Published Online First: August, 2024
https://doi.org/10.21123/bsj.2023.7481
P-ISSN: 2078-8665 - E-ISSN: 2411-7986
Baghdad Science Journal

A New Algorithm for Finding the Roots of Nonlinear Algebraic Equations

***Ahmad yousef Alrazzo*[1*], *Nasr Al Din Ide*[2], *Mohammad Assaad*[3]**

[1]Department of Mathematics, Faculty of Science, University of Aleppo, Aleppo, Syria
[2]Department of Mathematics, University of Tishreen, Latakia, Syria.
*Corresponding Author.

Received 01/06/2022, Revised 25/02/2023, Accepted 27/02/2023, Published Online First 20/08/2024

Abstract

In this paper, the algorithm (Stochastic Gradient Descent) SGD, which is one of the most famous optimization algorithms, was hybridized with genetic algorithms in finding the roots of non-linear equations, which is one of the most important mathematical problems due to its application in all sciences. Genetic algorithms are used here to find the optimal primary root of SGD algorithm and its application in reducing the studied objective function. Some famous algorithms need initial point to reach the solution in terms of stability. The proposed algorithm is tested on several standard functions and the results are compared with the famous algorithms, and the results show the efficiency of the proposed algorithm through tables and figures.

Keywords: Genetic algorithms, Nonlinear equations, Objective function, Optimizations, SGD algorithm.

Introduction

In most of the problems related to engineering and applied sciences, the problems come to a non-linear equation to be studied in the form:

$$f(x) = 0 \qquad 1$$

Such as problems that require finding critical values and problems that search for eigenvalues by minimizing the objective function[1]. Most of the numerical analysis methods depend in their development on Newton's iterative method, which is to give a starting point to find the root of the studied function[2,3].

Many papers have worked in finding the roots of Eq. 1 in several ways, including (Newton's method, partition method, Bisection method, Regula Falsi, Nonlinear Regression) and other modern methods[4-7].

Algorithms that search for the roots of a nonlinear algebraic equation are divided into two parts the first section is: the algorithms that are made with a certain number of steps and start with an initial value within the scope of the solution and with a number of iterations, the solution is reached, but inaccurately and with a large error.

The second section of algorithms: that depend on classification, which is the fastest in finding roots, and this method was proposed in this research based on genetic algorithms that choose the best element to be a candidate as the root of the studied function based on generating an initial population. The selection process is carried out according to the proposed steps with the SGD algorithm[2].

In this paper, the genetic algorithm and the SGD algorithm were hybridized to solve Eq.1 by determining the appropriate starting point by generating an initial population of the genetic algorithm in the first step of its steps, which most other iterative algorithms suffer from in accurately reaching the desired solution.

Published Online First: August, 2024
https://doi.org/10.21123/bsj.2023.7481
P-ISSN: 2078-8665 - E-ISSN: 2411-7986
Baghdad Science Journal

Then the appropriate studied function of the SGD algorithm was configured and worked to reduce it and improve its learning rate by suggesting an update relationship in each iteration. Some numerical examples were also presented that confirm the theoretical results that allow to compare this method with other standard methods.

Genetic Algorithms:

The genetic algorithm is one of the general search algorithms based on the natural selection mechanism and the natural gene system that is used to solve complex problems. It was used by the scientist John Holland in 1975 at the University of Michigan[8] as he published many researches in this field. The main goal included the development of many algorithms, software and systems using this algorithm, and a genetic algorithm is known as a smart algorithm that depends carefully on the ideas of genetic engineering, which is characterized by the intended production of new individuals with desirable (good) characteristics through the intended switch and modification of inherited groups (adding certain genetic materials or replacing them) with the aim of forming individuals with good qualities. On this basis, the genetic algorithm selects the preferred solutions from a large number of solutions and makes some overlaps and alterations between these solutions in order to create better solutions. As for genetic research, it is the process of choosing a quality scale so that the genetic processes generate the required goals that to find.

