A Novel Method for Solving Fixed Points of Various Functions by Bisection-Multi-Verse Optimization Algorithm

Sayyed Masood Zekavatmand #1, Javad Vahidi *2, Mohammad Begher Ghaemi #3
Department of Applied Mathematics, Iran University of Science and Technology, Tehran, Iran
S_zekavatmand@mathdep.iust.ac.ir

Abstract
The necessary goal is to obtain the fixed point of variant functions by using a novel repetitive method. We merge implications offered in the Bisection method and the Multi-Verse Optimizer algorithm. This algorithm is more effective for finding fixed point functions. We also implement this method for four functions and compare the current method with other methods such as WOA, SSA, ALO, MVO algorithms. The proposed method shows a decent functionality than the other methods.
Keywords: Fixed-point theory, Multi-Verse Optimization algorithm, Bisection procedure.

Tob Regul Sci. ™ 2022;8(1): 1692-1707
DOI: doi.org/10.18001/TRS.8.1.131

Introduction

Many real-world problems can be modeled as mathematical problems. Many of these problems come in the form of one or more nonlinear equations that solve a major problem in nature. Therefore, solving equations and gaining roots of functions, is of special importance in engineering, basic sciences and applied sciences. This reason has led many researchers in recent years to research to solve these problems and present the results of their efforts in the form of various articles [2,18,19,21,32,35,36,37,40,41,43].

The bisection algorithm is a rooting-discovering procedure. It handles for each function that is continuous for per identifies two valences by contrary emblems. The procedure includes frequently halving distance determined with these valences. Afterwards choosing the sub-distance wherein the function shifts emblem. It should include a root. It is a straightforward and sturdy procedure; however, it is as well as comparatively lagging. For this reason, it is frequently utilization to gain a precise estimate of one solution that is henceforth utilization as a beginning point to rather quickly converging procedures [4]. A meta heuristic optimization algorithm is an innovative method that can be applied to various optimization problems with little modification. Meta heuristic algorithms significantly increase the ability to find high-quality solutions to difficult optimization problems [1,11,17,22,23,24,25,26,28,31,34].

A common feature of these algorithms is the use of local optimization exit mechanisms. Meta heuristic algorithms are divided into two general groups, two methods based on one answer and population. One-answer algorithms change an answer during the search process, while population-based algorithms consider a population of answers while searching.

One-answer algorithms focus on local search areas; In contrast, population-based algorithms can search simultaneously in different areas of the answer space [1,5,6,10,12,15,17,27,33,38,42,43]. Various criteria can be used to classify meta-heuristic algorithms. Some of the one-answer meta-heuristic algorithms that have been introduced in recent years and have many applications for solving various problems are: simulated annealing algorithm (SA) [44], tabu search algorithm (TSA) [45], algorithm GRASP search (GSA) [43], variable neighborhood search (VNS) algorithm [47], guided local search algorithm (GLS) [48], iterated local search algorithm (ILS) [11]. Also, a number of population-based meta-innovative algorithms that have been introduced in recent years and have many applications for solving various problems are: Imperialist Competitive Algorithm (ICA) [49], Artificial Immune System Algorithm (AIS) [50], Harmony Search Algorithm (HS) [9]. In recent years, researchers have also been able to solve complex problems in the field of engineering and mathematics using a number of these algorithms. Some of these algorithms are: Cuckoo Optimization Algorithm (COA) [51], Gravitational Search Algorithm (GSA) [52], Invasive Weed Optimization Algorithm (IWO) [53], League Championship Algorithm (LCA) \cite{54}, Optics Inspired Optimization Algorithm (OIO) [55], Shuffled Frog Leaping Algorithm (SFL) [56], Stochastic Fractal Search Algorithm (SFS) [57] Teaching Learning Based Optimization Algorithm (TLBO) [58], Wind Driven Optimization Algorithm (WD) [59], Harris Hawks Optimization Algorithm (HHO) [60], Bat Algorithm (BA) [61], Crow search Algorithm (CSA) \cite{62}, Chimp optimization Algorithm (COA) [63], Group teaching Optimization Algorithm (GTO)) [64], symbiotic organisms search Algorithm (SOS) [65], Farmland fertility Algorithm (FFA) [66] and etc [3,7,9,13,14,16,20,29,30,39].

In the present article, we bring a novel repetitious method that merges the benefits of both the Multi-Verse Optimizer Algorithm and the Bisection method to dissolve the difficult fixed point problem. The residual segments of the present paper are organized as follows: at Sect. 2, a short overview is performed on the Bisection procedure. Also the Multi-Verse Optimization mechanism is said. Then the Fixed-point theory is explained. In the following , offered mechanism communicate in the next section . at Sect. 4, evaluates precision of suggested procedure with distinct procedures on various functions. Finally, the result is given in the final section.

Basic implications

By reading this segment, a concise overview is accomplished on the Multi-Verse Optimizer Algorithm, and the Bisection method is clarified. In the continuation of this segment fixed point theory is indicated.

Multi-Verse Optimization mechanism

Multi-Verse Optimization algorithm, which is called MVO for short, is a meta-heuristic and population-based algorithm. This algorithm was published by Seyedali Mirjalili in 2015 in The

Natural Computing Applications magazine of Springer Publications. Multi-order theory is based on multi-order theory, which believes that the universe was built on the basis of several large explosions. This theory suggests the existence of several parallel worlds that continue in parallel [23]. The MVO algorithm is based on three main contents called white hole, black hole and wormhole.

