RESEARCH PAPER | Engineering | Volume : 2 | Issue : 3 | December 2012 | ISSN - 2249-555X

Optimization of Linear Equations using Genetic Algorithms

KEYWORDS: Genetic algorithms, Fitness Function, Optimization problem, Mutation, Search space.

Roshni .V Patel
K-16, Aakansha Flats, Nr sola Railway crossing,

Jignesh. S Patel
K-16, Aakansha Flats, Nr sola Railway crossing,

ABSTRACT *For solving linear system of equations iterative algorithms are used. This paper proposes a new paradigm for solving systems of linear equations through using Genetic Algorithms (GA) techniques. Conventional numerical methods such as Gaussian Elimination method produce a set of solutions for a particular system of simultaneous linear equations, but genetic algorithm is capable to produce more than one set of solutions for certain systems of equations. In this paper we have used Genetic algorithms to find solution of linear equations since it is difficult to describe the solution set with infinitely many solutions. This study investigates the applicability and effectiveness of GA in finding optimal solution of systems of simultaneous linear equations, which includes a search for optimal values for the unknown variables in the equation that best fit the system of linear equation. To avoid the disadvantages like rounding errors, inverting large matrices, GA are introduced.*

Introduction

This paper aims to apply Genetic Algorithms as a non-linear technique in solving linear equation system and to investigate the major benefits obtained as a result of using genetic algorithms. There are several iterative algorithms for solving linear system of equations but the problem of solving such system with using Genetic Algorithm based approaches is analyzed and studied here in this paper. Normally, GA are simulation programs which create an environment where population of data can compute and only the fittest survive, sort of evolution on a computer. Genetic algorithms are excellent for all tasks requiring optimization and highly effective in any situation where many inputs (variables) interact to produce a large number of possible outputs (solutions).

Major purpose of this paper is to apply GA as a soft computing technique in solving the linear equation system and investigating the returned benefits as a result of using Genetic algorithms. First of all ,Genetic algorithms are computer programs which create an environment where populations of data can compete and only the fittest survive, sort of evolution on a computer.GA is excellent. Some situations where it plays major role are Portfolio balancing, Forecasting, Budgeting, Investment analysis, Distribution, Scheduling, Project management, task assignment, Clustering, Path finding and ordering, Data fitting etc. It is difficult to describe the solution set of linear system with infinitely many solutions. To avoid the disadvantage of solving large systems of linear equations. The genetic algorithms are introduced and discussed in details in this paper.

A Genetic algorithm is a problem solving method that is inspired by its genetics as its model of problem solving. A genetic algorithm (GA) is a search heuristic that mimics the process of natural evolution. This heuristic is routinely used to generate useful solutions to optimizations and search problem

This first population must offer a wide diversity of genetic materials. The gene pool should be as large as possible so that any solution of the search space can be propagated. Then, the genetic algorithm loops over an iteration process to make the population evolve. Each iteration consists of selection, crossover, mutation and replacement. This population will now traverse through search space by utilizing three operators Selection, Crossover, Mutation. Each of the individuals in the population undergoes fitness evolution, i.e. Objective function. The resulting number (fitness) is the quantitative measure of solution "goodness" and it represents the only connection between genetic algorithm and the problem solved. Individual fitness directly affects the probability of its survival.

Selection: It is a process of choosing two parents from the population for crossing. After deciding on an encoding, the next step is to decide how to perform selection. Some of the selection methods like Roulette wheel Selection, tournament Selection, Elitism etc are used.

Now from the current population, the algorithm selects individuals that will proceed (survive) to the next generation. By random selection of N individuals, temporary population is made from the population. Probability of selecting each individual is biased by its fitness- fittest individual has greater chance to be selected. Each individual can be selected more than once. Here individuals are selected based on a probability of selection given in equation-1 where f(parent) denotes the fitness of the ith parent.

Selections = F(parenti) / Ei f(parenti).....(1)

Objective function is to be strictly positive in this selection method, if this is hard to achieve, there are different selection schemes, which do not impose such constraint.

String no.	Initial population	x Value	Fitness $f(x) = x^2$	$Prob_i$	Expected count	Actual count
1	0 1 1 0 1	13	169	0.14	0.58	1
2	1 1 0 0 0	24	576	0.49	1.97	2
3	0 1 0 0 0	8	64	0.06	0.22	0
4	1 0 0 1 1	19	361	0.31	1.23	1
Sum			1170	1.00	4.00	4
Average			293	0.25	1.00	1
Max			576	0.49	1.97	2

Table 1- Roulette wheel Selection

Crossover: Crossover is a genetic operator that combines (mates) two chromosomes (parents) to produce a new offspring (chromosomes). The simplest kind is single-point crossover, where a random location (locus) in parent chromosomes is selected, and then exchanges the portion of chromosome preceding the selected point from parent1 to child1 and from parent2 to child2. Portion of chromosome following the selected point is copied from parent1 to child2 and from parent2 to child1. After the crossover, children are put back into temporary population in place of their parents. Pcross is set to an arbitrary value and typically high probability values within range [0.6, 0.8] have been to work best in most situations.

