A Robust Method for Solving Transcendental Equations

Md. Golam Moazzam, Amita Chakraborty and Md. Al-Amin Bhuiyan

Department of Computer Science and Engineering,
Jahangirnagar University, Savar, Dhaka-1342, Bangladesh.

Abstract

This paper provides a robust method for solving transcendental equations. The approach, based on the Genetic Algorithm, is commenced with the evaluation of the mathematical equations by their fitness ratio. As the genetic algorithm is a computationally expensive process, the searching space for possible solutions is limited to possible chromosomes for which the function values are closest to zero. These chromosomes are then selected for the next generation using roulette wheel strategy. Thus, the required processing timing is greatly reduced. Experimental results demonstrate that this process works reliably and gives better results than the well-known False Position method and the Bisection method for the numerical solution of transcendental equations.

Keywords: *Genetic Algorithm, False Position, Bisection, Transcendental Equation, Cross-over, Mutation, Roulette Wheel Selection.*

1. Introduction

Scientists and Engineers are often faced with the task of finding out the roots of the non-algebraic equations such as trigonometric, exponential, and logarithmic functions. These equations are known as transcendental equations[1].

Various approaches to find the roots of non-linear equations are made over the last few decades, such as direct analytical methods, trial and error methods, iterative methods [1], [2]. However, each of them imposes some constraints. Direct analytical method miserably fails to solve transcendental equations [2]. Trial and error methods suffer from the problem of convergence to local optima. This method involves a series of guesses for *x*, each time evaluating the function to see whether it is close to zero. Iterative methods, such as the Bisection method, the False Position method usually start with an appropriate value of the root, known as the *initial guess*, which is then successively corrected iteration by iteration [3]. So, these methods are also cumbersome and time consuming. Moreover, the accuracy of the results does not suit ideally to the requirements of many engineering and scientific problems.

This research explores an efficient method for finding the roots of the transcendental equations with greater precision. Genetic Algorithms (GAs) belong to a class of stochastic search method employed by natural population genetics. They perform a highly parallel adaptive search process. GAs have been successfully employed in a wide variety of problems related to parameter optimization [4], [5], pattern recognition, image processing [6], [7] and so on. In this paper, an attempt has been made to solve transcendental equations using Genetic Algorithm and to compare its efficiency and accuracy with that of the traditional False Position and Bisection methods.

2. GA to Solve Transcendental Equations

GA is a blind search technique applied to possible solutions to a problem on the mechanics of natural selection and genetics analogous to natural evolution [4], [8]. Central to the idea of GA is a population of individuals, each representing a possible solution to the given problem. Each individual, known as chromosome (usually represented by a bit string consisting of 0s and 1s), is assigned to a fitness value based on how good their solution to the problem is. The individuals then evolve through successive iterations called generations. During one generation, highly fit individuals are given the opportunity to mate with other individuals in the population. Since the least fit individual in the population are less likely to get selected for mating, they disappear from future generations. As a result, the population of individuals converges to an optimal solution to the problem.

The GA starts with an initial set of random solutions called the population. Each chromosome is assigned a fitness value depending on how meaningful its solution to the problem is. On its fitness allotment, the natural selection gets executed and the 'survival of the fittest chromosome' can prepare to breed for the next generation. A new population is then generated by means of genetic operations: *cross-over* and *mutation*. This evolution process is iterated until a near-optimal solution is obtained or a given number of generations is reached. Different steps employed in the genetic algorithm towards solving transcendental equations is given below.

2.1 Fitness Function

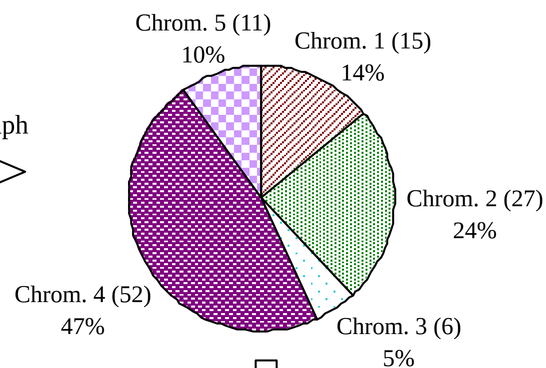
In order to identify the best individual during the evolutionary process, a function needs a degree of fitness to each chromosome in every generation. In our problem, the transcendental equation x^2-x-2=0 is taken. Thus, the fitness function $f(x) = x^2 - x - 2$ determines the fitness value of each chromosome.