The MVO meta-heuristic algorithm is one of the leading algorithms in solving optimization problems. This algorithm can be used by using special measures in discrete, continuous and binary conditions and obtain appropriate results from it. Inspired by three concepts in the multi-world theory for the production of the MVO meta heuristic algorithm, which are described below.

White hole: The white hole is not naturally visible from the origin of our world. Scientists in the field of astronomy believe that the basis of the Big Bang was a white hole. One of the main factors in the creation of the universe is the white hole.

Black hole: Black hole is a phenomenon that is affected by the two components of time and space and has extraordinary gravitational power that all the particles of the universe are inevitably affected by its power and no particle, whether light or electromagnetic particles. They do not have the ability to escape their gravitational field. Also, black holes have fundamental differences from white holes in terms of behavior, action and frequency.

Wormhole: It breaks the structure and dimension of space and time and creates tunnels and holes in which the speed of material will be faster than the speed of light. Wormholes also split and retract the dimension and structure of space, shortening the distance between two points in space. The interaction of these three concepts has created a stable situation, which is what inspires the algorithm. The performance of population-based search algorithms is divided into two parts: exploration and extraction.

The two concepts of black hole and white hole have been used for exploration in the search space, and in contrast, the concept of wormhole has been used to extract. Each answer is considered similar to a world, and each variable is the answer to an object in the world. Each response is assigned an inflation rate that is proportional to the value of the objective function for that response. The time variable is also used as the cycle of the algorithm. The following rules apply to the world:

The highest inflation rate indicates the highest probability of a white hole.

The highest inflation rate indicates the lowest probability of a black hole.

The world with the highest inflation rate is more inclined to send objects from the white hole.

The world with the lowest inflation rate is more inclined to receive objects from the black hole.

Objects in all worlds may randomly move to the best world through wormholes regardless of inflation.

When a black (white hole) tunnel is established between two worlds, the world with the highest inflation rate plays the role of the white hole, and the world with the lowest inflation rate plays the role of the black hole, and objects move in this pattern. To select the origin and destination according to their quality, the roulette wheel method, which is one of the most suitable methods of random selection, has been used [23].

Mechanism of fixed point theory

Point (b) is a fixed point for the function g(y), when point (b) holds for g(b) = b. This means that the point represented by a function on itself can be considered a fixed point for that function. To find the fixed point of a function, we must convert the relation y = g(y) to the relation f(y) = 0. Then using the recessive connection for $y_{i+1} = g(y_i), i = 0,1,2,\ldots$ and using the primary value y_0 which is chosen based on conjecture and coincidence, we obtain the fixed point of the function. For further understanding, the mechanism of the fixed point theory is shown in the following figure, and to summarize the iterations of the fixed point theory in (FPI).

Determined an relation $f(y) = 0$
Transform $f(y) = 0$ at the shape $g(y) = y$
Imaging the elementary conjecture exist y_0
Do
$y_{i+1} = g(y_i)$
While (nothing of the convergence standard $E1$ or $E2$ is available)

Figure 1 (FPI) Mechanism

According to Fig 1, an repetitious procedure that can be used to find the solutions to equation f(y) = y is the recessive connection $f(y_i) = y_{i+1}, i = 0,1,2,\ldots$ by several premier supposable y_0.The mechanism pauses until per of the tracking pausing scale is available: E1.

Confirmation former the whole number of repetitions N.E2.With examining the position $|g(x_i - x_{i+1})|$ (wherever i is the repetition number) lower than several toleration range, tell epsilon, fixed former. (Figure Fig 1).

For instance, function $f_1 = x^2 - 15\cos(2px) + 15, x Î [-5,+5]$ has a fixed point, But a series of functions do not have a fixed point. For instance, Function f(x) = 4x + 6, $x Î$ ¡ has no fixed points, Because no number can be found that establishes the equation x = 4x + 6 [43].

Definition of the Bisection method

One of its methods is the bisection method whenever we want to provide a numerical solution for the equation f(x) = 0. Note that the function f must be a continuous function, and the values f(a) and f(b) on the interval [a, b] must have opposite signs. In this case, we can use the theorem of the mean to say that the function f in the interval (a, b) has at least one root [43].

In the bisection method, in each step, using the relation $c = \frac{(a+b)}{2}$, we divide the interval into two parts. Then we calculate the value of f(c). In this case, two situations may occur:

Case 1: If the values f(a) and f(c) have different symbols, then using the mean value theorem, the function f in the interval (a, c) has at least one root.

Case 2: If the values f(c) and f(b) have different symbols, then using the mean value theorem, the function f in the interval (c, b) has at least one root [5].

These steps continue until the interval is small enough, and finally, we reach a small interval where the two ends of the interval are the same size.

Note that the values at each end of the interval must have different symbols so that we can select it as a new, smaller interval [43].

In Algorithm 1, you can see the pseudo-code of the bisection method.