Volume : 2 | Issue : 3 | December 2012 | ISSN - 2249-555X

String no.	Mating pool	Crossover point	Offspring after xover	x Value	Fitness $f(x) = x^2$
1	0 1 1 0 | 1	4	0 1 1 0 0	12	144
2	1 1 0 0 | 0	4	1 1 0 0 1	25	625
2	1 1 | 0 0 0	2	1 1 0 1 1	27	729
4	1 0 | 0 1 1	2	1 0 0 0 0	16	256
Sum					1754
Average					439
Max					729

Table 2-Parent crossover

Mutation: Mutation is a genetic operator used to maintain genetic diversity from one generation of chromosomes to the next. After crossover the strings are subjected to mutation, it is performed to one individual to produce a new version of it, where some of the original genetic material has been randomly changed. It consists of random altering of bit values inside chromosome with given probability. Random number P is generated for each allele in each chromosome and it satisfies relation P < Pmut is selected allele is altered. Pmut is the probability of mutation and it is typically set to low values within the range [0.01, 0.1]. Finally, the current population with temporary population and repeat the process until the termination condition (such as closeness to the solution is replaced, maximum number of generators, maximum number of simulations and so on) is met.

String no.	Offspring after xover	Offspring after mutation	x Value	Fitness $f(x) = x^2$
1	0 1 1 0 0	1 1 1 0 0	26	676
2	1 1 0 0 1	1 1 0 0 1	25	625
2	1 1 0 1 1	1 1 0 1 1	27	729
4	1 0 0 0 0	1 0 1 0 0	18	324
Sum				2354
Average				588.5
Max				729

Table 3-Offspring recombination

Genetic Algorithms: A genetic algorithm is a heuristic search technique used in finding true or approximate solutions to optimization and search problems, originally motivated by Darwinian principle of evolution through (genetic) selection. Each chromosome represents a solution to a problem and has fitness. Starting with a randomly generated population of chromosomes, a genetic algorithm carries out a process of fitness-based selection and recombination to produce a successor population, the next generation. During recombination, selecting parent chromosomes and their genetic material is recombined to produce child chromosomes. These then pass into the successor population. As this process is iterated, a sequence of successive generation evolves and the average fitness of the chromosomes tends to increase until some stopping criterion is reached. So in this way genetic algorithm "evolves" a best solution to the given problem. Genetic algorithm is very general algorithms that work well in any search space. A genetic algorithm is able to create a high quality solution. The algorithm are

- BEGIN
- INITIALISE population with random candidate solution.
- EVALUATE each candidate
- REPER UNTIL(termination condition)is satisfied DO

1. SELECT parents;
2. RECOMBINE pairs of parents;
3. MUTATE the resulting offspring;
4. SELECT individuals or the next generation;

- END

Biological Background: All the living organisms consist of cells. Chromosomes' characteristics are determined by the genes. Each gene has several forms of alternatives, which are called alleles. The set of chromosome is called genotype, which defines a phenotype (the individual) with certain fitness. During reproduction first occurs recombination (or crossover). Genes from parents form in some way the whole new chromosome. The new created offspring can then be mutated. The fitness of an organism is measured by success of the organism in its life. According to Darwinian Theory the highly fir individuals are given opportunities to "reproduce" whereas the least fit members of the population are less likely to get selected for reproduction, and so "die out".

System of Linear Equations: A system of equations is a collection of two or more equations with the same set of unknowns. In solving a system of equations, we try to find values for each of the unknowns that will satisfy every equation in the system. A system of linear equations (or linear system) is a collection of linear equations involving the same set of variables.

Vector equation: Each unknown is a weight for a column vector in a linear combination.

$$x_1 \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{m1} \end{bmatrix} + x_2 \begin{bmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{m2} \end{bmatrix} + \cdots + x_n \begin{bmatrix} a_{1n} \\ a_{2n} \\ \vdots \\ a_{mn} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

Fig 4- Vector equation

This allows all the language and theory of vector spaces (or more generally, modules) to be brought to bear.

For example, the collection of all possible linear combinations of the vectors on the left hand side is called their span, and the equations have a solution just when the right-hand vector is within that span. If every vector within that span has exactly one expression as a linear combination of the given left-hand vectors, then any solution is unique. In any event, the span has a basis of linearly independent vectors that do guarantee exactly one expression; and the number of vectors in that basis (its dimension) cannot be larger than m or n, but it can be smaller.

Solution set:
A solution of a linear system is an assignment of values to the variables x1, x2, , ..., xn such that each of the equations is satisfied. The set of all possible solutions is called the solution set.

