IJCSC Volume 4 • Number 2 September 2013 pp.191-197 ISSN-0973-7391

Effect of changing alpha in arithmetic crossover of GA for solving optimization functions

Girdhar Gopal, Deepak Rathee, Jagdeep Rathee
Department of Computer Science & Applications, Kurukshetra University Kurukshetra, Haryana, India
girdhar.gopal@kuk.ac.in, deepakrathee5@gmail.com, rathee.jagdeep@gmail.com

Abstract— Genetic algorithms are good to find optimum solutions for a broad class of problems. Crossover is the important and main role playing operator in GA toolbox. Reproduction is done through crossover, hence one can optimize crossover to perform at the best. In this paper, arithmetic crossover researched and comparison of gradually changing value of alpha in is compared with fixed value of alpha. The experiments have been conducted using five different benchmark functions and implementation is carried out using MATLAB. Results show the improvement over simple genetic algorithm.
Keywords: Genetic algorithm, Arithmetic Crossover, Alpha Value, Optimization Functions.

I. INTRODUCTION

Genetic algorithms are adaptive algorithms proposed by John Holland in 1975 [1] and were described as adaptive heuristic search algorithms [2] based on the evolutionary ideas of natural selection and natural genetics by David Goldberg. They are powerful optimization techniques that employ concepts of evolutionary biology to evolve optimal solutions to a given problem. Genetic algorithm works with a population of individuals represented by chromosomes. Each chromosome is evaluated by its fitness value as computed by the objective function of the problem. The population undergoes transformation using three primary genetic operators – selection, crossover and mutation which form new generation of population. This process continues to achieve the optimal solution.
The remainder of this paper is organized as follows: Section II presents Overview of Crossovers. Section III describes proposed crossover. Section IV tests the performance of arithmetic crossover with fixed alpha and changing value of alpha crossover and discusses the experimental results. Lastly, Section V contains the conclusion.

II. Overview of Crossovers

Genetic algorithms are population based search techniques. It searches through the state space by exploiting only the coding and objective function in each generation.
Single point crossover is the pioneer crossover technique used in the past [1,2]. In this, a single crossover point on both parent chromosomes is selected by choosing a random number between (1, length-1) where length is length of chromosome. Both the parent chromosomes are split at the crossover point chosen and all data beyond that point in either chromosome is swapped between the two parent chromosomes [3].
N-point crossover operator was first implemented by De Jong in 1975. It is generalized form of single-point crossover differing in number of crossover points [2]. For two point crossover, the value of N is 2. The value of N may vary from 1 to N-1. The basic principle of crossover process is same as that of one point crossover i.e. to exchange genetic material of the two parents beyond the crossover points.
Uniform Crossover operator does not divide the parent chromosome into segments for recombination. Rather, it treats each gene of the chromosome independently to choose for the offspring. In Uniform crossover, number of crossover points is not fixed initially. It recombines genes of parent chromosomes on the basis of crossover mask. It selects x number of crossover points in the chromosome where the value of x is a random value less than the length of the chromosome. Crossover mask is generated according to this random value. In this crossover, each gene in the offspring is created by copying the corresponding gene from one of the parents. The selection of the corresponding parent is undertaken via a randomly generated crossover mask [2,3].
Arithmetic Crossover is used in case of real-value encoding. Arithmetic crossover operator defines a linear combination of two chromosomes [4,5]. Two chromosomes are selected randomly for crossover and produce two offsprings which are linear combination of their parents as per the following computation[6,7]:

$$Cigen+1 = a.Cigen + (1-a).Cjgen$$
$$Cjgen+1 = a.Cjgen + (1-a).Cigen$$

where Cgen an individual from the parent generation , Cgen+1 an individual from child generation, a (alpha) is the weight which governs dominant individual in reproduction and it is between 0 and 1.

IJCSC Volume 4 • Number 2 September 2013 pp.191-197 ISSN-0973-7391

III. Proposed Arithmetic Crossover

The alpha value in arithmetic crossover is fixed at the starting phase of GA in traditional Genetic algorithm. This alpha value plays an important role in deciding dominant parents to form the new generation. This value might be the centre of algorithm, hence a new crossover is proposed with changing value of alpha as the search proceeds. In proposed algorithm alpha value is set to 0.25 initially, and then gradually increased generation wise with a value of 0.2, until it reaches a hill of 0.5. Once it reached to 0.5, it again set back to 0.25.