Algorithm 1 Algorithm of the bisection method

```
Input: A continuous function f(x) , interval [a, b], Tolerance: TOL; maximum number of iterations: $n_0$
  if $\frac{(b-a)}{2} < TOL$
    exit
    Set i=1
    while $i < n_0$ and $\frac{(b-a)}{2} < TOL$ do
      Set $c = a + \frac{(b-a)}{2}$ Compute f(x)
      if f(c) =0
         output c and exit
      end if
      if f(a) f(c) > 0
        set a=c and f(a) = f(c)
    else
      set b=c and f(b) = f(c)
     end uf
    i=i+1
  end while
   if $\frac{(b-a)}{2} < TOL$ then
     output c, 'tolerance limit exceeded!'
   end if
   if $i > N_0$
     output c, 'number of iterations reached $N_0$'
end if
end if
```

Mechanism of Bisection-Multi-Verse Optimization Algorithm

The main purpose of studying this section is to fully explain the mechanism of our new method, which is a combination of the MVO algorithm and the bisection method. We show this algorithm briefly as BMVO. The introduced algorithm provides the ability to obtain fixed points of different functions with very close approximations to the exact answer. For this purpose, consider the function g (x) = 0. We know that to get the fixed points of this function, it must be g (x) = x. Now consider the function f(x) as f(x) = g(x) -x. We can obtain the answers of the function f(x) to obtain the fixed points of the function g(x). Now consider the function h(x) as h(x) = |f (x)|. In simpler terms, we can say that the minimum solutions of the function $h(x)$ are the same as the fixed point of the function g(x). So instead of solving the function g(x) = x, we get the minimum answer of the function h(x). In this step, using the mvo algorithm, consider an answer such as x_k that is randomly selected in the range $I_k = [a_k, b_k]$. In fact, the selection of the initial answer x_k is completely random and is done using the mvo algorithm. Then, at this stage, the bisection method is used. We calculate the values of $f(x_k)$ and $f(a_k)$ and see if the product of them is a negative number or not. We also do the same for $f(x_k)$ and $f(b_k)$ and see if the product of these two numerical values is negative. Note that the termination condition of the algorithm is that $f(x_k) = 0$.

The Algorithm of the Bisection-Multi-Verse Optimization Algorithm is determined in Algorithm 2.

In Algorithm 2 for per repetition we gain accidentally modern estimation amount x_k of the answer equation with MVO algorithm on I_k, for per $k \in ¥$ b_k (just $[a_k, \frac{x_k + a_k}{2}]$ is either , I_{k+1}. a_k (just $[\frac{b_k + x_k}{2}, b_k]$ shifts) or b_k and a_k (both $[x_k, \frac{x_k + b_k}{2}]$ shifts) or we have or shifts). So, we achieve $I_{k+1} \subseteq I_k$ Afterwards, c belongs to the subscription of all I_k.

Finally, the limit $c - x_k$ will be equal to c when k tends to infinity.

Algorithm 2 the pseudo-code of the Bisection-Multi-Verse Optimization Algorithm

Given fixed point problem $ g(x)=x, \,\, x \in [a, b] $
End points: $a_0 = a, b_0 = b$; tolerance: $TOL(1e^{-t}, t >> 0)$
Let f(x) = g(x) - x, h(x) = | f(x) |, k=0
while $h(x_k) \geq TOL$ do
  $x_k \in [a_k, b_k]$ is generated by MVO algorithm that minimize $h(x)$
if $f(x_k) f(a_k) < 0$
  $b_{k+1} = \frac{x_k + b_k}{2}, temp = \frac{x_k + a_k}{2}$
 if $f(temp) f(x_k) < 0$
  $a_{k+1} = temp$
 else
  $a_{k+1} = a_k$

end if
else
$a_{k+1} = \frac{x_k + a_k}{2}, temp = \frac{x_k + b_k}{2}$
if $f(temp)f(x_k) < 0$
$b_{k+1} = temp$
else
$b_{k+1} = b_k$
end if
end if
k=k+1 $
end while
Output: $c = x_k$

Using the introduced algorithm for several functions

At present part , clarify proposed method by several examples and compare the outcomes with other evolutionary optimization methods similar WOA, SSA, ALO and as well as MVO. Consider the following functions.

$g_1(x): x^2 - 15\cos(2\pi x) + 15 = x, x \in [-5,5]$

$g_2(x): e + 20 - 20e^{-0.2\sqrt{x^2}} - e^{\cos(2\pi x)} = x, x \in [1,21]$

$g_3(x): \sin x - \frac{x^3}{8} = x, x \in [-10,+10]$

$g_4(x): 2x^3 - 4\cos x + 4 = x, x \in [-25,25]$

In Table 1, these functions are categorized and introduced.

Table 1Category of functions introduced

Function	Domain
$f_1(x) \to x^2 - 15\cos(2\pi x) + 15 = 0$	$[-5,+5]$
$f_2(x) \to e + 20 - 20e^{-0.2\sqrt{x^2}} - e^{\cos(2\pi x)} = 0$	$[1,21]$
$f_3(x) \to \sin x - \frac{x^3}{8} = 0$	$[-10,+10]$
$f_4(x) \to 2x^3 - 4\cos x + 4 = 0$	$[-25,+25]$

According to the definitions mentioned in befor section, for the function g_1 the fixed point is the number 0. Also, fixed point of g_2 function is a number very close to 20 and the fixed point of the g_3 function is 0, and finally the fixed point of the function g_4 is also 0.

