ICIC Express Letters ICIC International ©2016 ISSN 1881-803X
Volume **10**, Number **1**, January **2016** pp. **103–109**

A HYBRID ALGORITHM BASED ON TEACHING-LEARNING-BASED OPTIMIZATION AND NEWTON METHOD FOR SOLVING NONLINEAR EQUATIONS

Xia Li[1,2], Peifeng Niu[1,*], Litong Zhao[2], Guoqiang Li[1] and Jianping Liu[2]

[1]Key Lab of Industrial Computer Control Engineering of Hebei Province
Yanshan University
No. 438, West Hebei Avenue, Qinhuangdao 066004, P. R. China
*Corresponding author: niupeifeng2014@163.com

[2]Department of Mathematics and Information Science
Hebei Normal University of Science and Technology
No. 360, West Hebei Avenue, Qinhuangdao 066004, P. R. China

Received June 2015; accepted September 2015

Abstract. *Solving nonlinear equations is an important problem in engineering field. Therefore, the study of efficient algorithm is of great significance. Teaching-Learning-Based Optimization (TLBO) has the advantages of simple algorithm, less parameters, strong global convergence ability and insensitivity to initial point, but it is weak at local optimization and slow later convergence. On the other hand, Newton method has the advantage of local deep search and fast convergence rate, but it is sensitive to the initial point. Therefore, considering advantages and disadvantages of two methods above, this paper proposes a hybrid algorithm (TNHA) based on the TLBO and Newton method for solving nonlinear equations. The hybrid algorithm both combines all these merits of the TLBO and Newton method but does not have the defects. Finally, numerical examples verify the efficiency of the TNHA.*

Keywords: Nonlinear equations, Teaching-Learning-Based Optimization, Newton method, Hybrid algorithm

1. **Introduction.** Nonlinear problem is the main subject in the modern mathematics research. In theory research and practical application, many practical engineering problems are converted into equations. Solving the system of nonlinear equations becomes a key problem. When the initial value given is close to the exact solution, the Newton method [1] has second-order convergence rate, and is a very effective local search algorithm. However, the method is very strict with the selection of the initial value. The convergence speed has great changes due to different initial value, and bad initial value even causes the divergence of the method.

On the other hand, solving equations problem can be attributed to the function optimization problem, so we can find the solution of equations by optimization technology. The key problem is to design effective algorithm. During the last three decades, more and more researchers are inspired by nature phenomenon, and they have proposed lots of heuristic optimization algorithms. There are some famous heuristic optimization algorithms, such as Genetic Algorithm (GA) [2,3], Particle Swarm Optimization (PSO) algorithm [4-6] and Gravitation Search Algorithm (GSA) [7,8]. However, a common problem on these optimization techniques is controlling parameters and a change in the parameters changes the effectiveness of the algorithms. In order to overcome the limitations, Teaching-Learning-Based Optimization (TLBO) algorithm, which is a parameter free algorithm, is proposed by Rao et al. [9-11]. The algorithm simulates teaching-learning phenomenon of a classroom to solve multi-dimensional, linear and nonlinear problems with appreciable efficiency, which has been applied to many engineering optimization problems and has been proved effective to solve some problems [12-15]. However, to our best knowledge, solving equations based on TLBO has not been found in literature.

TLBO algorithm has the advantages of simple algorithm, less parameters, strong global convergence ability, insensitivity to initial point and implicit parallelism. Therefore, TLBO is suitable for solving a large size of equations. However, there are some shortcomings such as slow later convergence and poor local ability. On the other hand, Newton method has the high local convergence speed. Therefore, we can take advantage of the two methods and propose an effectively hybrid algorithm (TNHA) to solve nonlinear equations. The TNHA both obtains the strong global optimization ability by TLBO and the strong local convergence by Newton method. Numerical examples also show that the TNHA not only has fast convergence speed and high precision, but also finds all the solutions of equations in the given interval.

The rest of this paper is organized as follows. The brief introductions of TLBO algorithm and Newton method are given in Section 2 and Section 3 respectively. Following by Section 4, we present the hybrid algorithm TNHA and give the detailed steps of the algorithm. The numerical experiments and result discussion are provided in Section 5. Finally, Section 6 presents conclusions resulting from the study.

2. Teaching-Learning-Based Optimization (TLBO).

Teaching-Learning-Based Optimization (TLBO) is based on the effect of the influence of a teacher on learners in a class. Like other nature-inspired algorithms, the TLBO is also a population-based method that uses a population of solutions to proceed to the global solution, but the method has no user-defined parameter. A group of learners is considered as the population (X). Every learner is considered as an individual (X_i, $i = 1 : S$, where S is the population size).

In the TLBO algorithm, different subjects offered to learners are considered as different design variables. The learning result of a learner is analogous to the 'fitness' (function value $f(X_i)$, $i = 1 : S$) as in other optimization algorithms. The teacher is considered as the most knowledgeable person in a class who shares his/her knowledge with the students to improve the marks of a class. The quality of the learners is evaluated by the mean value of the student's mark in a class. There are two parts in the TLBO: 'Teacher Phase' and 'Learner Phase'. The Teacher Phase means learning from the teacher and the Learner Phase means learning through the interaction between learners.