The genetic algorithm shortened a lot of effort and time required by the designers of systems and programs, by finding a general algorithm that is reliable in various types of issues, instead of building a special algorithm for each issue, taking into account the necessary changes that are commensurate with the specificity of each issue in terms of the size, type and nature of the data used, objective function, and constraints for each problem.

The Algorithm:

In the genetic algorithm process is as follows[8]:

- Selection: The process of selecting parents from the community in order to intersect and produce a new generation.
- Crossover: This process is represented by a switch between the corresponding values of the two syllables of the elected parents in order to form the new syllable.
- Mutation: Mutation Operator: The key idea is to insert random genes in offspring to maintain the diversity in the population to avoid premature convergence.
- Solution (Best Chromosomes)

The flowchart of algorithm can be seen in Fig 1.

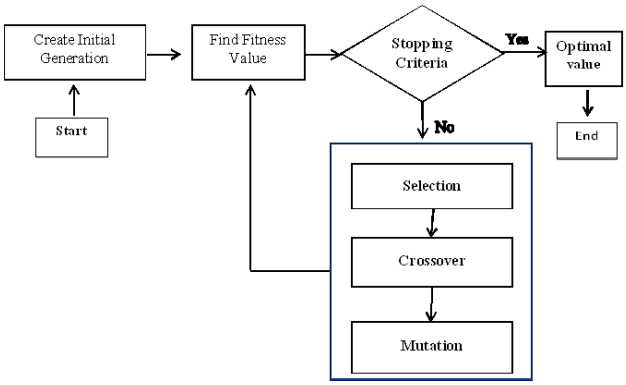

Figure 1. Genetic algorithm flowchart

Stochastic Gradient Descent:

The SGD algorithm is considered one of the most important optimization algorithms, which is also used in training artificial neural networks, which depends on the first derivative of the studied objective function and is considered the beginning of the development of other optimization algorithms.

But it is related to the learning rate, which takes a fixed value during the algorithm iteration process, so it makes the algorithm slow and may not reach the required solution, and many researchers are working on developing it, such as the Adagrad, Adam algorithms[9].

2. Make a guess θ_0 for θ.
3. Grate a set of labeled training data $\{(x_i, y_i)\}_{i=1}^{N_{training}}$
4. Choose suitable batch size $N \leq N_{training}$
5. Choose suitable learning rate $\Delta t > 0$
6. For $m \in \{1,2,\dots,M\}$ do
   - Choose random $\{j_k\}_{k=1}^{N}$ from $\{1,2,\dots,N_{training}\}$
   - Compute $\theta_{m+1} = \theta_m - \Delta t \frac{1}{N} \nabla_\theta \sum_{i=1}^{N} f(\alpha(x_{jk}, \theta_m), y_{jk})$

Which it's called stochastic gradient descent (SGD) [9]. If $N = N_{training}$ get standard GD instead.

Published Online First: August, 2024
https://doi.org/10.21123/bsj.2023.7481
P-ISSN: 2078-8665 - E-ISSN: 2411-7986
Baghdad Science Journal

At the beginning of the algorithm the parameter values are defined which are the learning rate Δt and batch size N and in each batch of training data the learning rate value is improved.

Learning Rate:

The learning rate[10] is the most important measure in the algorithms for searching for the desired solution, which expresses the amount of the step in each search process or the transition from one solution to another to be more accurate, so if the step amount is relatively large, the exact solution is passed along the graph of the studied function. The amount of the step is rather small, the solution is reached accurately, but the algorithm takes more steps and more time, so many researchers work to estimate the learning rate, either by inference or by giving it an appropriate and fixed value in each iteration.

In this research, an appropriate function has been proposed to generate an appropriate value for the learning rate in each iteration, its value between zero and one, while preserving the value of the learning rate so that it does not converge to zero, because that in turn stops the algorithm without reaching the required solution.

The appropriate relationship for the learning rate has been proposed, thus taking advantage of the increase in the exponential function, as follows:

$$m = \exp\left(-\frac{1}{t^2}\right), \quad t = 1,2,\ldots \qquad 2$$

where t: represents the iteration counter and m is learning rate.

