OPTIMIZATION OF BENCHMARK FUNCTIONS USING GENETIC ALGORITHM

TWINKLE GUPTA
Department of Computer Science,
Hindu Kanya MahaVidyalya,
Jind, India

***Abstract*— In this paper, Benchmark functions for single-objective optimization cases are presented and their performance is optimized using an optimization technique called Genetic algorithm. Genetic algorithm was introduced by John Holland at University of Michigan, United State in 1970s. It is a population and nature-inspired algorithm which selects the chromosomes of better fitness from the current population using Roulette wheel and removed the worst one. Selected chromosomes are used to produce the chromosomes of the best fitness for the next generation by applying genetic operators called offspring. After successive generations, the population moved towards an optimal solution and does not stuck out to local optima. Simulation results are shown by taking various Benchmark Functions namely: Sphere, Ackley, Booth, Easom and Matyas and their results are compared on parameters: population size, number of generations and dimensions.**

Keywords- Genetic Algorithm, Optimization, Genetic operators, Roulette Wheel Selection, Crossover, Mutation, Benchmark Functions.

I. INTRODUCTION

In day to day life, we are encountered with various optimization problems like Travelling salesman problem, Robotics, Clustering etc [1]. These problems can be divided into two categories i.e.

1. Single-objective optimization problems.
2. Multi-objective optimization problems.

Single-objective optimization problems are those problems that give us minimum or maximum whichever is objective on a single value only and in case of multi-objective optimization problems there are more than one optimization solutions of a single problem. Here we are presenting only single-objective Benchmark Functions and solution is generated using an optimization algorithm called Genetic algorithm. Various single-objective Benchmark Functions are: Sphere function, Rosenbrock function, Ackley's function, Beale's function etc

There are various optimization algorithms divided into two categories: Evolutionary algorithms and Swarm optimization algorithms [2]. Genetic algorithm, an evolutionary algorithm, is a nature-inspired [3] and population based algorithm given by John Holland at University of Michigan, United State in 1970s and presented in this paper [4]. Genetic algorithm is based on an evolution process in such a way that firstly it finds the fitness of each chromosome (possible solution in a search space) and selects the best fit chromosome among population of chromosomes using Roulette Wheel selection method and removed the worst one i.e. best fit always survives. Now, this selected chromosome become parents and give offspring of more fitness by applying genetic operators: crossover and mutation and these new chromosomes become the population for next generation. Same process is continued until a specific criterion is reached like maximum number of generations. Its main characteristics are:

1. Employs some parameters like number of generation, population size, probability of crossover and mutation.
2. Flexible and robust.
3. Simple to implement.
4. Easily integrated with other optimization algorithm.

Rest of the paper is divided in different sections as follows: Introduction to Genetic algorithm is described in section 2. Experiments and its simulation results to show performance on several Benchmark functions are described in section 3 and in the last; Conclusion of the paper is discussed.

II. GENETIC ALGORITHM

Genetic algorithm begins with some parameter fixed like maximum number of generations to meet stopping criterion, fitness function, probability of crossover for exploitation and probability of mutation for exploration [5]. Steps of Genetic Algorithm are:

A. Initialization

Population (a set of possible solutions in a search space) is initialized and represented by chromosome. Chromosome is a string of binary digits and each bit is called gene.

B. Evaluate fitness

Fitness of each chromosome is calculated based on fitness function that is used.

C. Selection

Chromosomes of better fitness value are selected using Roulette wheel selection method. More the fitness value more the chances of selection of a chromosome.

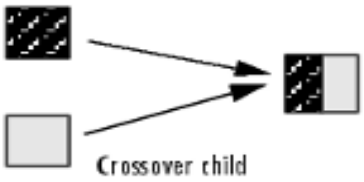
D. Apply genetic operator

New population is generated by producing offspring after applying genetic operators:

1) Crossover:

It combine the features of two chromosomes and produce the offspring of better fitness value than both of the old chromosome's fitness value. Two chromosomes are selected based on their probability i.e. a random value is generated for each chromosome if this value is less than probability of crossover then it is selected otherwise not.
For crossover a position is selected to interchange information. Example:

ISSN: 2278 – 1323



TABLEI CROSSOVER OPERATOR

S.No.	Chromosome Selected	Pos selected	Chromosome after crossover
1	111000|0000111111000111	6	111000|1110110101011111
2	100010|1110110101011111	6	100010|0000111111000111

2) Mutation:

It provides the exploration in a search space and provides new genetic material to avoid local optima. In this operator a bit is flipped i.e. if it is one then become 0 and it is zero then become one , if its probability is less than probability of mutation.