Here we want to provide a complete description of solving the function $f_1(x) = 0$ using the mentioned algorithm. first we produce a accidental initial value for the function f_1 at $x_0 \in [-5,5]$ using the MVO algorithm. In the next step, using the method mentioned in Section 3, we gain modern approximate value x_1 Î I_1 Í I_0 as a solution. Then we continue this work until the desired solution is obtained with the favorable accuracy. The outcomes acquired toward any function via WOA, SSA, ALO, MVO and BMVO mechanisms are presented in Table 2. Shapes 2 to 5 demonstrate scheme of improvement procedure of g_1 to g_4 functions via BMVO mechanism in (a) and shape of solving fixed point of g_1 to g_4 functions via BMVO mechanism in (b). Shapes 6 to9 demonstrate scheme of the outcome of WOA, SSA, ALO, MVO algorithms and suggested algorithm to dissolve the fixed point problem of each function in one graph. The results and figures show that the BMVO algorithm, compared to the other four algorithms, has a better performance and a more accurate solution for finding fixed points of functions.

Table 2 The outcomes acquired toward any function via WOA, SSA, ALO, MVO and BMVO algorithms

algorithm	Components	$g_1(x)$	$g_2(x)$	$g_3(x)$	$g_4(x)$
WOA	error	6.02E-15	2.96E-11	0	8.12E-14
	X-best	0.003366	20.82583	-3.27E-09	0.36841714
	mean(e)	2.9E-7	0.000101	3.99E-10	2.70E-05
	std(e)	8.39E-6	0.002235	1.26E-08	0.00084628
SSA	error	1.84E-11	8.67E-09	0	1.93E-09
	X-best	0.003366	19.92297	-1.54E-08	-1.26754337
	mean(e)	1.95E-6	0.004717	1.51E-10	4.99E-05
	std(e)	2.52E-5	0.00807	1.92E-09	0.00110784
ALO	error	7.03E-10	1.22E-09	0	1.05E-08
	X-best	0.003366	2.08E+01	-1.50E-08	-1.26754336
	mean(e)	1.45E-5	8.57E-05	7.91E-11	1.42E-04
	std(e)	3.8E-5	1.44E-03	1.24E-09	4.04E-04
MWO	error	4.55E-8	1.70E-05	1.11E-19	2.19E-07
	X-best	0.996634	2.01E+01	7.25E-07	3.68E-01
	mean(e)	9.13E-6	1.40E-03	3.11E-12	1.81E-04
	std(e)	3.8E-5	3.59E-03	1.61E-11	3.22E-03
BMVO	error	2.54E-79	1.07E-14	0	4.44E-16
	X-best	2.54E-79	20.09815	-1.14E-09	-1.26754337
	mean(e)	9.67E-9	2.78E-05	4.09E-18	2.52E-05
	std(e)	1.17E-7	0.000189	4.85E-18	0.00021581

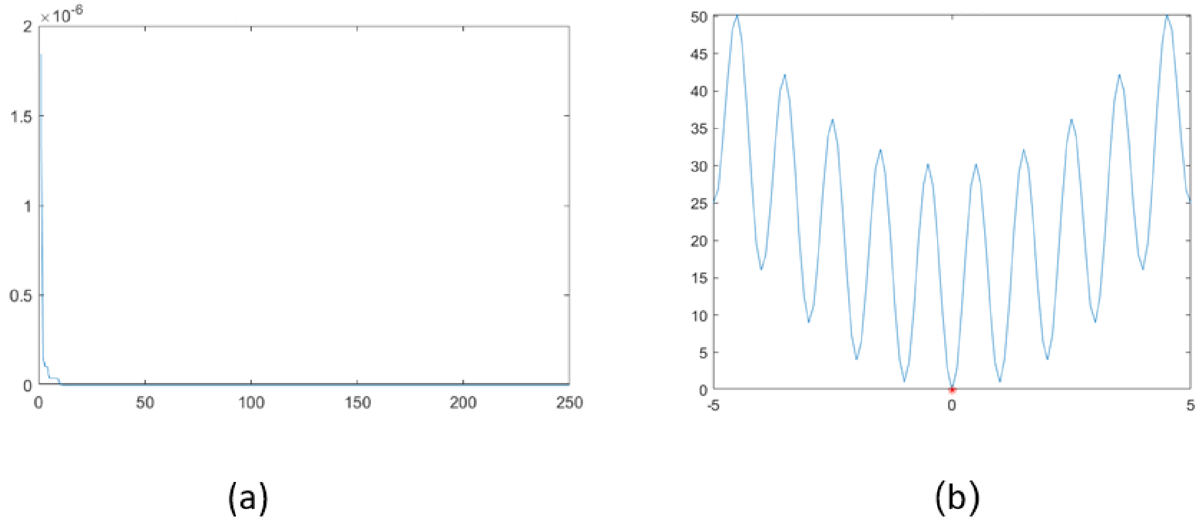


Figure 2 Shape of improvement procedure of $g_1(x)$ function via BMVO algorithm at form (a) and shape of solving fixed point of the $g_1(x)$ function via BMVO algorithm at form (b)

