COMP 6776: Project Report

Solving Non-Homogeneous System of Linear Equations using Genetic Algorithm

Submitted By:
Aafia Jabeen – 201790648
Asma Javaid – 201790860

Abstract

This project finds the roots of non-homogeneous system of linear equations using genetic algorithm. Our genetic algorithm approach follows the concept of solution evolution by stochastically evolving generations of population solutions using a definite fitness function to determine the best fit solution to the problem. After finding the solution using G.A we compared it with conventional numerical algorithms and show that that results obtained by Genetic algorithm approach is more effective than fundamental numerical iterative methods for system of linear equations.

1. Introduction

Linear Equations play an important role in almost all fields of Science and Engineering [1]. In all of these fields nominal computation is required for finding numerical solution to various equations which represent realistic problems like natural phenomenon or engineering problem. A System of Linear Equations is a collection of two or more linear equations and it has the following form

$$
\begin{gathered}
a_{11}x_1 + a_{12}x_2 + \cdots \ldots\ldots\ldots\ldots\ldots . a_{1n}x_n = b_1 \\
a_{21}x_1 + a_{22}x_2 + \cdots \ldots\ldots\ldots\ldots\ldots . a_{2n}x_n = b_2 \\
a_{31}x_1 + a_{32}x_2 + \cdots \ldots\ldots\ldots\ldots\ldots . a_{3n}x_n = b_3 \\
. \\
. \\
. \\
a_{m1}x_1 + a_{m2}x_2 + \cdots \ldots\ldots\ldots\ldots\ldots . a_{mn}x_n = b_m .
\end{gathered}
$$

These equations help in the representation of such natural phenomenon or engineering where the coefficient matrix A is sparse or typically a large matrix of size[1]. For finding the roots of these equations many numerical iterative and non-iterative approaches have been introduced and some of most popular non-iterative methods are *Gauss Elimination Method, Gauss Jordon Method and iterative methods are Jacobi's Method and Gauss Seidal method.*

Jacobi's Method and Gauss Seidal method(G.S) guaranteed the roots of these equations but the procedure is very time-consuming and convergence rate is low [1]. Here we find roots of equations through the evolutionary process of genetic algorithm for system of equations [3]. 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.

In 1999 Simon and Sean[5] give an overview of genetic algorithm for the solution of optimization problem in which they defined that system of equations can be solved using genetic algorithm, if we can define it in form of an objective function. Recently Lubna zahgul[3] solve one linear equation using genetic algorithm.
We used Genetic Algorithm, on one hand, and *Gauss Seidal*, on the other hand, to solve system of simultaneous linear equations. We particularly used diagonally dominant system of linear equation to find solution using G.A because we have to compare our result with existing numerical algorithms which only converge if equations are diagonally dominant. However, it can even find a solution if equations are not diagonally dominant. Here diagonally dominant means that coefficient of first variable of first equation should be greater then the sum of coefficients of other variables of the first equation and so on that is

$$|a_{ii}| \geq \sum a_{ij} \ \forall \, i, j$$

2. Genetic Algorithm Methodology

Below is the typical genetic algorithm approach that we used in this project.

Step1 Encode the roots of equation into chromosomes for our initial population within a defined domain.
Step2 Find the minimum absolute minimum value of the below set of equations in each generation against each chromosome

$$\text{absolute value of } (F_1 \left(x_1, x_{2,} \ldots \ldots, x_n\right) - b_1)$$
$$\text{absolute value of } (F_2 \left(x_1, x_{2,} \ldots \ldots, x_n\right) - b_2)$$
$$\text{absolute value of } (F_n \left(x_1, x_{2,} \ldots \ldots, x_n\right) - b_n)$$

Step3 The fitness function that need to be minimized by the genetic algorithms is define as

$$K = \max[abs(f_i)] \text{ for i=1, 2, 3,……, N.}$$

Here **K** is the fitness function that need to be minimized after every generation, max [abs (f_1)] is the maximum absolute value of individual equations in the system f(x) = 0, and N is the number of equations in the system [3].

Step4 Use roulette wheel selection method to select the individual that generate the next population. Individuals with high fitness probability are selected and the rest are eliminated. This is done by the imaginary wheel containing the search space and functions as follows

- Evaluate the fitness function F_i for each chromosome in the population.
- Calculate the probability for each chromosome by P[i]= Fitness[i]/Total
- Compute the cumulative probability values
- Generate a random number r between 0 and 1
- If $C_i \leq \mathrm{r} \leq C_{i+1}$ then select i + 1 chromosome as a chromosome in the new population for next generation.

Step5 Use one-point crossover operator by randomly selecting two chromosomes from the parent pool.

Step6 Use random mutation.

Stept7 Iterate algorithm from step 2-6 until number of generations achieved or best solution is obtained.

