Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 6.017

IJCSMC, Vol. 6, Issue. 4, April 2017, pg.23 – 28

Enhancing the Results of Genetic Algorithm by Using Fitness Scaling Functions

Sangeeta[1], Bal Kishan[2]
[1]M.Tech Student DCSA, MDU Rohtak, Haryana, India
[2]Assistant Prof. DCSA, MDU Rohtak, Haryana, India
[1] army.sangeeta46@gmail.com
[2] balkishan248@gmail.com

Abstract - Genetic algorithm is one of the most optimizing techniques used today. GA can use the raw fitness of population which can be used for evolution. In this paper we have just using the fitness scaling with genetic algorithm and just observe the output of simple GA and the output of GA with fitness scaling. Fitness scaling can covert the raw fitness of population into the fittest population. In this paper we can apply various types of fitness scaling with GA and try to analysis the results with each type of scaling. The main drawback of simple genetic algorithm is premature convergence but by applying fitness scaling this problem is somehow solved. Our main motive in this paper is just to analyze the difference between good result and best result. The results in this paper show a better diversity and accuracy by using proposed fitness scaling with genetic algorithm.

Keywords-Genetic algorithm (GA), Fitness function, Fitness scaling, Premature convergence, Evolutionary programming, Mutation, Crossover, Selection

I. INTRODUCTION

During the last thirty years, there has been the growth of interest in the field of evolutionary programming relies on the analogies based on natural processes. Genetic algorithms are one of the heuristic based optimization techniques.GA is a stochastic based searching algorithm which can helps to search in a large search space. It is a directed search which can be inspired by the biological evolution.

GA is an iterative process in which the new population is created by selecting the best chromosomes from previous “generation”. The selected chromosomes are recombined, crossover, mutated and then the new generation is created. The classical GA’s are just implemented over fixed length binary strings and only binary crossover and mutation is applied on them so they were not being the case of evolutionary programming and affected by premature convergence [1]. But now a days, the GA is not just limited to binary values instead it can be applied over a huge area of research and computational problems (NP Complete) like TSP (Travelling salesman problem).

A. Evolutionary Programming:

The evolution program is nothing but just a probabilistic king of algorithm which can maintain the population of individuals likes:

$$P(t)=(w_1^t,w_2^t\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.w_n^t)$$

For each iteration (t).Each solution ($w_i^{t)}$ is evaluate to calculate their "fitness". And by selecting the fit individuals the next population is created i.e. (t+1). After that some genetic operators are applied on the selected population. Crossover and mutation are mostly used operators.

B. Crossover:

This is a process of interchanging the genes of two chromosomes. The main purpose of interchange is to get fitter chromosomes. The crossover point decides that from where to where the genes are replaced. These fitter chromosomes are then used for further evolution. The chromosomes are selected with the probability (P_c) whose value lies between 0 and 1. It can helps to exploitation of search space.

C. Mutation:

Mutation alters one or more genes of chromosomes according to their mutation rate. The mutation can be done according to the mutation probability (P_m) ranges between 0-1. It also exploits the search space. It can play a vital role in to ensuring genetic diversity in the whole population.

II. FITNESS FUNCTION

The fitness of an individual is just an objective function value of any GA. In multicriterion optimization the fitness function is much more complicated to understand. Here the difficulty lies in the way to identify that how one solution is better than another. The fitness function may be either a simple binomial function or a complex trigonometric function. But there is not any proper method how to define which one is better than another. It totally depends on the situation where a normal function gives better result than a complex one. So in short the fitness function can helps to select the best suited individuals for the next generation [2].