The learning rate values over several iterations can be illustrated in Table 1:

Table 1. Values of the learning rate in 15 iterations

Iter	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Lr	0.54	0.66	0.73	0.78	0.82	0.85	0.87	0.88	0.89	0.89	0.90	0.91	0.92	0.95	0.98

Table 1 shows the increasing values of the learning rate(lr) in each iteration(iter) by a small amount, which in turn leads to an acceleration of the algorithm to reach the solution, and this is better than the fixed value for it from the beginning of the algorithm.

The learning rate graph can be plotted during the working phase of the algorithm as shown in Fig.2.

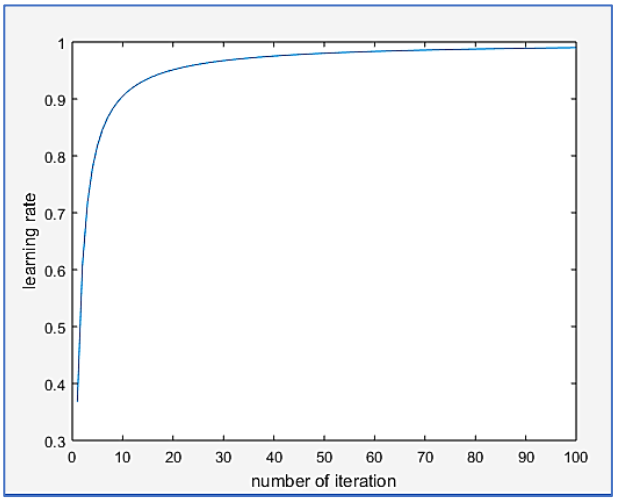

Figure 2. Graph of the learning rate for 100 iterations.

Proposed Algorithm:

To find the roots of a function f(x), assuming the cost function $L(x)$ and work to reduce this function as follows:

$$L(x) = \frac{1}{2}(f(x) - y_0)^2 \qquad 3$$

where $y_0$is the root of the function and in our case its value is equal to zero and try from the genetic algorithms to find all the roots of the studied function by generating an elementary community and evaluating this community through the matching function of genetic algorithms and thus get the optimal initial roots to get rid of randomness in giving primitive values to it that are far from the solution and then apply the SGD algorithm to reduce the cost function to the smallest possible and thus get the exact root.

Proposed Algorithm Steps:

1- Entering the objective function, **Popsize**, determining the solution field [a, b], learning rate $\boldsymbol{\alpha}$, mutation probability **Pm** (value between zero and one), number of **Genno** iterations.
2- Populating the population with random values with values in the binary system.
3- Converting the elements of society into values in the decimal system within the scope of the solution.

Published Online First: August, 2024
https://doi.org/10.21123/bsj.2023.7481
P-ISSN: 2078-8665 - E-ISSN: 2411-7986
Baghdad Science Journal

4- Determining an initial value for the number of cycles represented by the variable $g = 1$.
5- Calculating the elements of the new society in relation to:

$$\boldsymbol{pop} = \boldsymbol{pop} - \boldsymbol{\alpha} * \boldsymbol{L'(pop)} \quad (4)$$

Where: α represents the learning rate, L' is the derivative of the studied objective function.
6- Computing the fitness function:

$$\boldsymbol{p(x_i)} = \frac{f(x_i)}{\sum_{i=1}^{n} f(x_i)} \quad (5)$$

Where: $\boldsymbol{x_i}$ is the population, and n is the number of the population.

7- Determining an initial value for counter I.
8- The selection process was carried out based on the matching function, and the selection was applied in the form of Tournament Selection.
9- Crossover was performed and the two-point crossover was applied.
10- Mutation was performed and the bit-inversion mutation was applied.
11- Increasing the value of the counter I by one.
12- Checking counter I if it is less than (**Popsize**) Return to step 8.
13- Increasing the value of the g counter by one.
14- Checking stop criterion as g is compared with **Genno** (number of cycles entered) if g is less than **Genno** Return to step 6.
15- end of the algorithm.