E. Accepting

Place new offspring in the population.

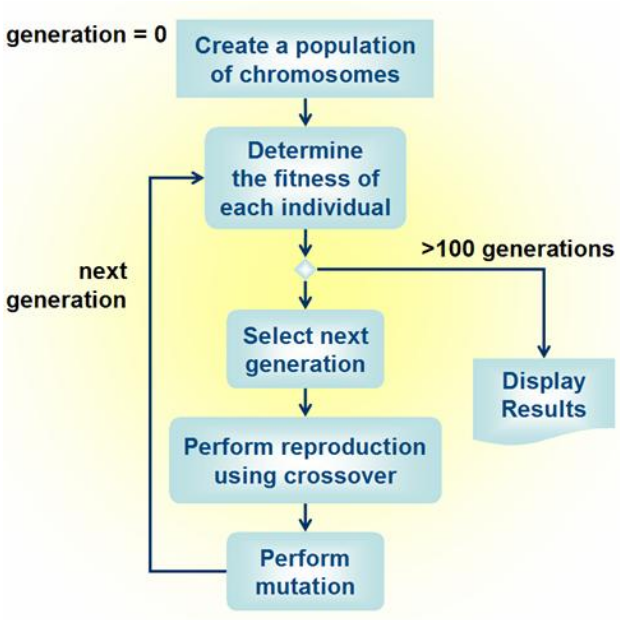
F. Check stopping criterion

Now check the stopping criterion either the maximum no. of generation is reached or the fitness limit is reached.

G. Stop/ Run the next generation

If the stopping criterion is reached then stop otherwise LOOP to the STEP 2 with the new population generated in STEP 5.

III. FLOWCHART OF GENETIC ALGORITHM



IV. EXPERIMENTS AND SIMULATION RESULTS OF BENCHMARK FUNCTIONS

A. Benchmark Functions

The Benchmark Functions are used to compute the performance of Genetic algorithm [6, 7]. Its optimization is compared with different probability of crossover and mutation and illustrated below:

1) Sphere:s

$$ObjVal = \sum_{j=1:p} x^2$$

Where x_i=[-100.000, 100.000]

2) Ackley

$$ObjVal = -20 * e^{-0.2*\sqrt{\frac{1}{2}*(x_1{}^2+x_2{}^2)}} - e^{\frac{1}{2}*(\cos(2\pi x_1)+\cos(2\pi x_2))} + 20 + e^1$$

Where x_i=[-20.000, 20.000]

3) Booth:

$$ObjVal = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$$

Where x_i=[-30.000, 30.000]

4) Easom:

$$ObjVal = -\cos(x_1)\cos(x_2) * e^{(-((x_1-\pi)^2+(x_2-\pi)\wedge 2))}$$

Where x_i=[-100.000, 100.000]

5) Matyas:

$$ObjVal = 0.26(x_1{}^2 + x_2{}^2) - 0.48x_1x_2$$

Where x_i=[-100.000, 100.000]

Here, ObjVal is the function value calculated for each chromosome. A chromosome is a possible solution and is represented by x. Population size is taken of n*p matrix where n is the no. of possible solutions and p represents the dimension of each solution.

B. Performance measures and Simulation results

MGN (maximum generation number) is set as 1000 and each algorithm is run for 1000 iteration and best value is recorded of each generation. In order to provide the quantitative assessment of the performance of an optimization algorithm, Global Minimum for each Benchmark functions is recorded.