IV. COMPUTATIONAL EXPERIMENTS AND RESULTS

A. Experimental Set-up: In this paper, 5 different functions are examined in order to compare performance of genetic algorithms and proposed memetic algorithm. Table 1 lists the five test functions – their names, type and their description.

Function	Name	Type
F1	Sphere Function	Unimodal
F2	Rosenbrock's Function	Unimodal
F3	Rastrigin's Function	Multimodal
F4	Schwefel's Function	Multimodal
F5	Ackley's Path Function	Multimodal

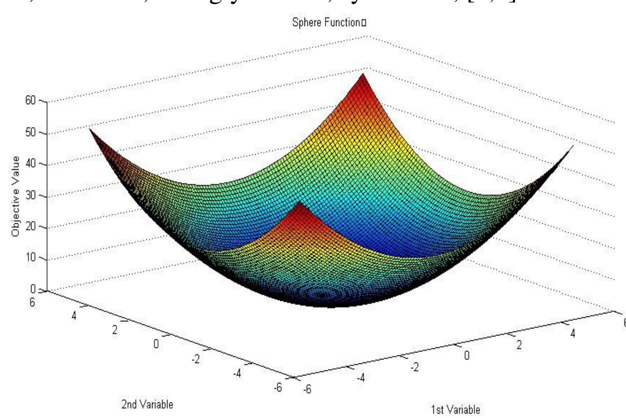
First two functions are by De Jong and unimodal (only one optima) functions, whereas, other three are multimodal (containing many local optimas, but only one global optima) functions.

Sphere [F1] is simple quadratic parabola. It is smooth, unimodal, strongly convex, symmetric, [8,9].

$$F1(x) = \sum_{i=1}^{n} x_i^2$$

$-5.12 \le x(i) \le 5.12$

global minimum: fn(x)=0, xi=0, i=1:n

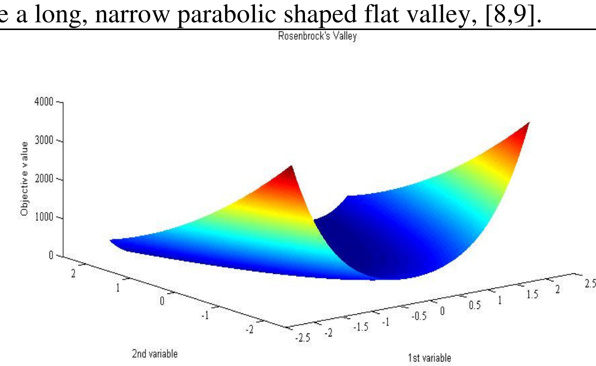


Rosenbrock [F2] is considered to be difficult, because it has a very narrow ridge. The tip of the ridge is very sharp, and it runs around a parabola. The global optimum is inside a long, narrow parabolic shaped flat valley, [8,9].

$$F2(x) = \sum_{i=1}^{n-1} 100 \cdot (x_{i+1} - x_i^2)^2 + (1 - x_i)^2$$

-2.048<=x(i)<=2.048

global minimum: f(x)=0; x(i)=1, i=1:n

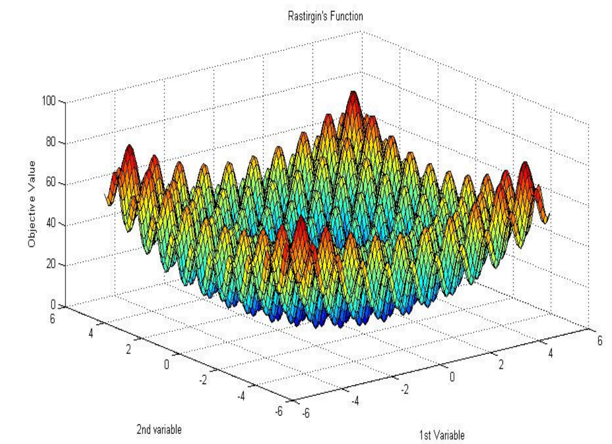


The Rastrigin, [F3] functions is example of non-linear multimodal functions. It is highly multimodal and has a complexity of O(n log(n)), where n is the number of the function parameters. This function contains millions of local optima in the interval of consideration. It has several local minima [8,9].