3. The Proposed Methodology of Solution

To validate the applicability of genetic algorithm in finding the solution to the system of equation, the genetic algorithm paradigm has been applied to following set of linear equations and find the values for the unknowns in the equations. These set of equations are already diagonally dominant, and we choose them intentionally. However, one can consider any set of equations. The set of unknowns in these equations are x_1 , x_2 , x_3 .
Several of the chromosomes initially generated for these unknown's variables. These chromosomes form the initial population for the next generation and subsequently until the stopping criteria 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, we varied the number of generations 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.

$6x_1 + x_2 + x_3 = 105$ $4x_1 + 8x_2 + 3x_3 = 155$ $5x_1 + 4x_2 + 10x_3 = 165$	$4x_1 + x_2 + 3x_3 = 17$ $1x_1 + 5x_2 + 1x_3 = 14$ $2x_1 - x_2 + 8x_3 = 12$

4. Simulated Output Results using G.A and G.S

In this section we will solve the above define linear system of equations using the following parameters define in table 1. Since we have to find the value of three variables, so we selected string length of three variables as three and then display the result in table 2. In Table 2, we also display the result of Gaussian Elimination method.

Table1: Parameter setting used by the evolutionary approach.

Parameter	Values for Example 1
Population size	150
Number of generations	100
mutation	0.23
crossover	0.8

Table2: Results from *Gauss Seidal* and Genetic Algorithm Linear Equation Solver.

Equations	Gaussian Seidal	Genetic Algorithm
$4x_1 + x_2 + 3x_3 = 17$ $1x_1 + 5x_2 + 1x_3 = 14$ $2x_1 + x_2 + 8x_3 = 12$	$x_1 = 3.455284$ $x_2 = 2.032520$ $x_3 = 0.382113$	$x_1 = 3$ $x_2 = 2$ $x_3 = 1$

