ISSN:
Print - 2277 - 0593
Online - 2315 - 7461

Journal of Natural Science, Engineering and Technology

PARAMETER VARIATION FOR LINEAR EQUATION SOLVER USING GENETIC ALGORITHM

***[1]A. M. IKOTUN, [2]A. T. AKINWALE AND [2] O. T. AROGUNDADE**

[1]Department of Computer Technology, Yaba College of Technology, Lagos.
[2,3]Department of Computer Science, Federal University of Agriculture, Abeokuta, Nigeria
***Corresponding author:** biodunikotun@gmail.com, **Tel:** +2348188890899

ABSTRACT

Genetic Algorithm has been successfully applied for solving systems of Linear Equations; however the effects of varying the various Genetic Algorithms parameters on the GA systems of Linear Equations solver have not been investigated. Varying the GA parameters produces new and exciting information on the behaviour of the GA Linear Equation solver. In this paper, a general introduction on the Genetic Algorithm, its application on finding solutions to the Systems of Linear equation as well as the effects of varying the Population size and Number of Generation is presented. The genetic algorithm simultaneous linear equation solver program was run several times using different sets of simultaneous linear equation while varying the population sizes as well as the number of generations in order to observe their effects on the solution generation. It was observed that small population size does not produce perfect solutions as fast as when large population size is used and small or large number of generations did not really have much impact on the attainment of perfect solution as much as population size.

Keywords: Genetic Algorithm, Genetic Algorithm Parameters, GA Number of Generations, GA Population Size, Genetic Algorithm Runs, Systems of Linear Equation.

INTRODUCTION

Genetic Algorithm, as a branch of evolutionary computation is a search and optimization technique which works based on Evolutionary Principle of Natural Chromosomes. It is a non-mathematical, non-deterministic but stochastic process for solving optimisation problems. They represent an intelligent exploitation of a random search used in solving optimization problems. They exploit historical information to direct the search into the region of better performance within the search space. The idea of genetic algorithm was introduced by Holland in 1975. It was based on the evolutionary theory of Darwin (1859) called Darwinian evolution. The simulation of Charles Darwin's evolutionary process using the computer produces Genetic Algorithm (Holland, 1975).

A System of Linear Equation is a set or collection of two or more linear equations. Linear Equations play an important role in virtually all fields of Science and Engineering. In Science and Engineering field, technical computation are required for finding numerical solution to various equations which represent realistic problems like natural phenomenon or engineering problems. Systems of Linear equation of the form $Ax = b$ is used in the representation of such natural phenomenon or engineering where the coefficient matrix A is typically a large matrix of size $n X n$.

The Genetic Algorithm approach follows the concept of solution evolution by stochastically developing generations of solutions population using a definite fitness function to determine the best fit solution to the problem. Genetic Algorithm has been successfully applied for solving systems of Linear Equations, however the effects of varying the various Genetic Algorithms parameters such as the population size and the number of generations on the GA systems of Linear Equations solver has not been deeply researched into.

This study investigates the effects of varying the GA parameters – population size and number of generations on the time required to search for optimal values for the unknown variables in the equations that best fit the systems of linear equations.

Over the years, great attention has been given to building computer systems, which can exhibit the distinct characteristics of intelligence of man. There have been classical and neo-classical attempts at modelling intelligence. The earliest and most significant paper on machine intelligence titled *"Computing machinery and intelligence"* was written by Turing (1950). Turing, in his paper, defined the intelligence behaviour of a computer as the ability to achieve the human–level performance in cognitive tasks (Lawal, 2003).

Several research attempts have been made to simulate the complex thinking processes found in human and two prominent field of research arose, which is regarded as the Connectionism (that is, Neural Network and Parallel Processing) and the Evolutionary Computing.

Connectionism involves the researches that investigate the activities of neurons in the brain in relation to intelligence characteristics exhibited by man. This is called Artificial Neural Network.

Researches in Evolutionary Computing investigate the computational capabilities of the biological genetics in relation to intelligence creation, which could be observed over several generation of evolution. This gave birth to Evolutionary Computations.