ISSN: 2278 – 1323

The experimental results of Genetic algorithm in MATLAB are taken under various parameters:

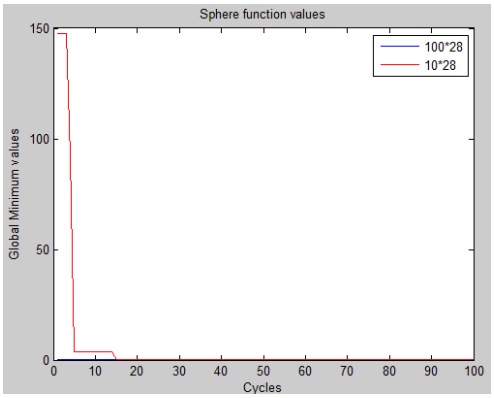
1) Population Size:

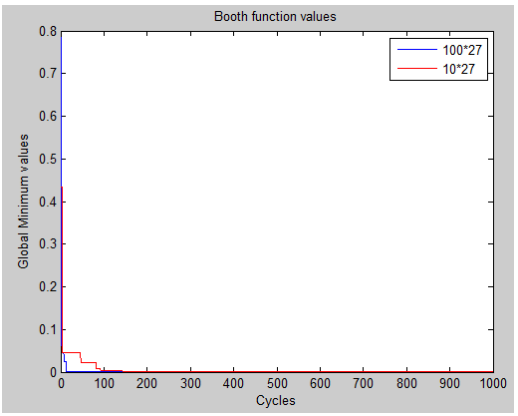
It is represented by n*p matrix where n represents the no. of solutions and p represents the dimension. Here 10 and 100 possible solutions are taken in each generation i.e. n=10 and precision is set to 1000. P is decided according to this precision and range of a Benchmark function and result is compared.

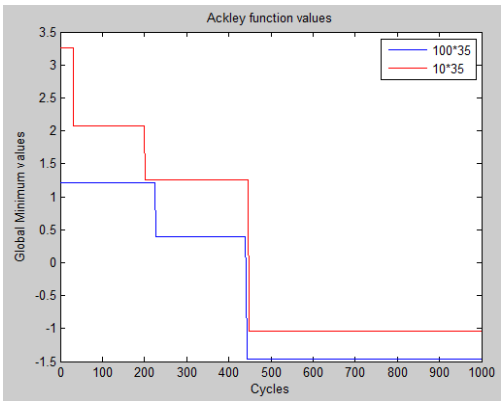
TABLEII GLOBAL MINIMUM VALUE ON TWO DIFFERENT POPULATION SIZE

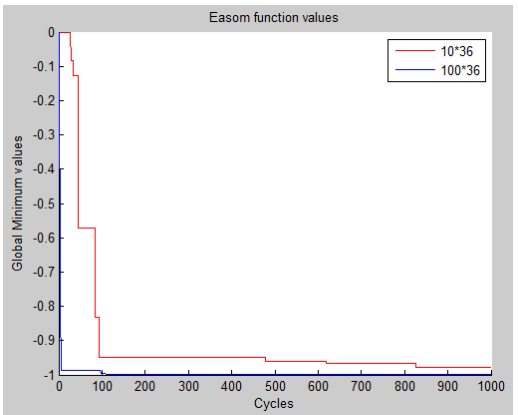
Benchmark function	Crossover Probability	Mutation Probability	Population size	First variable (x1)	Second variable (x2)	Global minimum value	Actual Global optima
Sphere	0.7	0.2	10*28	0.000e+003	0	1.1067e-007	F(0)=0
			100*28	0.000e+006	0	1.3878e-013	F(0)=0
Ackley	0.7	0.2	10*35	-0.0038	-0.0784	-1.0392	F(0,0)=0
			100*35	-0.0560	-3.0518e-004	-1.4707	F(0,0)=0
Booth	0.7	0.2	10*27	0.9995	3.0010	2.1459e-006	F(1,3)=0
			100*27	1.0002	2.9998	1.1923e-007	F(1,3)=0
Easom	0.7	0.2	10*36	3.2520	3.0827	-0.9768	F(pi,pi)=-1
			100*36	3.1208	3.1414	-0.9994	F(pi,pi)=-1
Matyas	0.7	0.2	10*36	-1.6674	-1.5453	0.1069	F(0,0)=0
	0.7	0.2	100*36	0.0393	0.0744	4.3715e-004	F(0,0)=0

FIG. I Optimization of Benchmark functions using GA with different population size