A. Genetic Algorithm Table Before Applying Fitness Scaling Function:

TABLE 1
SIMPLE GENETIC ALGORITHM TABLE USING (PI,CI)=(12,23)

G_i	S_i	A_i	M_i	MN_i
20	266.89	22.24	28.62	10.79
50	237.44	19.78	25.48	13.73
100	270.16	22.51	32.07	15.31
150	277.87	23.15	32.56	10.06
200	250.99	20.92	25.97	10.79

TABLE 2
SIMPLE GENETIC ALGORITHM TABLE USING (PI,CI)=(15,30)

G_i	S_i	A_i	M_i	MN_i
20	340.29	22.69	32.31	17.21
50	331.49	22.09	27.29	13.04
100	313.78	20.89	26.25	16.34
150	315.61	21.04	31.18	13.82
200	346.17	23.08	36.31	13.82

P_i,C_i): Population size, Chromosome size
G_i: Number of generations
S_i: Sum of fitness
A_i: Average of fitness
M_i: Maximum of fitness
MN_i: Minimum of fitness

Here, in both the tables we can evaluate the sum, average, minimum and maximum fitness of simple genetic algorithm. For evaluating the results we can use the fix probabilities of crossover and mutation i.e. P_c and P_m(0.6,0.01). Accordingly we get distinct number of crossover and mutation for every iteration. In this the genetic algorithm can generate the fitness values up to six decimal places but can round off them only up to two decimal places in our table.

III. FITNESS SCALING

The simple genetic algorithm when applied has a limitation of premature convergence which is somehow reduced by using fitness scaling. Fitness scaling is applicable at the initial stage of genetic algorithm which can helps in the exploration of search space by selecting better kind of chromosomes from the whole set of population [2]. And conversely, at the later stage of evolution the fitness scaling can helps to strengthen of selection pressure of chromosomes. The fitness scaling mainly converts the raw fitness of population retrieved by fitness function into the new scaled fitness which is suitable for the selection function [8].

Fitness function only differentiates between good result and best result but simple GA faces some problem in this procedure. So the fitness scaling can helps to overcome these obstacles by scaling this fitness in the whole population there by modifying the fitness function [14].

There is various kind of fitness scaling functions. Some of them are:-

A. Linear scaling:

It is one of the most useful and simplest kinds of scaling. It can highly dependent on two constants a and b. The raw fitness f can be used for deriving the scaled fitness f' such that

$$f' = af + b \qquad [8]$$

Our main concern is that in all the cases-

$$f'_{avg} = f_{avg} \qquad [3]$$

So that the further selection procedure can participate in generation of offspring in next iteration.

Algorithm of linear fitness scaling

```
Algo Scalepop(pop, max, avg, min)
begin
Float a, b;
Prescale(max, avg, min, a, b);
For i= 1 to popsize
pop[i].scaled_fitness=a*pop[i].fitness+b;
end
```

Prescale(max,min,avg,a,b)//CALCULATE VALUE OF A,B

```
if(min>(avg*cmult-max)/(cmult-1.0))
{
  ϖ=max-avg;
  if(ϖ<=0.00001)
  {
  a=1;b=0;

  else

  a=((cmult-1.0)*avg)/ϖ;
  b=avg*(max-cmult*avg)/ϖ;
 }
}
else
{
   ϖ=avg-min;
  if(ϖ<=0.00001)
     {
     a=1;
     b=0;
     }
```

B. Power law scaling:
This scaling is firstly proposed by(Gillies 1985). The power scaling can greatly enhance the performance of genetic algorithm. Here the f(z) can be described as-

$$f(z)=z^{\alpha}$$

In this α is used as a constant and its best value is 1.005; but its other values also applicable as per complexity of problem criteria.

C. Exponent scaling:
This type of scaling is most widely used in robotics and is motivated by the analogy with simulated annealing. The exponential scaling is evaluated as-

$$f(z)=exp(-\beta z)$$ [18]

In this β is used as constant and its value always (>=0).