A linear system may behave in any one of three possible ways:
1. The system has infinitely many solutions.
2. The system has a single unique solution.
3. The system has no solutions.

THE PROPOSED METHODOLOGY OF SOLUTION:
Coding Scheme: In our attempt to develop a code for genetic algorithms, we had used cyclic distribution of input matrix and vector in a cyclic manner.

Process 0 generates the random number sets and distributes them to all processes. Each process calculates their values of and is reduced by addition operation to store in a matrix of process 0.

We have used various methods for combining the two selected parents from the population including Jacobi method, arithmetic crossover, real valued recombination etc

For each set of the population, an element is selected to be mutated. This is done by generating a Random Number rand between 0 to 1 and comparing it with the ratio i/n , where I ranges from 1 to n.

$$\boldsymbol{rand} \leq \frac{\boldsymbol{i}}{\boldsymbol{n}}, \boldsymbol{i} \in \mathbf{1}\ \boldsymbol{ton} \ldots\ldots\ldots\ldots (2)$$

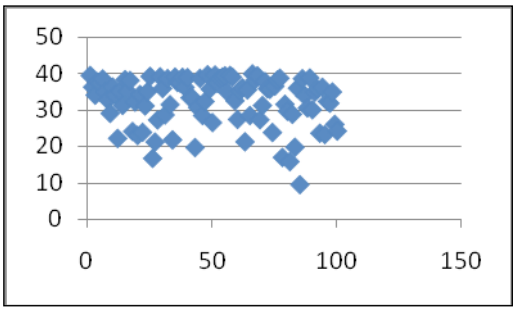
Volume : 2 | Issue : 3 | December 2012 | ISSN - 2249-555X

After the computation of the above inequality, ith element from the set is selected to be mutated.

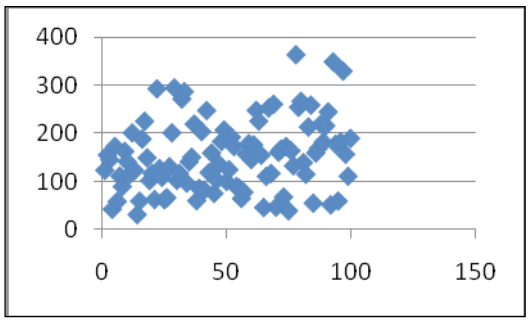
Fitness function
The fitness function to be minimized by the genetic algorithms is:

g = max [abs (fi)] for i =1,2,3,…,N Where:
-g is the fitness function to be minimized,
-max [abs (fi)] is the maximum absolute value of individual equations in the system f(x) = 0
-N is the number of equations in the system

Simulated Output results:



In the initial phase



After 300th iteration

We notice that we reach to the optimal solution when the results are equal in the final generations, so we stopped it and consider it as final output. Here we have seen that the solution is almost equal to the analytical one. Nevertheless on using G.A to solve linear system, we obtain the result very swiftly.

Conclusion and Future works
How genetic algorithms can be successfully applied to solve linear equation has been shown in this paper. By using the specifically defined Fitness function f(x) for the given set of linear equations, the random numbers can be mutated or crossed over forming new random numbers which will be more in sync with the solution that is needed for the set of linear equations. By numerous iterations of the same, we hope to get closer to the required solution set which would otherwise impossible to find by the traditional methods. The result of using genetic algorithms compared to the exact solutions obtained by some different numerical methods turned to be confirming.

As a future work we plan to apply genetic algorithms to training and designing artificial intelligence systems such as artificial neural networks, prediction of 3 dimensional protein structure. Also the same technique can be used and applied to more complex non-linear systems.

REFERENCE 1. RC Chakra borty (2010) 'Genetic Algorithms and Modeling' –Soft Computing Course. | 2. Randy L. Haupt and Sue Ellen Haupt (2004) 'Practical Genetic Algorithms'. | 3. "Genetic Algorithms", Thomas Jefferson High School for Science and Technology, http://www.tjhsst.edu/~ai/AI2001/GA.htm. | 4. Jähne, B., "Digital Image Processing", Springer – Verlag Berlin, Heidelberg , 2002. | 5. McCall, J., "Genetic Algorithms For Modeling And Optimization", Journal of Computational and Applied | 6. "Genetic Algorithms", Thomas Jefferson High School for Science and Technology | 7. Sandikci, B., "Genetic Algorithms", http://www.ie.bilkent.edu.tr/~Lors/ie572/barhan eddin.pdf, accessed August 2009 | 8. Renner, G., and A. Ekárt , "Genetic Algorithms In Computer Aided Design " , Computer Aided Design 35 , pp. 709 – 726 , 2003. | 9. Barry Wilkinson and Michael Allen (2006) 'Parallel Programming-Techniques and Applications. | 10. Sahoo, R.K., T. Banerjee, S.A. Ahmed and A. Khanna, 2006. Improved Binary Parameters Using GA for Multi-Component Aromatic Extraction.