F3(x) = $10.n+\sum_{i=1}^{n}(x_i^2-10.\cos(2.\pi.x_i))$

-5.12 ≤ x(i) ≤ 5.12

global minimum: f(x)=0; x(i)=0, i=1:n

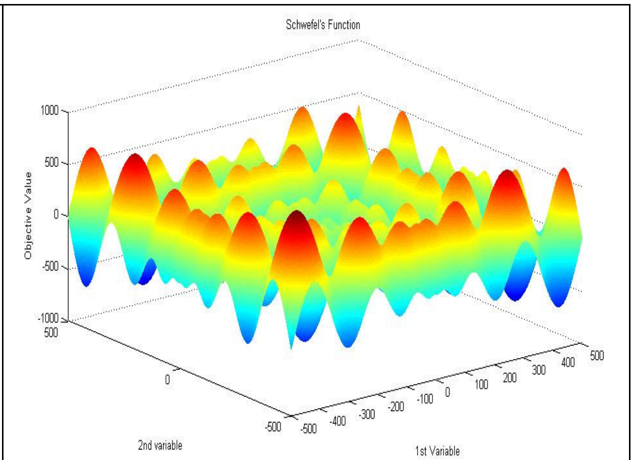


Schwefel's function (F4) is deceptive in that the global minimum is geometrically distant, over the parameter space, from the next best local minima. Therefore, the search algorithms are potentially prone to convergence in the wrong direction, [8,9].

F4(x) = $\sum_{i=1}^{n}-x_i.\sin(\sqrt{|x_i|})$

-500 ≤ x(i) ≤500

global minimum: f(x)=-n·418.9829; x(i)=420.9687, i=1:n



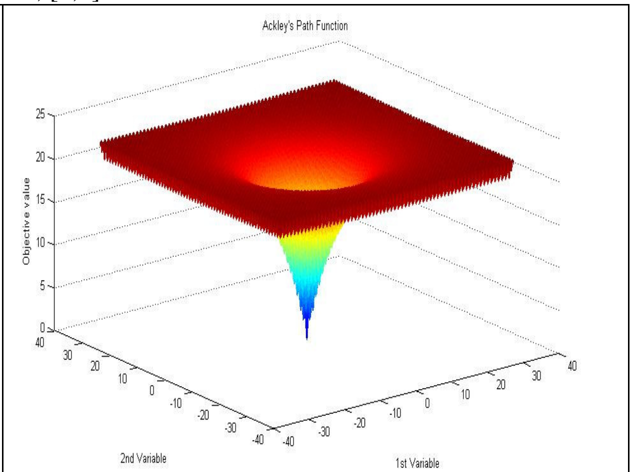
Ackley's Path (F5) is a widely used multimodal test function, [8,9].

F5(x) = $-a.e^{-b.\sqrt{\frac{\sum_{i=1}^{n}x_i^2}{n}}}-e^{\frac{\sum_{i=1}^{n}\cos(c.x_i)}{n}}+a+e^{1}$

a=20; b=0.2; c=2*pi; i=1:n;
-32.768<=x(i)<=32.768

global minimum:

f(x)=0; x(i)=0, i=1:n



The following parameters are used in this implementation:

- Population size (N): 10, 20, 50
- Number of generations (ngen) : 100 and 500

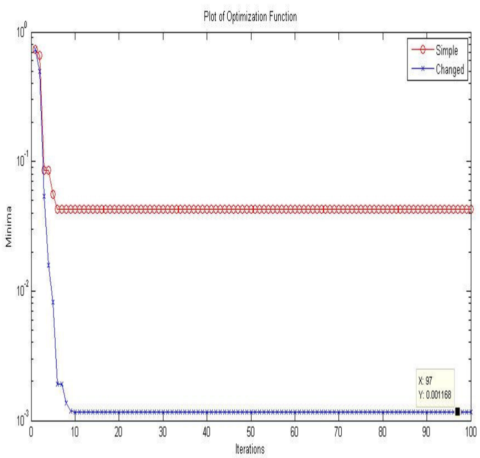
- Selection method: Roulette Wheel Selection (RWS)
- Crossover Operator: Arithmetic Crossover (alpha = 0.25)
- Mutation: uniform with mutation probability 0.01%
- Algorithm ending criteria: Execution stops on reaching ngen generations
- Fitness Function: Objective value of function

A. Experimental Results

Average and minimum values of each function is recorded and examined for further analysis.

Table II Minimum values for F1

N		10	20	50
Gen = 100	GA with fixed alpha	0.6572	0.043	5.258e-004
	GA with changed alpha	0.3316	0.001168	7.2137e-006
Gen = 500	GA with fixed alpha	0.3516	0.0153	1.729e-138
	GA with changed alpha	0.25	4.0797e-004	0 (4.941e-324 at 350th Iteration)



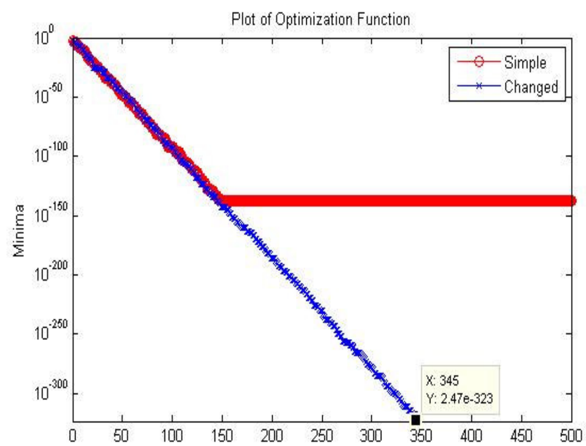
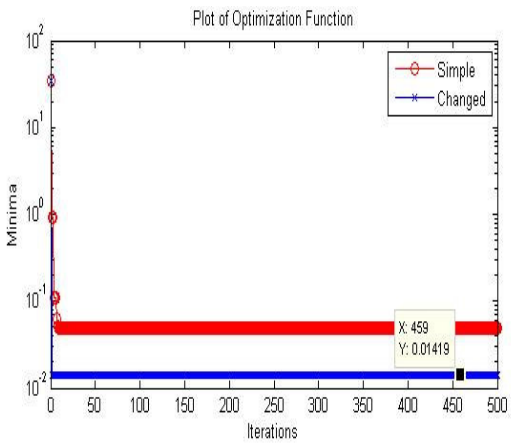


Table III Minimum values for F2

N		10	20	50
Gen = 100	GA with fixed alpha	0.952	0.9618	0.468
	GA with changed alpha	0.6544	0.5372	1.1942e-005
Gen = 500	GA with fixed alpha	0.485	0.0495	0.1969
	GA with changed alpha	0.0277	0.0141	7.354e-005



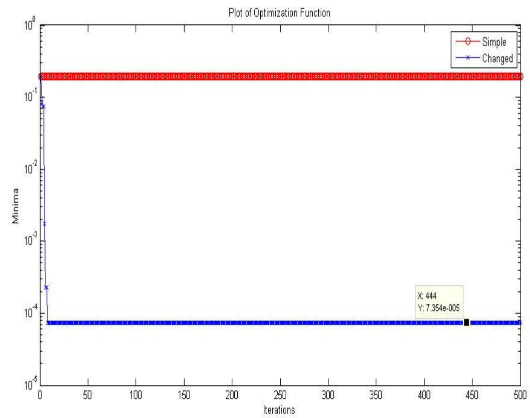
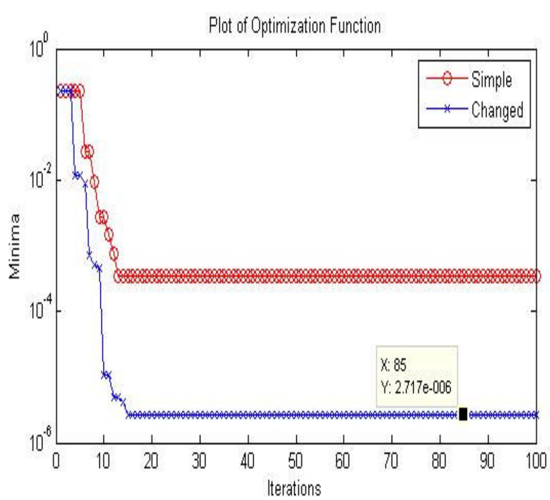