A. Genetic Algorithm Table After Applying Linear Fitness Scaling Functions:

TABLE 3
.LINEAR FITNESS SCALING APPLYING IN GA WITH (P_I,C_I)=(12,23)

G_i	S_i	A_i	M_i	MN_i
20	267.33	22.27	34.68	0.00
50	251.17	20.93	38.39	0.00
100	257.21	21.43	45.07	5.55
150	280.70	23.39	46.31	5.72
200	249.97	20.83	31.37	-0.00

TABLE 4
LINEAR FITNESS SCALING APPLYING IN GA WITH (P_I,C_I)=(15,30)

Gi	*Si*	*Ai*	*Mi*	*MNi*
20	346.28	23.09	45.37	9.77
50	344.53	22.96	34.78	-0.00
100	311.37	20.76	41.78	3.17
150	304.71	20.31	42.08	6.06
200	351.04	23.09	46.15	6.92

In Tables 4, 5 we just apply the linear scaling over the same population and chromosome size. We can use same crossover and mutation probability but the results are more optimal as compared to simple GA.

B. Genetic Algorithm Table After Applying Power Fitness Scaling Functions:

TABLE 5
POWER LAW FITNESS SCALING IN GA WITH (P_I,C_I)=(12,23)

Gi	*Si*	*Ai*	*Mi*	*MNi*
20	271.09	22.59	29.11	10.92
50	241.04	20.08	25.89	13.91
100	274.43	22.86	32.63	15.52
150	282.30	23.52	33.14	16.28
200	254.86	21.23	26.40	10.92

TABLE 6
POWER LAW FITNESS SCALING IN GA WITH (P_I,C_I)=(15,30)

G_i	S_i	A_i	M_i	MN_i
20	345.67	23.04	32.87	17.45
50	336.68	22.44	27.74	13.21
100	318.19	21.21	26.68	16.57
150	320.48	21.36	31.72	14.00
200	351.69	23.44	36.95	14.00

In Tables 5, 6 the same input set of values provide more efficient results as compare to linear scaling. If we compare table 5 and 6 when population and chromosome size will be 15, 30 then in linear scaling minimum fitness is 10 but in case of power scaling it will remains greater than 13.

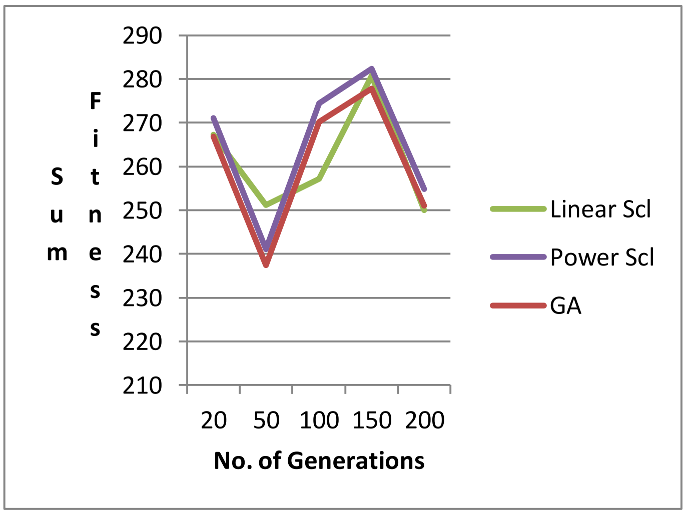

Fig.1 Sum_Fitness Of Genetic Algorithm Using Power, Linear And Without Scaling Using (P_i, C_i)=(12,23)

In figure 1, show the output of variation in sum fitness of genetic algorithm using linear fitness scaling, power scaling and without any fitness scaling in GA. As we have seen in fig1.1, the power fitness scaling provides better outputs as compared to linear scaling and simple GA. In fig. 1, When the population size is 12 and chromosomes size is 23 than initially when the number of generations are less than the linear scaling gives better result but as the number if generations increases the power fitness scaling proves to be better than the linear fitness scaling.

IV. CONCLUSION

In this paper we just apply linear and power fitness scaling algorithms on simple GA and try to analyze the resulting outputs according to the different size of population and chromosomes size. After doing a lot of research on the input parameters we can conclude that the power scaling provide more better fitness as compared to simple GA and linear scaling. But it is not always true because sometimes the linear scaling provides better outputs than power scaling according to the value of population and chromosomes size. In the resulting figure 1, we are able to see that the power scaling is better than linear scaling. By applying complex trigonometric function and larger size of population and chromosome size we can analyze the performance of GA and its premature convergence.