5. Comparison and Conclusion

It was firstly observed during the experiment that our G.A does not give a perfect solution for small population size. But when number of generations was varied with a different population sizes we came to know that large population size guarantees the exact solution. It was also analyzed during the experiment that gauss siedal methods stops when roots start repeating even if we run it for 100 generations but G.A keeps running until number of generations reach.

```
Generation ----------------->1
****************************************************************
100
Values of xyz ented by user----->chrmosome 0 ||x=>12||y=>15||z=>11
Consolidated chromosome report eq1-->72 15 11||eq2-->48 120 33||eq3-->60 60 110
Values of xyz ented by user----->chrmosome 1 ||x=>13||y=>14||z=>13
Consolidated chromosome report eq1-->78 14 13||eq2-->52 112 39||eq3-->65 56 130
Values of xyz ented by user----->chrmosome 2 ||x=>0||y=>14||z=>15
Consolidated chromosome report eq1-->0 14 15||eq2-->0 112 45||eq3-->0 56 150
Values of xyz ented by user----->chrmosome 3 ||x=>4||y=>11||z=>1
Consolidated chromosome report eq1-->24 11 1||eq2-->16 88 3||eq3-->20 44 10
Values of xyz ented by user----->chrmosome 4 ||x=>10||y=>9||z=>13
Consolidated chromosome report eq1-->60 9 13||eq2-->40 72 39||eq3-->50 36 130
Values of xyz ented by user----->chrmosome 5 ||x=>17||y=>6||z=>2
Consolidated chromosome report eq1-->102 6 2||eq2-->68 48 6||eq3-->85 24 20
Values of xyz ented by user----->chrmosome 6 ||x=>4||y=>18||z=>17
Consolidated chromosome report eq1-->24 18 17||eq2-->16 144 51||eq3-->20 72 170
Values of xyz ented by user----->chrmosome 7 ||x=>6||y=>13||z=>4
Consolidated chromosome report eq1-->36 13 4||eq2-->24 104 12||eq3-->30 52 40
Values of xyz ented by user----->chrmosome 8 ||x=>11||y=>1||z=>17
Consolidated chromosome report eq1-->66 1 17||eq2-->44 8 51||eq3-->55 4 170
Values of xyz ented by user----->chrmosome 9 ||x=>2||y=>13||z=>18
Consolidated chromosome report eq1-->12 13 18||eq2-->8 104 54||eq3-->10 52 180
Values of xyz ented by user----->chrmosome 10 ||x=>18||y=>13||z=>1
Consolidated chromosome report eq1-->108 13 1||eq2-->72 104 3||eq3-->90 52 10
Values of xyz ented by user----->chrmosome 11 ||x=>6||y=>5||z=>2
Consolidated chromosome report eq1-->36 5 2||eq2-->24 40 6||eq3-->30 20 20
Values of xyz ented by user----->chrmosome 12 ||x=>2||y=>12||z=>15
Consolidated chromosome report eq1-->12 12 15||eq2-->8 96 45||eq3-->10 48 150
Values of xyz ented by user----->chrmosome 13 ||x=>12||y=>19||z=>3
Consolidated chromosome report eq1-->72 19 3||eq2-->48 152 9||eq3-->60 76 30
Values of xyz ented by user----->chrmosome 14 ||x=>18||y=>4||z=>20
```

```
Fitness calculated for the chromosome  7921.0  14400.0  2209.0  47961.0  225.0  4900.0  7921.0  15129.0  100.0  169.0  225.0  61009.0  1681.0  625.0  29
Maximum probabillity of selection is 0.0
For chromosome--->45
having fitness value-->0.0
Chromosome--->x=15 y=12 z=4
Chromosome selected are at position ------------>
0||2||3||4||5||6||7||8||9||11||13||14||15||17||18||19||20||21||22||23||24||25||26||27||28||30||31||33||34||36||38||39||40||41||42||43||44||45||46||47||4
Selected crossoverpoint-->
1||1||1||1||2||1||2||1||2||2||1||2||2||1||1||1||2||2||1||2||2||1||2||1||2||1||1||1||1||1||2||1||2||2||1||2||2||1||1||1||1||1||1||2||2||2||2||2||1||1||2|
Size of x-->100
Values in x array before crossover[17, 12, 20, 8, 2, 12, 8, 19, 18, 9, 4, 2, 15, 4, 4, 6, 18, 20, 19, 20, 17, 7, 7, 18, 0, 9, 18, 10, 19, 18, 13, 6, 9,
Size of y-->100
Values in y array before crossover[17, 18, 9, 11, 10, 19, 11, 12, 19, 5, 11, 10, 11, 11, 18, 5, 6, 14, 11, 10, 17, 5, 13, 19, 2, 2, 19, 5, 11, 19, 18, 1
Size of z-->100
Values in z array before crossover[9, 17, 8, 2, 10, 20, 2, 15, 20, 2, 1, 10, 1, 1, 17, 2, 10, 3, 19, 8, 9, 4, 0, 20, 16, 6, 20, 4, 19, 20, 13, 1, 4, 2,
Size of x100
Values in x array after cossover[17, 12, 20, 8, 2, 12, 8, 19, 18, 9, 4, 2, 15, 4, 4, 6, 18, 20, 19, 20, 17, 7, 7, 18, 0, 9, 18, 10, 19, 18, 13, 6, 9, 6,
Size of y-->100
Values in y array after cossover[9, 18, 11, 10, 19, 19, 12, 12, 5, 5, 11, 10, 11, 18, 18, 5, 6, 11, 10, 17, 17, 5, 19, 19, 2, 19, 19, 11, 11, 19, 16, 5,
Size of z-->100
Values in z array after cossover[8, 17, 2, 10, 20, 2, 15, 20, 2, 10, 1, 1, 1, 17, 2, 3, 10, 19, 8, 9, 4, 0, 20, 16, 6, 20, 4, 19, 13, 20, 1, 2, 4, 9, 3,
Value of mutationrate val-->69.0
Mutation Rate-->69
Random value generated 69
Random value generated [66, 30, 46, 0, 32, 39, 30, 7, 64, 67, 51, 3, 13, 49, 35, 24, 53, 46, 51, 11, 21, 39, 18, 32, 65, 4, 67, 13, 55, 63, 18, 41, 13,
Value of the filtered array newreplacelistval  [66, 30, 46, 0, 32, 39, 7, 64, 67, 51, 3, 13, 49, 35, 24, 53, 11, 21, 18, 65, 4, 55, 63, 41, 61, 19, 34,
Sum of the values of the new array is 1538
```

References

[1] Hoffman, J.D. and Frankel, S., 2001. *Numerical methods for engineers and scientists*. CRC press.

[2] Deolekar, R.V., 2016, March. Defining parameters for examining effectiveness of genetic algorithm for optimization problems. In *Computing for Sustainable Global*

Development (INDIACom), 2016 3rd International Conference on (pp. 1061-1064). IEEE.

[3] Bashir, L.Z., 2015. Solve Simple Linear Equation using Evolutionary Algorithm. *World Scientific News*, *19*, pp.148-167

[4] IKOTUN, A., AKINWALE, A. and AROGUNDADE, O., 2017. PARAMETER VARIATION FOR LINEAR EQUATION SOLVER USING GENETIC ALGORITHM. *Journal of Natural Sciences Engineering and Technology*, *15*(2), pp.42-50.

[5] Mardle, S. and Pascoe, S., 1999. An overview of genetic algorithms for the solution of optimisation problems. *Computers in Higher Education Economics Review*, *13*(1), pp.16-20.

.