Evolutionary Computation was introduced by Ingo (1964) in his work titled *"Evolution Strategy"*. It is an aspect of intelligence computing which involves the simulation of evolution on a computer. It comprises of Genetic Algorithm, Genetic Programming and Evolution Strategies. Genetic Algorithm was initiated by John Holland (1975), Genetic Programming by Foget (1966). Further research works in Evolutionary Computation led to the discovery of Classifier systems by Holland (1994), Messy Genetic Algorithm and several others (Zbigniew, 1996).

Evolutionary Computation is an evolutionary approach to problem solving based on computational models of natural selection and genetics. In solving problems using this approach, the idea is that the best or some good/better solutions are searched for from several possible solutions. The space of all feasible solution is tested by applying it to the problem. The better fitted ones are retained for further usage in the search for the optimal solution while the lesser fitted ones are discarded.

The fundamental constituents of evolutionary computation include:

1. Selection
2. Reproduction
3. Crossover
4. Mutation

Evolutionary computation maintains a population of structures that evolve according to the rules of selection and the genetic operators of crossover and mutation. Each individual in the population receives a measure of its fitness in the environment. Reproduction focuses attention on high fitness individuals, thus exploiting the available fitness information. Recombination and mutation perturb those individuals, providing general heuristics for exploration. These computations, though simple from a biologist's point of view, provides robust and powerful adaptive search mechanism.
Punam(2012) researched solving linear and non-linear equation using Genetic algorithm. He applied Genetic Algorithm as a nonlinear technique in solving linear and nonlinear equation system and also investigated the major benefits obtained as a result of using GA. He used the GA approach to confirm that the Gauss-Legendre numerical integration is more effective than fundamental method of Newton's in system of nonlinear equation. According to him, "The result of using GA compared to the exact solutions obtained by some different numerical methods turned to be confirming".

Ikotun *et al* (2011) investigated the effectiveness of using GA to solve simultaneous equation investigated the applicability and effectiveness of genetic algorithms in finding the solutions of systems of linear equations. Solving a system of equations is regarded again as an optimization problem. The proposed algorithm is able to produce more than one set of solutions for systems of equations. In that approach a chromosome is represented by a vector with integer values that corresponds to the unknowns of the system. The crossover and mutation operation are used in solution process.

Ibrahiemet al (2008) studied the application of GA in solving Non-Linear Equation Systems. Their paper proposes a new paradigm for solving systems of nonlinear equations through using Genetic Algorithm (GA) techniques. It illustrated how genetic algorithms (GA) techniques can be used in finding the solution of a system described by nonlinear equations. According to them, the obtained results indicated that GA is effective and represents an efficient approach to solve the systems of nonlinear equations that arise in the implementation of Gauss–Legendre numerical integration.

Roshiand Jignesh (2012) discussed the optimisation of linear equations using Genetic Algorithms. In their research, they confirmed that genetic algorithm is capable of producing more than one set of solutions for certain systems of equations. According to them " To avoid the disadvantages like rounding errors, inverting large matrices, GA are introduced."

NiKos (2005) investigated solving non-linear equation via Genetic Algorithm. He applied GA to solve a non-linear equation as well as systems of non-linear equation From his findings, he declared that "GA (Genetic Algorithms) is really a powerful tool for many problems in numerical analysis and scientific computation.

Isaac (2013) in his paper 'Solution of Nonlinear Equation via Optimisation' presented four optimisation methods for solving Nonlinear Equations.

Mafteiu-Scaiand Mafteiu-Scai proposes a memetic algorithm (MA) to solve linear systems of equations, by transforming the linear system of equations into an optimization problem.

Researches on the effects of GA parameter variations are also reported. For instance, Sarac and Cvetkovski (2011) developed different motor models based on parameter variations in GA. Avnii *et al* (2013) analyses the impact of parameter values on the Genetic Algorithm in TSP. Bo and Marcus proposes a hybrid approach to parameters tuning in Genetic Algorithm .

Nazmat *et al* (2009) investigated the effects of fitness scaling and adaptive parameters on GA based equalisation for DS-UWB Systems. The influence of the Population size on the GA performance in case of cultivation Process Modelling was investigated by Olympa et al (2013).