Table IV Minimum values for F3

N		10	20	50
Gen = 100	GA with fixed alpha	1.309	2.1611	3.3726e-004
	GA with changed alpha	0.0905	0.1899	2.7171e-006
Gen = 500	GA with fixed alpha	0.00128	0.09865	4.7916e-009
	GA with changed alpha	1.6120e-004	0.07995	1.0579e-011



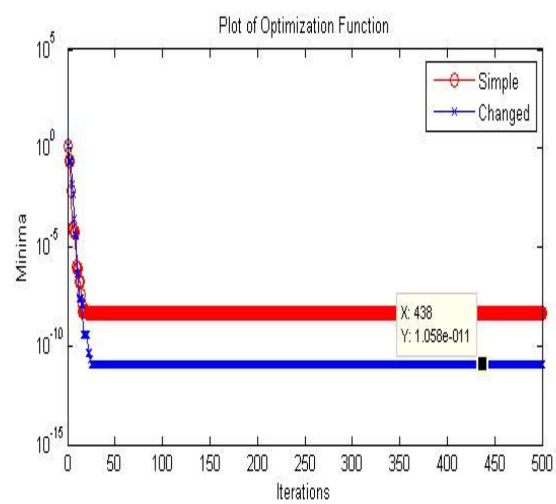
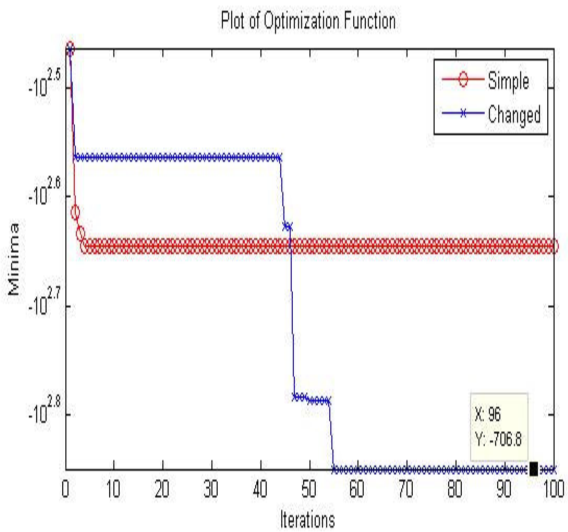


Table V Minimum values for F4

N		10	20	50
Gen = 100	GA with fixed alpha	-4.4229e+002	-4.653e+002	-6.159e+002
	GA with changed alpha	-7.0679e+002	-5.113e+002	-8.216e+002
Gen = 500	GA with fixed alpha	-6.695e+002	-7.153e+002	-7.058e+002
	GA with changed alpha	-7.995e+002	-7.785e+002	-8.050e+002