REFERENCES

1. Bhawna, Gaurav Kumar and pardeep Kumar Bhatia,"*Software Test Case Reduction using Genetic Algorithm:A Modified Approach*"IJISET,Vol 3,Issue 5,May 2016.
2. Anamika Taya," *Performance Improvement of Genetic Algorithm by Fitness Scaling and Diversity Maintenance*" in International Journal for Research in Technological Studies, vol. 2, Issue 7, June 2015.

3. V. Kreinovich, C. Quintana, and O. Fuentes, "*Genetic algorithms: what fitness scaling is optimal?*," Cybernetics and Systems, vol. 24, no. 1, pp. 9–26, 1993.
4. J. H. Holland and D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading, MA, 1989.
5. M. Zhou, S. D. Sun, "*Theory and application of genetic algorithm*" presented at National defense press, Beijing, 1996.
6. G.-S. Hao, Y. Yu-Chen, K.-X. Wei, G. Gong, and X.-T. Hu, "Parameters selection of fitness scaling in genetic algorithm and its application," presented at the 2010 Chinese Control and Decision Conference, 2010, pp. 2475–2480.
7. L. Nolle, A. Armstrong, A. Hopgood, and A. Ware, "*Optimum work roll profile selection in the hot rolling of wide steel strip using computational intelligence*" in International Conference on Computational Intelligence, 1999, pp. 435–452.
8. S. Hill, J. Newell, and C. O'Riordan, "*Analysing the effects of combining fitness scaling and inversion in genetic algorithms*" presented at the Tools with Artificial Intelligence, 2004. ICTAI 2004. 16th IEEE International Conference on, 2004, pp. 380–387.
9. J. H. Holland and D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading, MA, 1989.
10. S. N. Sivanandam and S. N. Deepa, Introduction to Genetic Algorithms. Springer Science & Business Media, 2007.
11. N. Surajudeen-Bakinde, X. Zhu, J. Gao, and A. K. Nandi, "*Effects of fitness scaling and adaptive parameters on genetic algorithm based equalization for DS-UWB systems*" presented at the Computers and Devices for Communication, 2009. CODEC 2009. 4th International Conference on, 2009, pp. 1–4.
12. B. Sareni and L. Krahenbuhl, "*Fitness sharing and niching methods revisited*" IEEE Transactions on Evolutionary Computation, vol. 2, no. 3, pp. 97–106, Sep. 1998.
13. P. Darwen and X. Yao, "*A dilemma for fitness sharing with a scaling function*" presented at the Evolutionary Computation, 1995, IEEE International Conference on, 1995, vol. 1, pp. 166-171.
14. Nitin S. Choubey and Madan U. Kharat ,"*Performance Evaluation of Methods for handling Premature Convergence in GA - Case of Grammar Induction*" in International Journal of Computer Applications, vol. 79, no. 2, pp. 9-13, Oct. 2013.
15. A. A. Hopgood and A. Mierzejewska, "*Transform ranking: a new method of fitness scaling in genetic algorithms*" in Research and Development in Intelligent Systems XXV, Springer, 2009, pp. 349–354.
16. P. Da:rwen and X. Yao, "*How good is fitness sharing with a scaling function*'' Tech. Rep. CS 8/95, University College, The University of New South Wales, Canberra, Apr. 1995.
17. Z.Michaelwicz,"*Genetic Algorithm + Data Structures = Evolution Programs*", Springer, 3rd revised and extendededition,Oct.1995.
18. F. Sadjadi, "*Comparison of fitness scaling functions in genetic algorithms with applications to optical processing*" in Optical Science and Technology, the SPIE 49th Annual Meeting, 2004, pp. 356–364.