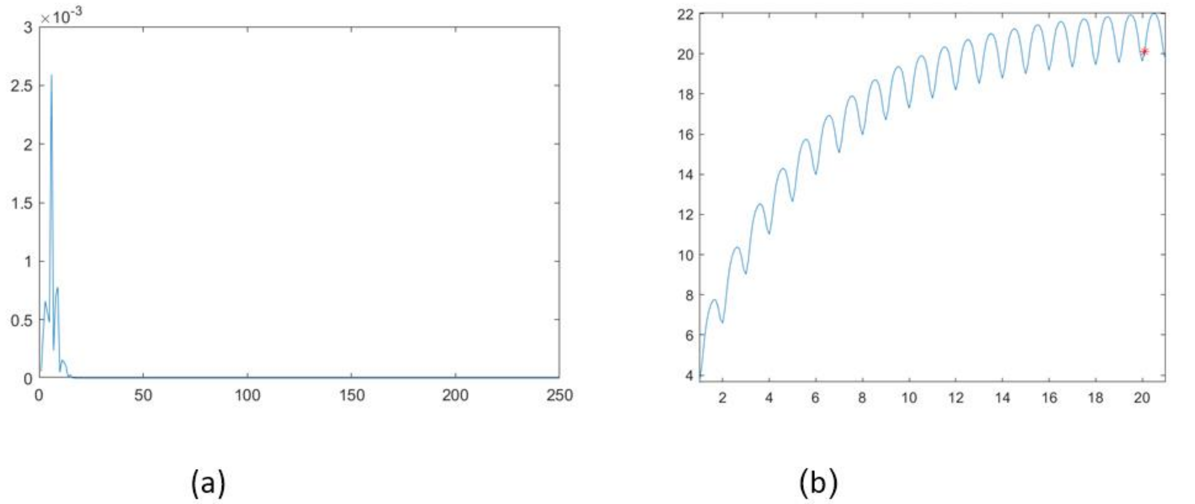
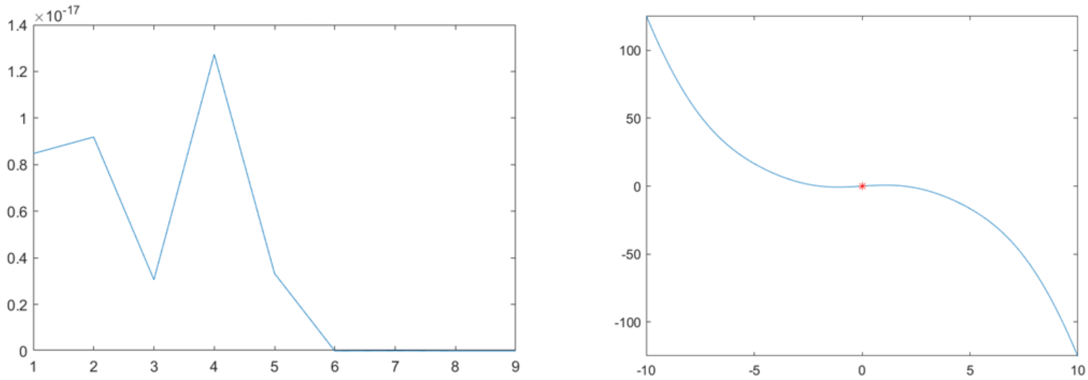
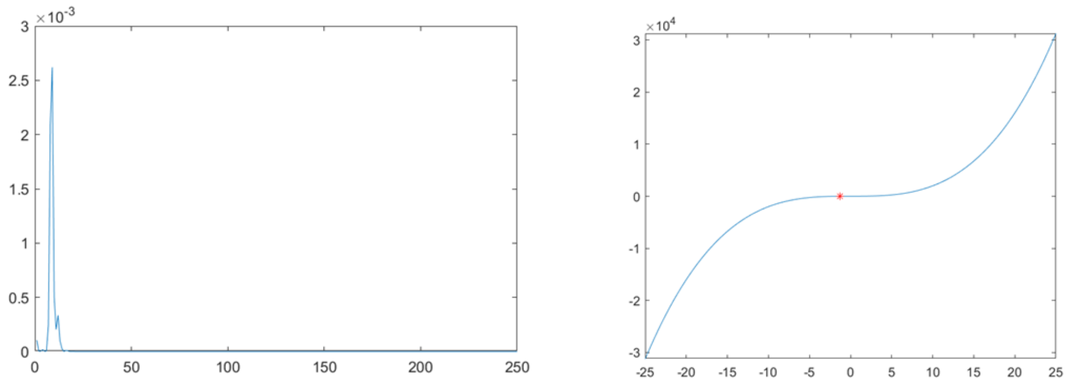


Figure 3 Shape of improvement procedure of $g_2(x)$ function via BMVO algorithm at form (a) and shape of solving fixed point of the $g_2(x)$ function via BMVO algorithm at form (b)



(a) (b)

Figure 4 Shape of improvement procedure of $g_3(x)$ function via BMVO algorithm at form (a) and shape of solving fixed point of the $g_3(x)$ function via BMVO algorithm at form (b)



(a) (b)

Figure 5 Shape of improvement procedure of $g_4(x)$ function via BMVO algorithm at form (a) and shape of solving fixed point of the $g_4(x)$ function via BMVO algorithm at form (b)

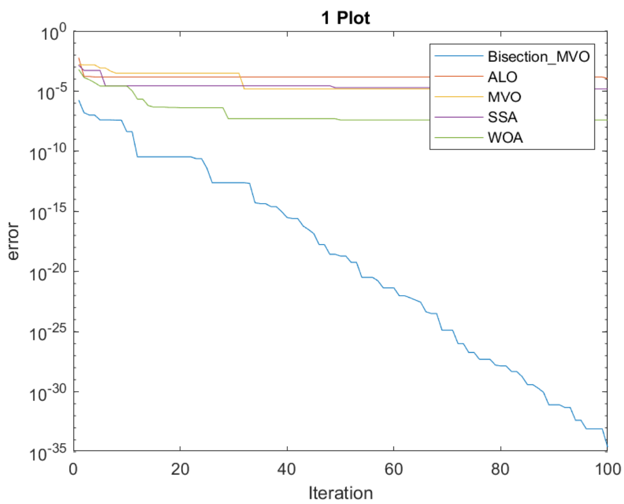


Figure 6 Shape of WOA, SSA, ALO, MVO algorithms and suggested algorithm to dissolve the fixed point of function $g_1(x)$