Marjan and Sadegh (2012) studied the effect of a New Generation Based Sequential selection operator on the Performance of GA. Liang and Changhai (2011) worked on Empirically identifying the Best GA for covering array generation .

Odat *et al* (2012) studied the effect of changes in Population size and Number of Generations in the application of GA for Node Placement in WMNs. Visniand MuraliBhaskarai (2013) worked on improving the performance of GA by reducing the Population Size. The effect of Local Search on the Performance of GA were studied by Richa and Mittal (2014)

This research work is a further investigation on the earlier work reported in Ikotun *et al* (2011) which investigated the effectiveness of Genetic Algorithm in Solving System of Linear Equation.

MATERIALS AND METHODS

To demonstrate the effect of the varying Genetic Algorithm parameters (Population Size and Number of Generation) on the execution time , the Genetic Algorithm paradigm was used to find the solution to the system of Linear equation , then the two parameters were then varied to find their effects on the time required to evolve solution from the GA Linear Equation Solver.

The GA Linear Equation solver reported in Ikotun *et al* (2011) was employed. The GA Linear Equation solver utilized randomly generated data to represent each unknown variables to form each chromosome. Several of the chromosomes were generated to form the initial generation. Integer values were randomly generated for each variable. The values were then used in each line of equation to find the fitness of each chromosome. Integer value encoding is applied and the integer values were directly manipulated. During the crossover operation and mutation operation, the genes (integer values) in each chromosome selected for the respective operation were used.

To compute the fitness of each chromosome, the concept of coefficient of multiple determination (R^2) was used. This concept is also called the squared multiple correlation coefficient

The values of R^2 range between 0.0 and 1.0 (i.e. $0.0 \leq R^2 \leq 1.0$); and the fitness values that are closer to 1 imply a better fitness while a fitness value of 1.0 gives the best fitness that produces the most accurate solution to the equations.

A well fitted set of generation forms the initial population for the next generation and subsequently until the stopping criteria of fitness of 1.0 or very close to 1.0 is obtained or the maximum generation indicated in the program has been reached.

To investigate the effect of number of generations on the genetic algorithm implementation, the number of generations was varied with a population size of 100 for each set of equation

To investigate the effect of the population size, maximum generation of 10 and 50 were used on the genetic algorithm implementation for each set of equations.

RESULTS

The GA Linear Equation Solver was run several times using different set of system of Linear Equations. The number of generations was varied with a population size of 100 for each set of equation and different population sizes were selected with maximum generation of 10 and 50 for each set of the equations. The various set of equations used is shown in the Table 1:

Table 1: Comparing Results from Genetic Algorithm Linear Equation Solver and Gaussian Elimination