Results and Discussion

Compare Results:

The programs were written using MATLAB 2016 program and the proposed method was applied to some test functions[11] shown in Table 2 with the initial point and exact root of each standard function, which most researchers adopt in testing their new methods. The results were compared with the most popular standard algorithms such as with the Newton's method (NM)[11], the Weerakoon-Fernando method (WFM)[11], Glis'ovic'et al. method (GOM)[12], the Kou-Li-Wang method (KLWM)[13], Wang's method (WM)[6], Zalinescu Method[14]. The stopping criteria was used $|f(x_{n+1})| < \varepsilon$, where $\varepsilon = 10^{-15}$.

Table 2. The test functions and their initial point and root α

f(x)	x_0	A
$f_1(x) = x^2 - e^x - 3x + 2$	3	0.257302854…
$f_2(x) = xe^{x^2} - \sin^2 x + 3\cos x + 5$	-2	−1.207647827 …
$f_3(x) = e^{x^2+7x-30} - 1$	3.25	3
$f_4(x) = \ln(x^2 + x + 2) - x + 1$	3	4.152590736 …

Finding the initial root of the function $f_1(x)$ within the range $[0,4]$ in the Table 3 using genetic algorithm:

Table 3. Fitness value f(x), probability p(x) of selection in the next population:

Initial population	f(x)	P(x)
0.1544	0.0775	0.0091
0.3813	0.1070	0.0126
0.1611	0.0677	0.0080
0.7581	1.6814	0.1983
0.8711	2.5178	0.2970
0.3508	0.0611	0.0072
0.6855	1.2346	0.1456
0.2941	0.0095	0.0011
0.5306	0.5103	0.0602
0.8324	2.2116	0.2608

The table shows a set of initial values for the initial population whose values range within the scope of the solution with the function values for each value and the third column represents the probability value for each value where the value with the greatest probability is chosen using the selection step of the genetic algorithm. The selection process according the roulette wheel method in Fig 3.

Published Online First: August, 2024
https://doi.org/10.21123/bsj.2023.7481
P-ISSN: 2078-8665 - E-ISSN: 2411-7986
Baghdad Science Journal

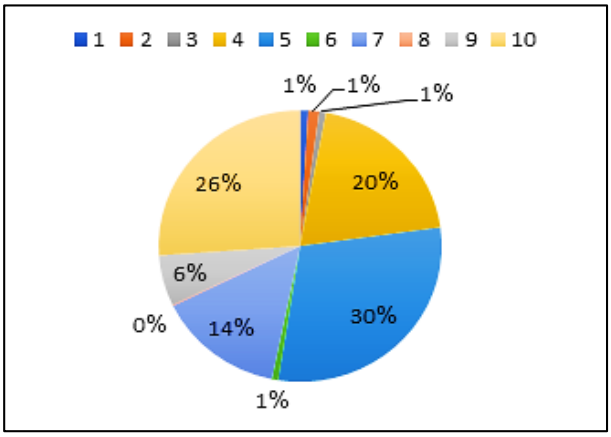

Figure 3. The selection process by the roulette wheel method for the function $f_1(x)$ of 10 primary chromosomes, where each sector represents the proportion corresponding to the selection process.

The MATLAB function for the method of selection process in the roulette wheel for N a certain number of chromosomes:

```
function roulette=RandChooseN(p,n)
binedges=[0,cumsum(p(:)')];
roulette=zeros(1,n);
for i=1:n
   x=rand;
   counts=histc(x,binedges);
   roulette(i)=find(counts==1);
end
end
```

The results were compared with the standard algorithms shown the results in Tables 4, 5, 6, and 7:

Table 4. The numerical results of the function $f_1(x)$

Algorithm	Iterations	$\|f(x_{n+1})\|$
NM	8	2.28E-25
WFM	6	2.80E-16
GOM	6	4.85E-25
KLWM	6	5.65E-13
WM	5	1.71E-33
Zalinescu	7	5.88E-50
Proposed algorithm	4	4.62E-75

Table 3, shows the comparison of the proposed algorithm with the standard algorithms in terms of the number of iterations, which amounted to 4 iterations, and the amount of error is very small for the standard algorithms, likewise for Tables 5,6,7.