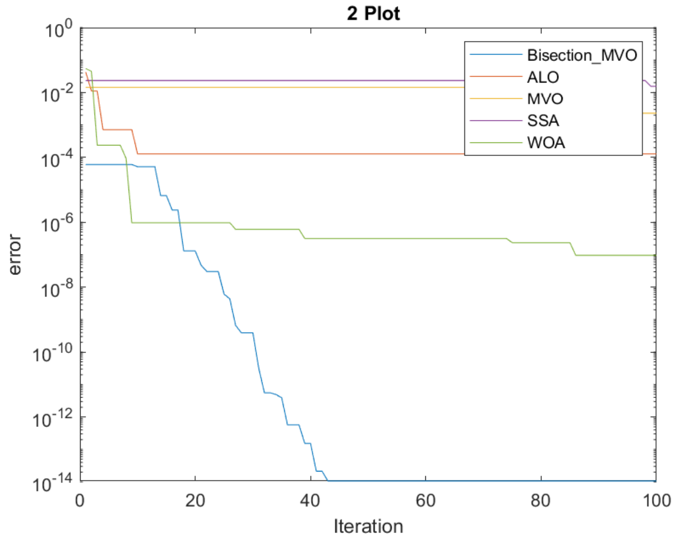


Figure 7 Shape of WOA, SSA, ALO, MVO algorithms and suggested algorithm to dissolve the fixed point of function $g_2(x)$

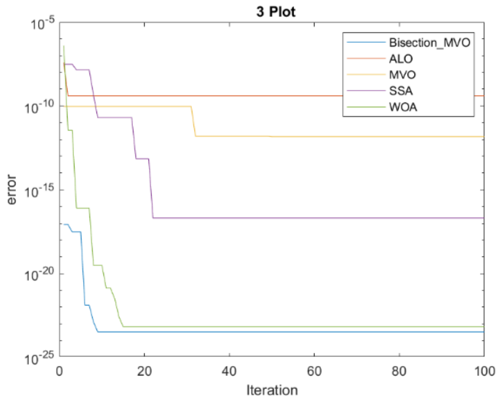


Figure 8 Shape of WOA, SSA, ALO, MVO algorithms and suggested algorithm to dissolve the fixed point of function $g_3(x)$

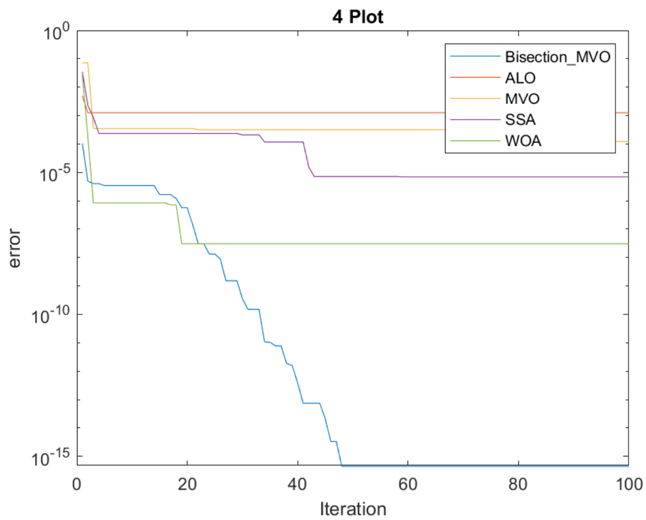


Figure 9 Shape of WOA, SSA, ALO, MVO algorithms and suggested algorithm to dissolve the fixed point of function $g_4(x)$

Conclusion

The present article's main endeavor is to discover a modern repetitive procedure that discovery fixed point of problems by utilizing the MVO algorithm and Bisection method. Finding good initial value in a proportional interval where the fixed point of the function is located can sometimes be difficult for complicated functions. we want to use the derivative method, some functions either. In that case, they do not have a derivative or are difficult to calculate. Then obtaining solutions of their derivative is time-consuming. So finding derivative of the function $f(y) = g(y)-y$ and then finding their solutions for all functions Not recommended. MVO algorithm aids in detecting an acceptable primary solution. In the following , suggested procedure does nowhere near via the want to calculate the derivative. This suggested method is ordinary to utilize and trustworthy. While a collation via alternative methods represents, precision of this suggested algorithm is acceptable.

Reference

[1] Abdelaziz, A. Y., Ali, E. S., & Abd Elazim, S. M. (2016). Flower pollination algorithm and loss sensitivity factors for optimal sizing and placement of capacitors in radial distribution systems. International Journal of Electrical Power & Energy Systems, 78, 207-214.

[2] Alizadegan, A., Asady, B., & Ahmadpour, M. (2013). Two modified versions of artificial bee colony algorithm. Applied Mathematics and Computation, 225, 601-609.

[3] Alderremy, A. A., Attia, R. A., Alzaidi, J. F., Lu, D., & Khater, M. (2019). Analytical and semi-analytical wave solutions for longitudinal wave equation via modified auxiliary equation method and Adomian decomposition method. Thermal Science, 23(Suppl. 6), 1943-1957.