2.2 Selection

Selection operator is a process in which chromosomes are considered a mating pool according to their fitness function. The chromosomes with fitness values closer to zero are treated as highly fit individuals for the next generation. The other chromosomes are maintained by a

Population	Fitness
Chromosome 1	15
Chromosome 2	27
Chromosome 3	6
Chromosome 4	52
Chromosome 5	11

Fitness graph



Randomly generated number=21

Chromosome 2 is selected

Fig. 1 Roulette wheel selection process

roulette-wheel selection process [8]. In roulette wheel selection, as shown in Fig. 1, each chromosome is given a slice of a circular roulette wheel. The area of the slice within the wheel equals the chromosome fitness ratio.

To select a chromosome for mating, a random number is generated in the interval [0, 1]. It is like spinning a roulette wheel where each chromosome has a segment on the wheel proportional to its fitness. The roulette wheel is spun, and when the arrow comes to the rest on one of the segments, the corresponding chromosome is selected.

2.3 Cross-over

Cross-over operator randomly chooses a crossover point where two parent chromosomes ‘break’, and then exchanges the chromosome parts after that point. As a result, two offspring are generated by combining the partial features of two chromosomes. If a pair of chromosomes does not cross over, the chromosome cloning takes place, and the offspring grow as exact copies of each parent. Here we deals with single point cross-over, two point cross-over and uniform cross-over operators for 8-bit chromosomes as shown in Fig. 2. The cutting points are selected randomly within the chromosome for exchanging the contents.

2.4 Mutation

Mutation, which is rare in nature, brings about a change in the gene and averts crisis in genetic diversity. Its role is to guarantee that the search algorithm is not trapped on a local optimum.

This operator alters a randomly selected gene of chromosome with a very low probability, P_M. For each chromosome generates a random value between [0, 1]. If the random value is less than P_M, choose a bit at a random location to flip its value from 0 to 1, or 1 to 0. The fundamental steps employed in the genetic algorithm to solve transcendental equations are shown in Fig. 3.

The parameter settings of the proposed approach is shown in Table 1.

Table 1. Parameter settings for GA

Chromosome Length	8 bits
Population Size	12
Number of Generation	7
Cross-over Probability, P_C	0.7
Mutation Probability, P_M	0.01

3. False Position Method

The False Position method also known as the linear interpolation method states that if $f(x)$ is real and

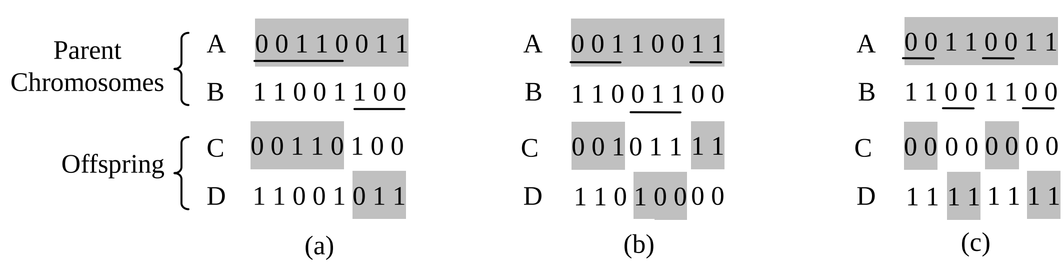


Fig. 2 (a) Single point cross-over, (b) Two point cross-over, and (c) Uniform cross-over

IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 6, No 2, November 2012
ISSN (Online): 1694-0814
www.IJCSI.org

continuous in the interval $a<x<b$, and $f(a)$ and $f(b)$ are of opposite sign, that is $f(a).f(b)<0$, then there is at least one root in the interval between a and b [1], [2], [3]. Take $x_1=a$ and $x_2=b$. Let us join the points x_1 and x_2 by a straight line. This method defines the point intersection of this line with x-axis such that $x_0 = x_1 - \frac{f(x_1)(x_2 - x_1)}{f(x_2) - f(x_1)}$. This point is called the false position of the root. It then replaces one of the initial guesses ($x_1$or x_2) that has a function value of the same sign as $f(x_0)$. This process is repeated with the new values of x_1 and x_2.

