 136-143, 1995.

[7] B. Sareni, L. Krahenbuhl, Fitness sharing and niching methods revisited, *IEEE Trans. Evolutionary Computation*, vol. 2, no. 3, pp. 97-106, 1998.

[8] A. Pètrowski, A clearing procedure as a niching method for genetic algorithms, *Proc. of IEEE International Conference Evolutionary Computation*, Nagoya, Japan, pp. 798-803, 1996.

[9] F. Herrera, M. Lozano1, C. Moraga, Hierarchical distributed genetic algorithms, *International Journal of Intelligent System*, vol. 14, pp. 1099-1121, 1999.

[10] M. M. Raghuwanshi and O. G. kakde, Quasi steady state genetic algorithm with niches and species, *IEEE Trans. Commerce and Enterprise Computing*, pp. 1-8, 2006.

[11] C. P. Erick, *A Survey of Parallel Genetic Algorithms*, Department of Computer Science and Illinois Genetic Algorithms Laboratory, Urbana-Champaign, 1999.

[12] M. M. Raghuwanshi and O. G. Kakde, Multi-parent recombination operators with multiple probability distributions for real coded genetic algorithm, *Proc. of the 10th On-Line World Conference on Soft Computing in Industrial Application*, vol. 36, pp. 393-402, 2005.

[13] Rahila Patel, M. M. Raghuwanshi, A. N. Jaiswal, Modifying genetic algorithm with species and sexual selection by using K-means algorithm, *Proc. of IEEE International Conference on Advance Computing* , pp. 114-119, 2009.

Appendix I

Unimodal and Multimodal Test Function				
Function	f	**Multimodal**	**Separable**	**Regular**
Ellipsoidal	$f_{Elp} = \Sigma_{i=1}^{n} x_i^{\;2}$	N	Y	n/a
Tablet	$f_{Tablet} = 10^6 x_1^2 + \Sigma_{i=2}^{n} x_i^{\;2}$	N	Y	n/a
Cigar	$f_{Cigar} = x_1^2 + \Sigma_{i=2}^{n} 10^6 x_i^2$	N	Y	n/a
Schwefel	$f_{Sch}(x) = \sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2$	N	N	n/a
Rastrigin	$f_{Rst}(x) = 10n + \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i))$	Y	Y	n/a
Sphere	$f_{Sph} = \Sigma_{i=1}^{n} x_i^{\;2}$	Y	Y	n/a
Griewangk	$f_{Grie} = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	Y	N	Y
Ackley	$f_{Ack}(x) = -20 \cdot \exp\left(-0.2 \cdot \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2}\right) -$ $\exp\left(\frac{1}{n} \sum_{i=1}^{n} \cos(2.0 * 3.14 * x_i)\right) + 20 + e^1$	Y	N	Y