[4] Atashpaz-Gargari, E., & Lucas, C. (2007, September). Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In 2007 IEEE congress on evolutionary computation (pp. 4661-4667). Ieee.

[5] Bayraktar, Z., Komurcu, M., & Werner, D. H. (2010, July). Wind Driven Optimization (WDO): A novel nature-inspired optimization algorithm and its application to electromagnetics. In 2010 IEEE antennas and propagation society international symposium (pp. 1-4). IEEE.

[6] Burden, R. L., & Douglas, J. (1985). Faires. Numerical analysis.

[7] Colorni, A., Dorigo, M., & Maniezzo, V. (1991, December). Distributed optimization by ant colonies. In Proceedings of the first European conference on artificial life (Vol. 142, pp. 134-142).

[8] Cuevas, E., Cienfuegos, M., Zald\'{i}var, D., & Perez-Cisneros, M. (2013). A swarm optimization algorithm inspired in the behavior of the social-spider. Expert Systems with Applications, 40(16), 6374-6384.

[9] David Edward, G. (2002). Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesle.

[10] De Castro, L. N., Castro, L. N., & Timmis, J. (2002). Artificial immune systems: a new computational intelligence approach. Springer Science & Business Media.

[11] Deb, K. (2012). Optimization for engineering design: Algorithms and examples. PHI Learning Pvt. Ltd.

[12] Djerioui, A., Houari, A., Machmoum, M., & Ghanes, M. (2020). Grey Wolf Optimizer-Based Predictive Torque Control for Electric Buses Applications. Energies, 13(19), 5013.

[13] Dorigo, M., Birattari, M., & Stutzle, T. (2006). Ant colony optimization. IEEE computational intelligence magazine, 1(4), 28-39.

[14] Eusuff, M., Lansey, K., & Pasha, F. (2006). Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization. Engineering optimization, 38(2), 129-154.

[15] Ezugwu, A. E., Adeleke, O. J., & Viriri, S. (2018). Symbiotic organisms search algorithm for the unrelated parallel machines scheduling with sequence-dependent setup times. PloS one, 13(7), e0200030.

[16] Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. Computers & operations research, 13(5), 533-549.

[17] Harifi, S., Khalilian, M., Mohammadzadeh, J., & Ebrahimnejad, S. (2019). Emperor Penguins Colony: a new metaheuristic algorithm for optimization. Evolutionary Intelligence, 12(2), 211-226.

[18] Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., & Chen, H. (2019). Harris hawks optimization: Algorithm and applications. Future generation computer systems, 97, 849-872.

[19] Holland, J. H. (1991). Adaptation in natural and artifficial systems, 1975.

[20] Huang, J., & Ma, Y. (2020). Bat algorithm based on an integration strategy and Gaussian distribution. Mathematical Problems in Engineering, 2020.

[21] Hussien, A. G., Amin, M., Wang, M., Liang, G., Alsanad, A., Gumaei, A., & Chen, H. (2020). Crow search algorithm: theory, recent advances, and applications. IEEE Access, 8, 173548-173565.

[22] Homaa, W., Eddaly, M., & Jarboui, B. (2021). Variable neighborhood search algorithms for the permutation flowshop scheduling problem with the preventive maintenance. Operational Research, 21(4), 2525-2542.

[23] Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization (Vol. 200, pp. 1-10). Technical report-tr06, Erciyes university, engineering faculty, computer engineering department.

[24] Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. Journal of global optimization, 39(3), 459-471.

[25] Kashan, A. H. (2014). League Championship Algorithm (LCA): An algorithm for global optimization inspired by sport championships. Applied Soft Computing, 16, 171-200.

[26] Kennedy, J., & Eberhart, R. (1995, November). Particle swarm optimization. In Proceedings of ICNN'95-international conference on neural networks (Vol. 4, pp. 1942-1948). IEEE.

[27] Kennedy, J., Eberhart, R. C., & Shi, Y. (2001). Swarm Intelligence: Morgan Kaufmanns Academic Press. San Francisco.

[28] Kenneth, S. R., & Storn, R. M. (1997). Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization, 11(4), 341-359.

[29] Khater, M., Attia, R. A., & Lu, D. (2019). Modified auxiliary equation method versus three nonlinear fractional biological models in present explicit wave solutions. Mathematical and Computational Applications, 24(1), 1.

[30] Khater, M. M., Park, C., Lu, D., & Attia, R. A. (2020). Analytical, semi-analytical, and numerical solutions for the CahnAllen equation. Advances in Difference Equations, 2020(1), 1-12.

[31] Khishe, M., & Mosavi, M. R. (2020). Chimp optimization algorithm. Expert systems with applications, 149, 113338.

[32] Kirkpatrick, S., Gelatt Jr, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. science, 220(4598), 671-680.

[33] Ma, H., & Simon, D. (2011). Blended biogeography-based optimization for constrained optimization. Engineering Applications of Artificial Intelligence, 24(3), 517-525.

[34] Maheri, A., Jalili, S., Hosseinzadeh, Y., Khani, R., & Miryahyavi, M. (2021). A comprehensive survey on cultural algorithms. Swarm and Evolutionary Computation, 62, 100846.

[35] Mansouri, P., Asady, B., & Gupta, N. (2015). The Bisection-artificial bee colony algorithm to solve fixed point problems. Applied Soft Computing, 26, 143-148.