4. Bisection Method

The Bisection method also known as binary chopping or half-interval method states that if $f(x)$ is real and continuous in the interval $a<x<b$, and $f(a)$ and $f(b)$ are of opposite sign, that is $f(a).f(b)<0$, then there is at least one root in the interval between a and b [1], [2], [3]. Take $x_1=a$ and $x_2=b$. This method defines another point x_0 such that $x_0 = \frac{x_1 + x_2}{2}$. Now three situations arise: i) If $f(x_0)=0$, we have a root at x_0; ii) If $f(x_0).f(x_1)<0$,

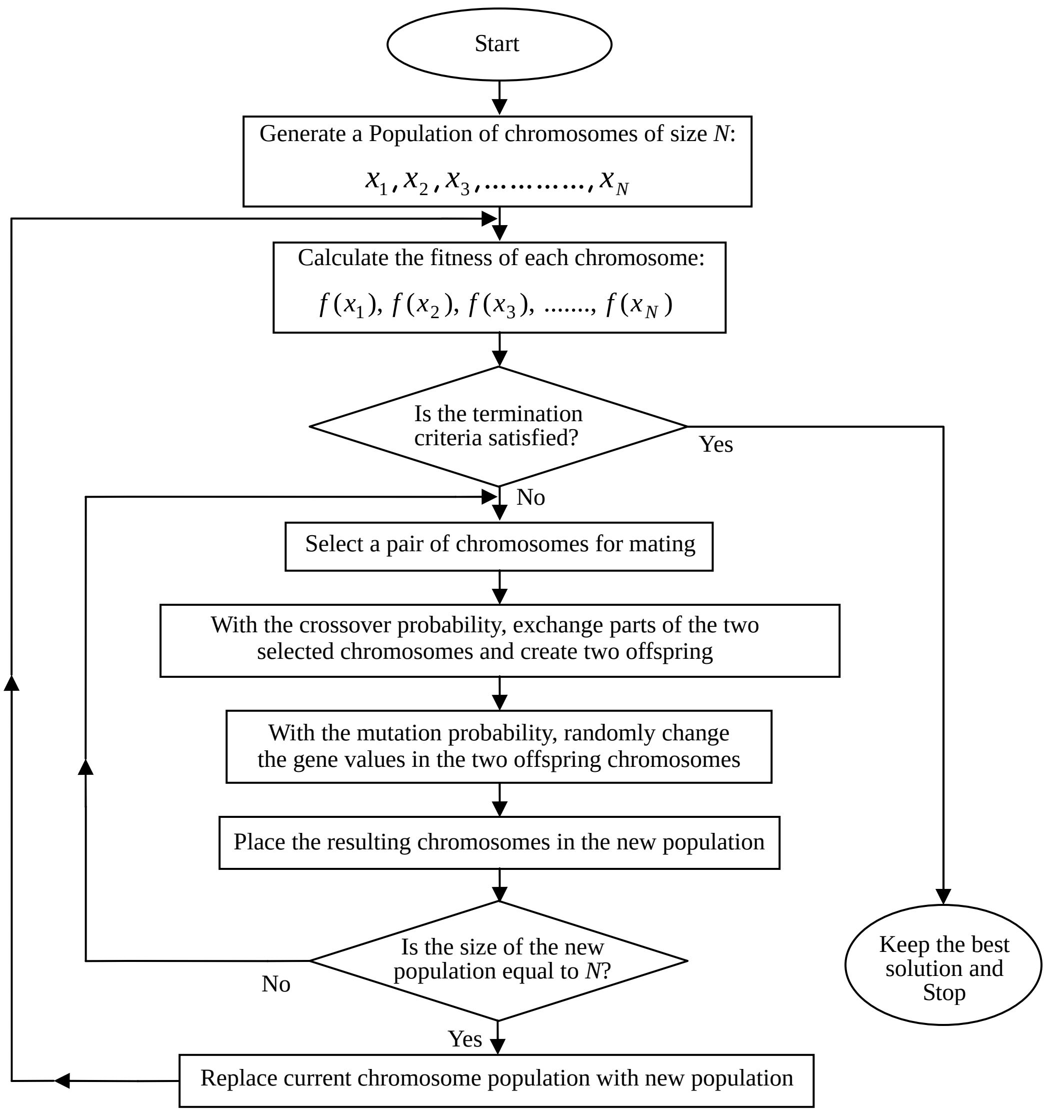


Fig. 3 Flowchart of the Genetic Algorithm

there is a root between x_0 and x_1; iii) If $f(x_0).f(x_2)<0$, there is a root between x_0 and x_2. By testing the sign of the function at the mid point, we can conclude which part of the interval contains the root.