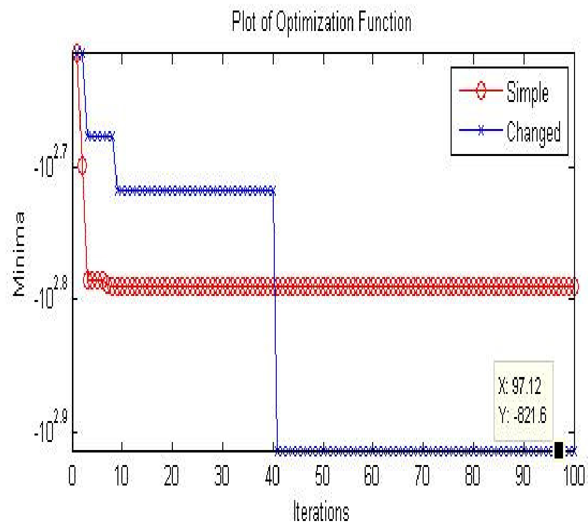
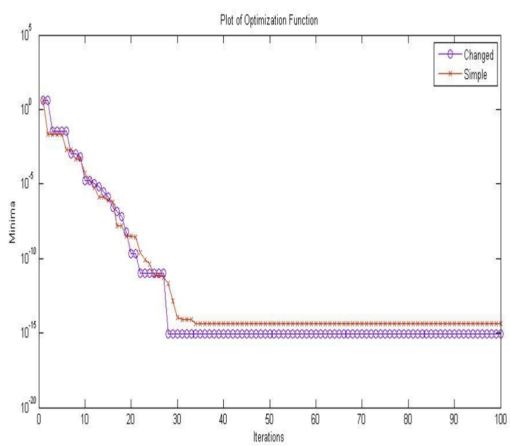
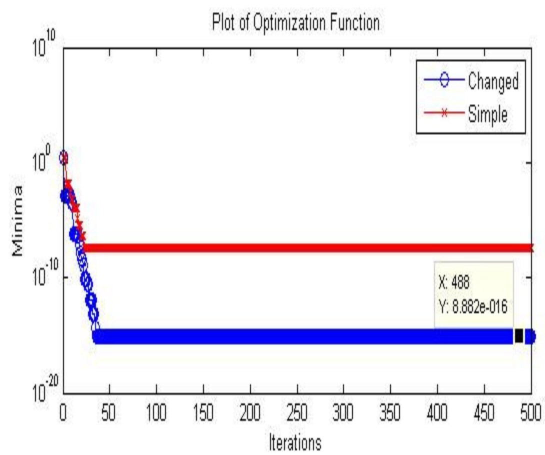


Table VI Minimum values for F5

N		10	20	50
Gen = 100	GA with fixed alpha	0.577	0.3565	4.4408e-015
	GA with changed alpha	0.466	0.1534	8.8817e-016
Gen = 500	GA with fixed alpha	0.365	0.2988	4.298e-008
	GA with changed alpha	0.256	0.0014	8.8817e-016





VI. CONCLUSION

In optimization problems, one solution is never available for all problems. Genetic algorithm is an art of creating solutions for a wide class of problems. It depends on encoding scheme that which operators we can use as selection, crossing over and mutation steps of GA. In this the most promising operator is crossover. Arithmetic crossover is used when real value encoding is used. Changing value of alpha in arithmetic crossover is better than the fixed value in all generations.

References

1. Holland J., Adaptation in natural and artificial systems, University of Michigan Press, Ann Arbor, 1975.
2. Goldberg D. E., Genetic algorithms in search, optimization, and machine learning, Addison Wesley Longman, Inc., ISBN 0-201- 15767-5, 1989.
3. Weise Thomas, "Global Optimization Algorithms – Theory and Application –", 2011, 3rd Edition.
4. Michalewicz Z., Genetic Algorithms + Data Structures = Evolution Programs, Springer-Verlag, 2nd edition, 1994.
5. Bethke Albert Donally. "Genetic Algorithms as Function Optimizers". PhD thesis, University of Michigan: Ann Arbor, MI, USA, 1980.
6. De Jong K., Spears W. M., "Using Genetic Algorithms to Solve NP Complete Problems," Proceedings of the Third International Conference on Genetic Algorithm, Morgan Kaufman, Los Altos, CA, 124 – 132, 1989
7. De Jong K. A., "An Analysis of the behavior of a class of genetic adaptive systems", (Doctoral dissertation, University of Michigan) Dissertation Abstracts International 36(10), 5140B University Microfilms No. 76/9381, 1975.
8. Salomon, R., "Re-evaluating genetic algorithm performance under coordinate rotation of benchmark functions: A survey of some theoretical and practical aspects of genetic algorithms", Elsevier:BioSystems, 1996, pp 263-278.
9. Digalakis J.G. and K.G.Margaritis, "An Experimental Study of Benchmarking Functions for Genetic Algorithms", International Journal of Computer Mathematics, 2002, Vol.79, No.4, pp 403-416.