[36] Merrikh-Bayat, F. (2015). The runner-root algorithm: a metaheuristic for solving unimodal and multimodal optimization problems inspired by runners and roots of plants in nature. Applied Soft Computing, 33, 292-303

[37] Mills, P., & Tsang, E. (2000). Guided local search for solving SAT and weighted MAX-SAT problems. Journal of Automated Reasoning, 24(1), 205-223.

[38] Mirjalili, S. (2016). Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. Neural computing and applications, 27(4), 1053-1073.

[39] Mirjalili, S. (2015). Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. Knowledge-based systems, 89, 228-249.

[40] Mirjalili, S. (2015). The ant lion optimizer. Advances in engineering software, 83, 80-98.

[41] Mirjalili, S. (2016). SCA: A Sine Cosine Algorithm for solving optimization problems. a School of Information and Communication Technology. Griffith University, Nathan Campus, Brisbane, QLD, 4111.

[42] Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. Advances in engineering software, 95, 51-67.

[43] Mirjalili, S., Mirjalili, S. M., & Hatamlou, A. (2016). Multi-verse optimizer: a nature-inspired algorithm for global optimization. Neural Computing and Applications, 27(2), 495-513.

[44] Misaghi, M., & Yaghoobi, M. (2019). Improved invasive weed optimization algorithm (IWO) based on chaos theory for optimal design of PID controller. Journal of Computational Design and Engineering, 6(3), 284-295.

[45] Ochoa, A., Margain, L., Hernandez, A., Ponce, J., De Luna, A., Hernandez, A., & Castillo, O. (2013, August). Bat Algorithm to improve a financial trust forest. In 2013 World Congress on Nature and Biologically Inspired Computing (pp. 58-62). IEEE.

[46] Olariu, S., & Zomaya, A. Y. (2005). Biology-derived algorithms in engineering optimization. In Handbook of bioinspired algorithms and applications (pp. 607-618). Chapman and Hall/CRC.

[47] Rahkar Farshi, T. (2021). Battle royale optimization algorithm. Neural Computing and Applications, 33(4), 1139-1157.

[48] Rajabioun, R. (2011). Cuckoo optimization algorithm. Applied soft computing, 11(8), 5508-5518.

[49] Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2009). GSA: a gravitational search algorithm. Information sciences, 179(13), 2232-2248.

[50] Rao, R. V., Savsani, V. J., & Vakharia, D. P. (2011). Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. Computer-aided design, 43(3), 303-315.

[51] Rezazadeh, H., Younis, M., Eslami, M., Bilal, M., & Younas, U. (2021). New exact traveling wave solutions to the (2+ 1)-dimensional Chiral nonlinear Schrodinger equation. Mathematical Modelling of Natural Phenomena, 16, 38.

[52] Salimi, H. (2015). Stochastic fractal search: a powerful metaheuristic algorithm. Knowledge-Based Systems, 75, 1-18.

[53] Saremi, S., Mirjalili, S., & Lewis, A. (2017). Grasshopper optimisation algorithm: theory and application. Advances in Engineering Software, 105, 30-47.

[54] Shah-Hosseini, H. (2009). The intelligent water drops algorithm: a nature-inspired swarm-based optimization algorithm. International Journal of Bio-inspired computation, 1(1-2), 71-79.

[55] Shayanfar, H., & Gharehchopogh, F. S. (2018). Farmland fertility: A new metaheuristic algorithm for solving continuous optimization problems. Applied Soft Computing, 71, 728-746.

[56] Vahidi, J., Zekavatmand, S. M., Rezazadeh, H., Inc, M., Akinlar, M. A., & Chu, Y. M. (2021). New solitary wave solutions to the coupled Maccaris system. Results in Physics, 21, 103801.

[57] Wazwaz, A. M. (2004). A sine-cosine method for handlingnonlinear wave equations. Mathematical and Computer modelling, 40(5-6), 499-508.

[58] Wazwaz, A. M. (2005). The tanh method and the sine-cosine method for solving the KP-MEW equation. International Journal of Computer Mathematics, 82(2), 235-246.

[59] Yang, X. S. (2009, October). Firefly algorithms for multimodal optimization. In International symposium on stochastic algorithms (pp. 169-178). Springer, Berlin, Heidelberg.

[60] Yang, X. S. (2010). Nature-inspired metaheuristic algorithms. Luniver press.

[61] Yao, S. W., Zekavatmand, S. M., Rezazadeh, H., Vahidi, J., Ghaemi, M. B., & Inc, M. (2021). The solitary wave solutions to the Klein-Gordon-Zakharov equations by extended rational methods. AIP Advances, 11(6), 065218.

[62] Yokus, A., Durur, H., & Ahmad, H. (2020). Hyperbolic type solutions for the couple Boiti-Leon-Pempinelli system. Facta Universitatis, Series: Mathematics and Informatics, 35(2), 523-531.

[63] Yonezawa, Y., & Kikuchi, T. (1996, October). Ecological algorithm for optimal ordering used by collective honey bee behavior. In MHS'96 Proceedings of the Seventh International Symposium on Micro Machine and Human Science (pp. 249-256). IEEE.

[64] Zekavatmand, S. M., Rezazadeh, H., Inc, M., Vahidi, J., & Ghaemi, M. B. (2021). The new soliton solutions for long and short-wave interaction system. Journal of Ocean Engineering and Science.

[65] Zhang, Y., & Jin, Z. (2020). Group teaching optimization algorithm: A novel metaheuristic method for solving global optimization problems. Expert Systems with Applications, 148, 113246.