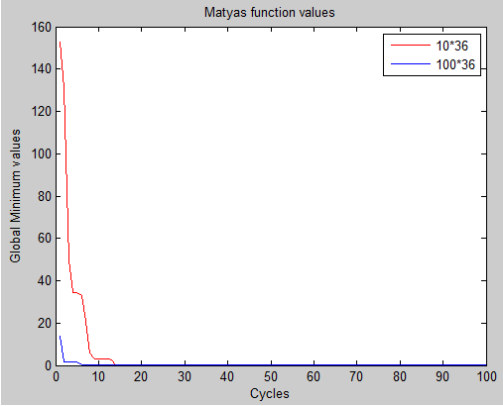








ISSN: 2278 – 1323



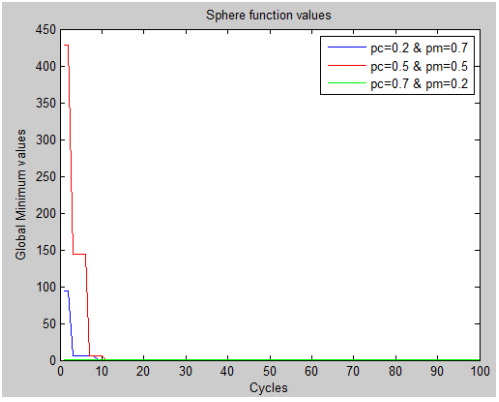
2) Probability of crossover and mutation:

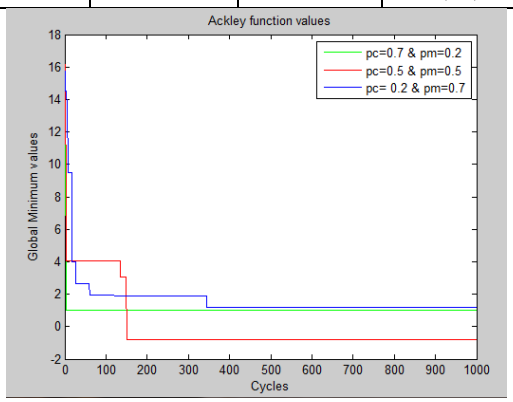
Different combination of probability of crossover (pc) and probability of mutation (pm) is taken and results are observed. Here we have taken three combinations for each Benchmark Functions: 1) pc= 0.7 & pm=0.2 2) pc=0.5 & pm=0.5 3) pc=0.2 & pm=0.7.

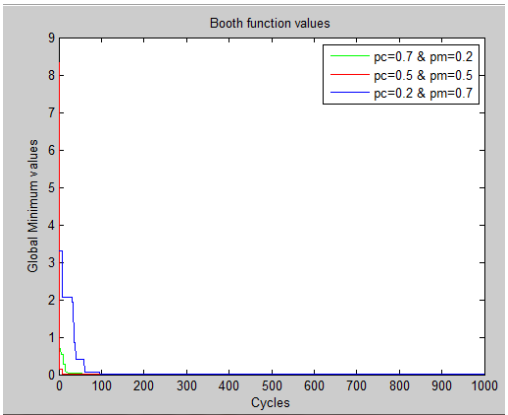
TABLEIII GLOBAL MINIMUM VALUE FOR THREE DIFFERENT VALUES OF CROSSOVER AND MUTATION PROBABILITY

Benchmark function	Population size	Crossover Probability	Mutation Probability	First variable(x1)	Second variable(x2)	Global minimum value	Actual Global optima
Sphere	10*28	0.7	0.2	0.00024	0	9.3742e-007	F(0)=0
		0.5	0.5	0.0458	0	0.0021	F(0)=0
		0.2	0.7	0.0256	0	2.3510e-004	F(0)=0
Ackley	10*35	0.7	0.2	0.2753	0.0345	1.0345	F(0,0)=0
		0.5	0.5	0.1084	-0.0479	-0.7818	F(0,0)=0
		0.2	0.7	-1.0162	0.0583	1.2164	F(0,0)=0
Booth	10*30	0.7	0.2	1.0024	2.9979	1.0806e-005	F(1,3)=0
		0.5	0.5	0.9958	2.9827	0.0022	F(1,3)=0
		0.2	0.7	0.9123	3.0327	0.0208	F(1,3)=0
Easom	10*36	0.7	0.2	3.1506	3.1139	-0.9987	F(pi,pi)=-1
		0.5	0.5	3.7289	3.2627	-0.5768	F(pi,pi)=-1
		0.2	0.7	1.8810	2.8980	-0.0570	F(pi,pi)=-1
Matyas	10*36	0.7	0.2	1.3096	0.7961	0.1103	F(0,0)=0
		0.5	0.5	-1.8978	-2.2732	0.2092	F(0,0)=0
		0.2	0.7	-3.4039	-3.1345	0.4456	F(0,0)=0