Experiment No	Equations	GA Result	Gaussian Elimination Solution	Actual Solution
1	x1 + 2x2 + 3x3 = 14 x1 + x2 + x3 = 6 3x1 + 2x2 + x3 = 10	x1 =1 x2 =2 x3 = 3 x1 =0 x2 =4 x3 = 2 x1 =2 x2 =0 x3 = 4	x1 =0 x2 =4 x3 = 2	x1 =1 x2 =2 x3 = 3
2	2x1 + 4x2 + x3 = 5 4x1 + 4x2 + 3x3 = 8 4x1 + 8x2 + x3 = 9	x1 =0 x2 =1 x3 =1	x1 =-0.5 x2 =1.25 x3 =1	x1 =0 x2 =1 x3 =1
3	10x1 + x2 + x3 = 12 2x1 + 10x2 + x3 = 13 2x1 + 2x2 + 10x3 = 14	x1 =1 x2 =1 x3 =1	x1 = 1 x2 = 1 x3 = 1	x1 =1 x2 =1 x3 =1
4	2x1 + x2 + 3x3 + x4 = 8 x1 + 2x2 + 2x3 + 3x4 = 15 3x1 + 3x2 + x3 + 2x4 = 13 4x1 + 4x2 + 3x3 + 3x4 = 20	x1 = 0 x2 = 2 x3 = 1 x4 = 3	x1=19.6212 x2=-23.303 x3=-12.258 x4=23.8331	x1 = 0 x2 = 2 x3 = 1 x4 = 3
5	4x1 + 3x2 + 2x3 + x4 = 30 3x1 + 2x2 + x3 + 4x4 = 25 2x1 + 1x2 + 4x3 + 3x4 = 16 x1 + 4x2 + 3x3 + 2x4 = 19	x1 = 5 x2 = 3 x3 = 0 x4 = 1	x1=4.1040 x2=2.8348 x3=-0.0977 x4=6.8750	x1 = 5 x2 = 3 x3 = 0 x4 = 1
6	x1 + x2 + x3 = 2 2x1 + 3x2 + 4x3 = 25	x1 = 1 x2 = 0 x3 = 1	No Solution	x1 = 1 x2 = 0 x3 = 1
7	2x1 + x2 + 3x3 = 10 3x1 + 2x2 + x3 = 13	x1 = 2 x2 = 3 x3 = 1	No Solution	x1 = 2 x2 = 3 x3 = 1
8	x1 + 2x2 = 8 2x1 + x2 = 7 3x1 + 2x2 = 12	x1 = 2 x2 = 3	No Solution	x1 = 2 x2 = 3
9	2x1 + x2 = 4 x1 + 2x2 = 5 3x1 + 2x2 = 7	x1 = 1 x2 = 2	No Solution	x1 = 1 x2 = 2

Effects of the number of Generations

To investigate the effect of number of generations on the genetic algorithm linear equation solver, the number of generations was varied with a population size of 100 for each set of equation and the result reported in Tables 2, 3 and 4.

Table 2: Effect of varying maximum generation for Equation 1

Maximum Generation Per Run	No. of Runs Required to obtain the closest/ best fitness	Best Fitness
10	2	1
20	9	1
30	12	1
40	15	1
50	18	1

Table 3: Effect of varying maximum generation for Equation 2

Maximum Generation Per Run	No. of Runs Required to obtain the closest/ best fitness	Best Fitness
10	2	1
20	4	1
30	4	1
40	2	1
50	3	1

Table 4: Effect of varying maximum generation for Equation 3

Maximum Generation Per Run	No. of Runs Required to obtain the closest/best fitness	Best Fitness
10	1	1
20	1	1
30	1	1
40	1	1
50	1	1

Effects of the population size

To investigate the effect of the population size, maximum generation of 10 and 50 were used on the genetic algorithm linear equation solver for each set of equations and the results reported in table 5, 6 and 7.

Table 5: Effect of varying population size for Equation 1(3x3)

Max Gen	Population	No of Runs	Best Fitness
10	50	3	1
10	100	2	1
10	150	1	1
10	200	3	1
10	250	3	1
50	50	>5	1
50	100	>5	1
50	150	1	1
50	200	2	1
50	250	2	1

Table 6: Effect of varying population size for Equation 2 (3x3)

Max Gen	Population	No of Runs	Best Fitness
10	50	6	1
10	100	2	1
10	150	2	1
10	200	4	1
10	250	8	1
50	50	3	1
50	100	3	1
50	150	2	1
50	200	3	1
50	250	3	1

Table 7: Effect of varying population size for Equation 4 (4x4)

No of Runs	Max Gen	Population	Best Fitness
>10	10	10	0.9943978
7	10	20	1
6	10	30	1
6	10	50	1
1	10	100	1
1	50	10	1
3	50	10	1
4	50	20	1
2	50	30	1
1	50	50	1
1	50	100	1

DISCUSSION

It was observed that small population size does not produce perfect solutions as fast as when large population size is used. Large population size guarantees wide search for candidate solutions, from which good and better solutions could be derived in successive generations. However, as population size gets larger, it produces an overhead cost on the processing speed and storage. As the storage gets filled up with more candidate solution solutions, it hampers the processing speed of the computer system on which the GA is being implemented. Thus, the population size should not be allowed to become outrageously large.