In this paper, the False Position method and the Bisection method are applied to solve the same transcendental equation. The results are then compared with that of GA in Table 2.

Table 2. Comparison between search spaces for GA, False Position and Bisection methods

No. of Iteration	Search Space					
	GA Method		False Position Method		Bisection Method	
	Left bound	Right bound	Left bound	Right bound	Left bound	Right bound
1	1.000000	4.000000	1.000000	4.000000	1.000000	4.000000
2	1.929412	2.047059	1.500000	4.000000	1.000000	2.500000
3	1.985236	2.014302	1.777778	4.000000	1.750000	2.500000
4	1.999712	2.006209	1.906977	4.000000	1.750000	2.125000
5	1.999712	2.001267	1.962085	4.000000	1.937500	2.125000
6	1.999712	2.000084	1.984718	4.000000	1.937500	2.031250
7	2 (root)		1.993869	4.000000	1.984375	2.031250
8			1.997544	4.000000	1.984375	2.007813
9			1.999017	4.000000	1.996094	2.007813
10			1.999607	4.000000	1.996094	2.001953
11			1.999843	4.000000	1.999023	2.001953
12			1.999937	4.000000	1.999023	2.000488
13			1.999975	4.000000	1.999756	2.000488
14			1.999990	4.000000	1.999756	2.000122
15			1.999996	4.000000	1.999939	2.000122
16			1.999998	4.000000	1.999939	2.000031
17			2 (root)		1.999985	2.000031
18					1.999985	2.000008
19					1.999996	2.000008
20					1.999996	2.000002
21					2 (root)	

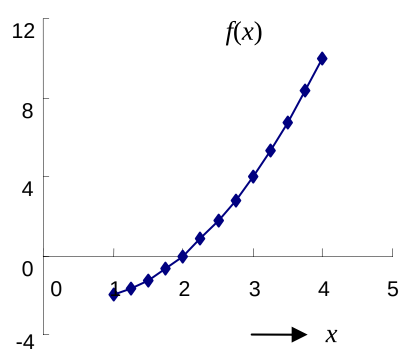


Fig. 4 Graph of the function, $f(x)=x^2–x–2$

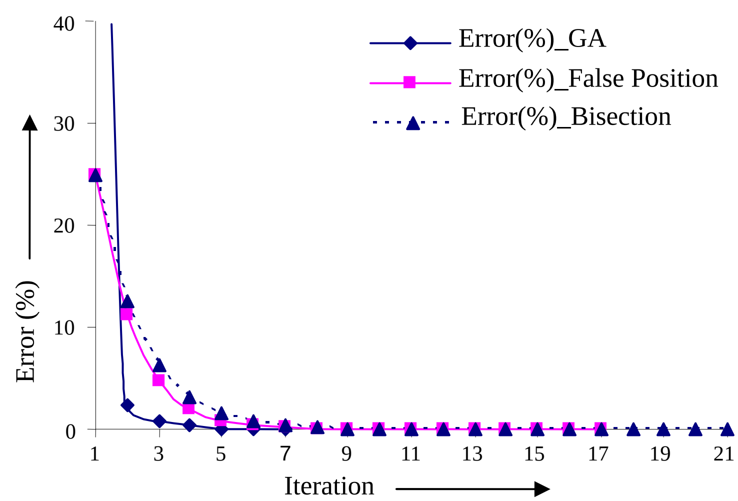


Fig. 5 Error versus iteration curves for solving the transcendental equation

5. Experimental Results

The effectiveness of the GA method is justified with the transcendental equation, $x^2 - x - 2 = 0$ (Fig. 4). Use is made of an approximation to ensure the existence of the root of the equation within the range between 1 and 4.