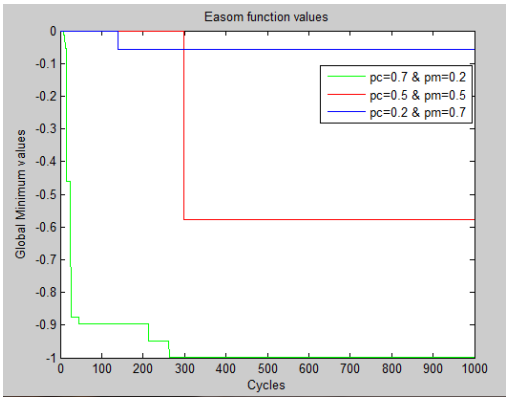
FIG. II Optimization of Benchmark functions using GA with different crossover and mutation probability

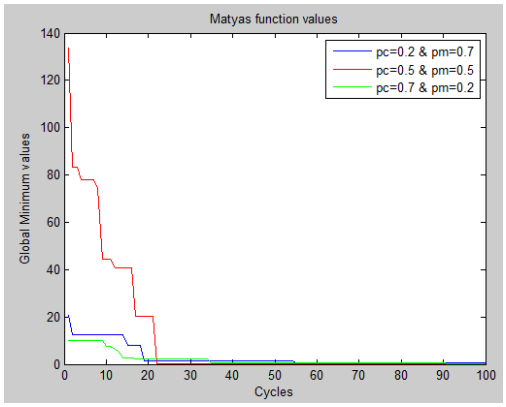






ISSN: 2278 – 1323





In GA algorithm there is a need to proper balance between exploration and exploitation that is what probability of crossover and mutation decides and same effect is shown on various Benchmark Functions using MATLAB in Table 2 with different pc and pm. From all figures we can conclude that we can reach at global optima on low value of pm and high value on pc. Here pc=0.7 and pm= 0.2 will be the best choice. First table show that GA also works best on high population size.

V. DISCUSSION AND CONCLUSION

In this paper, Various Single-objective Benchmark Functions are shown and Genetic algorithm is used to optimize the problem. Experiment is done using MATLAB with different population size which shows that GA performs with greater efficiency on high population size also. GA is also performed on various Benchmark function with different crossover and mutation probability showing that there is a great need of balance between exploration and exploitation in GA to optimize the problem

REFERENCES

[1] Man, K.F. and Tang, K.S. and Kwong, S., "Genetic algorithms: concepts and applications [in engineering design]" Industrial Electronics, IEEE Transactions on, vol. 43, no. 5, pp. 519-534, Oct. 1996.

[2] Kennedy and R. Eberhart, "Particle swarm optimization," in Proceedings of IEEE international conference on neural networks, 1995, vol. 4, pp. 1942–1948.

[3] Yugal Kumar and Dharmender Kumar, "Parametric Analysis of Nature Inspired Optimization Techniques" International Journal of Computer Applications, vol. 32, no. 3, pp. 42-49, Oct. 2011.

[4] P. J. Angeline, J. B. Pollack and G.M. Saunders, "An evolutionary algorithm that constructs recurrent neural networks," Neural Networks in IEEE Transactions on, vol. 5, no. 1, 1994, pp. 54-65.

[5] Twinkle Gupta, "Implementation of an Optimization Technique: Genetic Algorithm" International Journal of Advanced Research in Computer Engineering & Technology, vol. 4, no. 12, pp. 4359-4364, Dec. 2015.

[6] Marcin Molga, Czesław Smutnicki, "Test Functions for Optimization Needs" Test functions for optimization needs, May 2005.

[7] Jamil M, Yang XS , "A literature survey of benchmark functions for global optimisation problems" International Journal of Mathematical Modelling and Numerical Optimisation, vol. 4, no. 2, pp. 150-194, Jan. 2013.

ISSN: 2278 – 1323