Table 5. The numerical results of the function $f_2(x)$

Algorithm	Iterations	$\|f(x_{n+1})\|$
NM	11	1.08E-4
WFM	7	1.76E-4
GOM	7	4.66E-7
KLWM	7	2.44E-10
WM	7	6.22E-6
Zalinescu	9	1.19E-10
Proposed algorithm	6	1.55E-14

Table 6. The numerical results of the function $f_3(x)$

Algorithm	Iterations	$\|f(x_{n+1})\|$
NM	11	1.58E-4
WFM	7	1.86E-4
GOM	7	2.47E-6
KLWM	7	2.74E-7
WM	7	1.53E-5
Zalinescu	7	2.95E-9
Proposed algorithm	5	1.69E-11

Table 7. The numerical results of the function $f_4(x)$

Algorithm	Iterations	$\|f(x_{n+1})\|$
NM	7	7.03E-68
WFM	4	1.22E-116
GOM	5	4.74E-80
KLWM	5	3.39E-53
WM	5	3.36E-86
Zalinescu	6	2.00E-169
Proposed algorithm	4	1.78E-125

Discussion:

From the results, concluding that:

- Through the results, the (NM) method takes more iterations with a rather large amount of error compared to other comparison algorithms.
- There is a convergence between (WFM) and (GOM) methods in terms of the number of iterations and the value of $|f(x_{n+1})|$,while the methods(KLWM), (WM) and (Zalinescu method) the efficiency index is close.
- The proposed relationship to the learning rate that controls the behavior of the proposed method reduces the number of iterations and the amount of error because it does not take a fixed value during the iteration process, rather, it increases and

Published Online First: August, 2024
https://doi.org/10.21123/bsj.2023.7481
P-ISSN: 2078-8665 - E-ISSN: 2411-7986
Baghdad Science Journal

takes a new value that improves the required value.
- The genetic algorithm helps in finding the optimal and appropriate initial solution to start the proposed method away from randomness in taking the initial root.
- It is shown that this new method is more efficient than these existing methods and this method has lowest number of iteration and converges faster than the other methods.

Conclusion

Genetic algorithms and the SGD algorithm are among the most important algorithms used in artificial intelligence applications in optimization applications. Therefore, they were proposed to improve some numerical analysis algorithms in finding the roots of some functions by modifying the frequency relationship of these methods, as was done in this work.

Authors' Declaration

- Conflicts of Interest: None.
- We hereby confirm that all the Figures and Tables in the manuscript are ours. Furthermore, any Figures and images, that are not ours, have been included with the necessary permission for re-publication, which is attached to the manuscript.
- Ethical Clearance: The project was approved by the local ethical committee in University of Aleppo, Aleppo, Syria.

Authors' Contribution Statement

A.Y.A. made the conception, design, interpretation and results discussion. M.I. worked on acquisition of data and results analysis & discussion. N.A.I. worked on revision and proofreading and results discussion.