Table 2 reveals the same level of accuracy. The False Position and the Bisection methods require at least 17 and 21 iterations, respectively whereas the GA method needs only 7 iterations. On the other way, it can be said that it is possible to achieve higher accuracy using GA with the same number of iterations as False Position and Bisection methods. Therefore, GA is more time saving. Fig. 5 shows the change of errors in the solution of the transcendental equations for the three methods. It demonstrates that GA converges rapidly to optimum solution compared to the False Position and the Bisection methods.

Experiments were performed in a Pentium IV 1.70 GHz using Microsoft Visual C++. The algorithm is examined using single point, two points, and uniform cross-over with population sizes: 8, 10, and 12, respectively. The number of iterations required for different population size is shown in Table 3. It reveals that larger population size offers better performance for uniform cross-over because of the larger pool of diverse schemata available in the chromosome. Therefore, we adopt uniform cross-over with a population size of 12 in our approach.

Table 3. Iterations required for different population sizes

Population Size	Iteration
8	32
10	11
12	7

6. Conclusion

In this paper, we have proposed a robust method for solving transcendental equations based on genetic algorithm. The effectiveness of the algorithm is justified by applying it on a number of non-linear equations. It has been observed that the algorithm is capable enough of finding the solution of transcendental equations with greater precision compared to the False Position and the Bisection methods.

References

[1] S. C. Chapra and R. P. Canale, *Nunerical Methods for Engineers*, 2nd Edition, McGraw-Hill Book Company, 1990.
[2] E. Balagurusamy, *Numerical Methods*, Tata McGraw-Hill Publishing Company Limited, New Delhi, India, 1999.
[3] S. S. Shastri, *Introductory Methods of Numerical Analysis*, 3rd Edition, Prentice Hall of India Pvt. Ltd., 2000.
[4] D. E. Goldberg, *Genetic Algorithms in Search Optimization, and Machine Learning*, Addison-Wesley, 1989.
[5] Janiko, E. Z. A Knowledge Intensive Genetic Algorithm for Supervised Learning, *Machine Learning*, 13, pp. 189-228, 1993.
[6] Yuille, Alan, Cohen, David, Hallinan, and Peter, Feature extraction from faces using deformable templates, *Proc. IEEE Computer Soc. Conf. on computer Vision and Pattern Recognition*, p. 104-109, 1989.
[7] Wang, Jianguo, Tan, and Tieniu, A new face detection method based on shape information, *Pattern Recognition Letters*, 21(6), 1999, 463-471.
[8] M. Mitehall, *An Introduction to Genetic Algorithms*, Prentice-Hall of India. 1998.

Md. Golam Moazzam completed his B.Sc (Hons) in Electronics and Computer Science from Jahangirnagar University in 1997 and M.S. in Computer Science and Engineering from the same University in 2001, respectively. He is now an Associate Professor in the Dept. of Computer Science and Engineering, Jahangirnagar University, Dhaka, Bangladesh. His research interests include Digital Image Processing, Artificial Intelligence, Computer Graphics, Neural Networks, Computer Vision and so on.

Amita Chakraborty received her B.Sc. (Hons) in Electronics and Computer Science and MS in Computer Science and Engineering from Jahangirnagar University in 1997 and 2002 respectively. She is now an Assistant Professor in the Dept. of Computer Science and Engineering, Shaikh Burhanuddin College, Dhaka, Bangladesh. Her research interests include Artificial Intelligence, Object-Oriented Programming, Computer Graphics, Image processing and so on.

Md. Al-Amin Bhuiyan received his B.Sc (Hons) and M.Sc. in Applied Physics and Electronics from University of Dhaka, Dhaka, Bangladesh in 1987 and 1988, respectively. He got the Dr. Eng. degree in Electrical Engineering from Osaka City University, Japan, in 2001. He has completed his Postdoctoral in the Intelligent Systems from National Informatics Institute, Japan. He is now a Professor in the Dept. of CSE, Jahangirnagar University, Savar, Dhaka, Bangladesh. His main research interests include Image Face Recognition, Cognitive Science, Image Processing, Computer Graphics, Pattern Recognition, Neural Networks, Human-machine Interface, Artificial Intelligence, Robotics and so on.