It was also observed that small or large number of generations did not really have so much impact on the attainment of perfect solution as much as population size. At times small number of generations (e.g. 10) produces perfect results after a few runs of the GA program, while at some other times, large number of generations produces (e.g. 50 and above) perfect solutions after several runs of the program.

CONCLUSION

The efficiency of the GA program is dependent on the availability of enough storage and processing speed of the computer system being used for its implementation. Large population size is required for complex problem to obtain perfect solutions As the problem gets more complex, more storage and faster processing speed are required since larger population of candidate solutions guarantees attainment of good and better solutions in successive generations.

REFERENCES

Avni R., Adnan, M., Agni D. 2013. Analysis of the impact of parameters values on the Genetic Algorithm for *TSP. International Journal of Computer Science Issues*, 10

Benny R., 2000. Gaussian Elimination code in Java, Version 0.0.retrieved from http://www.wiley.com/WileyCDA/WileyTitle/productCd-0471487155.html

Fogel, D. B. 1998 Evolutionary Computation. *The Fossil Record, IEEE Press, New York.*

Franz, Rothlauf 2006. *Representations for Genetic and Evolutionary Algorithms*, Second Edition, Springer, USA

Goldberg, D. E 1989. Genetic Algorithms in Search, Optimization and Machine Learning. *Addison-Wesley, USA*.

Holand J. L. 1975. *Adaptation in Natural and Artificial System Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. University of Michigan Press, USA

Holland, J. L. 1994. The Self-Directed Search. *Odessa, FL: Psychological Assessment Resources*

Ibrahiem M.. El Emary, M., Mona, M., Abd E . 2008. On the application of Genetic Algorithm in Fingerprint Registration. *World Applied Science Journal.* 5(3): 276-281

Ikotun, A., Lawal, M., Olawale, N., Adelokun A. P. 2011. The Effectiveness of Genetic Algorithm in Solving Simultaneous Equations. *International Journal of Computer Applications 14(8):38–41*

Isaac S. 2013. Practical Portfolio Optimisation. *Apex Research Limited*, London.

Lawal, N. O. 2003. Genetic Programming: Applications. In: Econometrics. *An MSc Dissertation Submitted To The Department of Computer Sciences, School of Postgraduate Studies, University of Lagos, Lagos Nigeria*.

Lipson, M., Lipschutz, S. 2001. *Schaum's outline of theory and problems of linear algebra.* New York: McGraw-Hill.

Mafteiu-Scai L. O., Mafteiu-Scai E. J. 2013.Solving Linear Systems of Equations using a Memetic Algorithm. *International Journal of Computer Application.* 58(13): 16-22

Marjan, K. R., Sadegh E. 2012. The Effect of a New Generation Based Sequential Selection Operator on the Performance of Genetic Algorithm *Indian Journal of Science and Technology.* 5 (12):195-204

Mhetre, P. S. 2012. Genetic algorithm for linear and nonlinear equation. *Int. Journal of Advanced Engineering Technology*. 3:114-118.

NiKos E. M. 2005. Solving Non-Linear Equations Via Genetic Algorithms. *Proceedings of the 6th WSEAS Int. Conf. on Evolutionary Computing, Lisbon, Portugal, June 16-18.*24-28

Olympia R., Stefka, F., Paprzycki, M. 2013. Influence of the Population Size on the Genetic Algorithm Performance in Case of Cultivation Process Modelling. *Federated Conference on Computer Science and Information Systems*. 371–376

Roshni, V., Jignesh, P., Patel, S. 2012. Optimization of Linear Equations using Genetic Algorithms. *Indian Journal of Applied Research*. 2(3): 56-58.

Sarac, V., Cvetkovski, G., 2011. Different motor models based on parameter variation using method of genetic algorithms .*Electrical Review.* ISSN 0033-2097, R. 87 NR 3/2011

Turing, A.M. 1950. Computing Machinery and Intelligence .*MIND*, 59,433-460.

Michalewicz, Z. 1996. *Genetic Algorithms + Data Structures = Evolution Program*, Third, Revised and Expanded Edition, *Springer, USA*.

(Manuscript received: 4th July, 2015; accepted: 15th June, 2016).