References

1. Ricceri B A. class of equtions with three solutions. Mathematics. 2020;8(478): 1-8. https://doi.org/10.3390/math8040478.
2. Lu C, Shi J. Relative density prediction of additively manufactured Inconel 718: a study on genetic algorithm optimized neural network models. Rapid Prototyp J. 2022; 28(8): 1425-1436. https://doi.org/10.1007/s00170-021-08388-2 .
3. Ide N. A New Modified of McDougall-Wotherspoon Method for Solving Nonlinear Equations by Using Geometric Mean Concept. Comput. Appl Math Sci. 2019; 4(2): 35-38.
4. Rafiq N, Shams M, Ahmad B. Inverse Numerical Iterative Technique for Finding all Roots of Nonlinear Equations with Engineering Applications. J Math. 2021; (2): 1-10. https://doi.org/10.1155/2021/6643514.
5. Salman NK, Mustafa MM. Numerical Solution of Fractional Volterra-Fredholm Integro-Differential and Equation Using Lagrange Polynomials. Baghdad Sci J. 2020; 17(4): 1234-1240. http://dx.doi.org/10.21123/bsj.2020.17.4.1234
6. Qin X, Liu T, Li Q. An optimal fourth-order family of modified Cauchy methods for finding solutions of nonlinear equations and their dynamical behavior. Open Math. 2019; 17(1): 1567-1598. https://doi.org/10.1515/math-2019-0122.
7. Faez H, Riyam N. Using Evolving Algorithms to Cryptanalysis Nonlinear Cryptosystems. Baghdad Sci J. 2020; 17(2): 682-688. http://dx.doi.org/10.21123/bsj.2020.17.2(SI).0682
8. Holland JH. Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence. 2nd edition. USA: MIT Press; 1992. p. 211. https://ieeexplore.ieee.org/servlet/opac?bknumber=6267401.
9. Xie Z, Sato I, Sugiyama M. A Diffusion Theory for Deep Learning Dynamics: Stochastic Gradient Descent Escapes from Sharp Minima Exponentially Fast. arXiv preprint arXiv:2002.03495. 2020 Feb 10; 1-28. https://openreview.net/pdf?id=wXgk_iCiYGo.
10. Leclerc G, Madry A. The Two Regimes of Deep Network Training. arXiv preprint arXiv:2002.10376. 2020 Feb 24; 1-14. https://arxiv.org/pdf/2002.10376.
11. Frontini M, Sormani E. Some variant of Newton's method with third-order convergence. Appl Math Comput. 2003; 140: 419-426. https://doi.org/10.1016/S0096-3003(02)00238-2

Published Online First: August, 2024
https://doi.org/10.21123/bsj.2023.7481
P-ISSN: 2078-8665 - E-ISSN: 2411-7986
Baghdad Science Journal

12. Glisovic N, Ralevic NM, Cebic D. A variant of McDougall- Wotherspoon method for finding simple roots of nonlinear equations. Scientific Publications of the State University of Novi Pazar Ser A. Appl Math Inform Mech. 2018; 10(1): 55-61. https://doi.org/10.5937/SPSUNP1801055G
13. Li Y, Kou J, Wang X. A modification of Newton method with third-order convergence. Appl Math Comput. 2006; 181: 1106–1111. https://doi.org/10.1016/j.amc.2006.01.076.
14. Zalinescu C. On Berinde's Method for Comparing Iterative Processes. Fixed Point Theory Algorithm Sci Eng. 2019; 10 https://doi.org/10.48550/arXiv.1812.00958

خوارزمية جديدة لإيجاد جذور المعادلات الجبرية غير الخطية

احمد يوسف الرزو[1] ، نصر الدين عيد[1] ، محمد اسعد[2]

[1]قسم الرياضيات، كلية العلوم، جامعة حلب، حلب، سوريا.
[2]قسم الرياضيات، كلية العلوم، جامعة تشرين، اللاذقية، سوريا.

الخلاصة

تم في البحث تهجين خوارزمية SGD (Stochastic Gradient Descent) التي تعتبر من أشهر خوارزميات الأمثلية مع الخوارزميات الجينية في إيجاد جذور معادلات غير خطية التي تعتبر من اهم المسائل الرياضية نظرا لتطبيقها في جميع العلوم، حيث تم استخدام الخوارزميات الجينية في إيجاد الجذر الابتدائي الأمثل لخوارزمية SGD وتطبيقها في تقليل دالة الهدف المدروسة، حيث ان بعض الخوارزميات الشهيرة تحتاج الى نقطة ابتدائية للوصول الى الحل من حيث الاستقرار. تم اختبار الخوارزمية المقترحة على عدة دوال قياسية ومقارنة النتائج مع الخوارزميات الشهيرة وتبين النتائج كفاءة الخوارزمية المقترحة من خلال الجداول والأشكال.

الكلمات المفتاحية: الخوارزميات الجينية، المعادلات غير الخطية، دالة الهدف، الأمثليات، الانحدار التدريجي العشوائي.