2.1. Teacher Phase.

During the Teacher Phase, the teaching ($X_{teacher}$) is assigned to the best individual, whose 'fitness' ($f(X_{teacher})$) is best in a class. A teacher tries to enhance the mean value (X_{mean}) of a class up to his/her level. However, practically, it can be done to some extent according to the learning capability of the class. Suppose X_i and $X_{new,i}$, $i = 1 : S$ respectively denote the previous marks of every learner and his/her new marks through learning from a teacher. The Teacher Phase is formulated as

$$X_{new,i} = X_i + r_i(X_{Teacher} - T_F X_{mean}), \tag{1}$$

where $r_i \in [0, 1]$ is a random number, and T_F is a teaching factor. T_F is either 1 or 2, and it can be designed as follows

$$T_F = round[1 + rand(0, 1)]. \tag{2}$$

Accept $X_{new,i}$, if it gives a better function value. (When solving minimization problems, we accept $X_{new,i}$, if $f(X_{new,i}) < f(X_i)$. The reverse is true for maximization problems.)

2.2. Learner Phase.

Learners increase their knowledge by two different means: one through input from the teacher and the other through the interaction between themselves. A learner interacts randomly with other learners with the help of group discussions, presentations, formal communications, etc. A learner (X_i) learns something new by (3), if the other learner X_h has more knowledge than him or her. Otherwise, X_i is moved away from X_h by (4). The Learner Phase is expressed as

For $i = 1 : S$ randomly select another learner X_h, such that $i \neq h$
 if $f(X_i) < f(X_h)$

$$X_{new,i} = X_i + r_i(X_i - X_h), \tag{3}$$

 else

$$X_{new,i} = X_i + r_i(X_h - X_i). \tag{4}$$

 end if
End

Accept $X_{new,i}$, if it gives a better function value. The algorithm will continue its iterations until reaching the maximum number of generations. (The above equations are for minimization problems.)

3. **Newton Method for Solving Equations.** Newton method is one of the most effective methods for solving nonlinear equations. If the initial value given is fully close to the exact solution, Newton method is at least quadratic convergence [1]. Therefore, Newton method is local convergent and has the fast convergent speed. Its basic principle is given as follows. Newton method is actually a linear method. For the equations $F(x) = 0$, its basic idea is to change $F(x) = 0$ into linear equations. Let $x^{(k)}$ be the approximate solution of $F(x) = 0$, and expand function $F(x)$ at $x^{(k)}$. We can get

$$F(x) \approx F\left(x^{(k)}\right) + F'\left(x^{(k)}\right)\left(x - x^{(k)}\right). \tag{5}$$

Then, the equations $F(x) = 0$ can be described as

$$F\left(x^{(k)}\right) + F'\left(x^{(k)}\right)\left(x - x^{(k)}\right) = 0. \tag{6}$$

This is a system of linear equations, whose solution $x = x^{(k+1)}$ can be solved by (6)

$$x^{(k+1)} = x^{(k)} - \left[F'\left(x^{(k)}\right)\right]^{-1} F\left(x^{(k)}\right). \tag{7}$$

If we give initial value $x^{(0)}$, we can gain $x^{(1)}, x^{(2)}, \ldots$

4. **A Hybrid Algorithm Based on TLBO and Newton Method (TNHA).** When initial value given is close to the exact solution, Newton method converges fast. However, the method is sensitive to initial point. If the selection of initial value is improper, the method fails to find the solution of equations. From another point of view, TLBO algorithm has the advantages of simple algorithm, less parameters, strong global convergence ability and insensitivity to initial point. Thus, considering the advantages and disadvantages of the two methods, this paper designs a hybrid algorithm based on TLBO and Newton algorithm (TNHA) for solving nonlinear equations. The hybrid algorithm combines two mechanisms of global optimization and locally deep iteration. Therefore, not only can it give full play to the global convergence by TLBO, but also has the advantages of strong local convergence and high accuracy of the classical Newton method.

Let nonlinear equations be

$$F(x) = [f_1(x), f_2(x), \ldots, f_n(x)]^T = 0, \tag{8}$$

where $x = (x_1, x_2, \ldots, x_n)^T$, $a_i \leq x_i \leq b_i$, $i = 1, 2, \ldots, n$, and n is the number of equations. Solving (8) is equivalent to solving the following optimization problem

$$\min G(x) = \sum_{j=1}^{n} [f_j(x)]^2. \tag{9}$$

In the TNHA, we solve the optimization problem (9) by TLBO. When the termination criterion is reached, the corresponding solutions are used to the initial values of Newton method. Finally, we can gain the solution of the equations $F(x) = 0$ by (7). The detailed steps of the TNHA are shown as follows.

Step 1: Define the optimization problem and initialize the optimization parameters

1) Population size S;
2) Maximum number of generations N_1 for the TLBO;
3) Number of design variables D;
4) Limits of design variables $a_j \leq X_{i,j} \leq b_j$, $i = 1, 2, \cdots, S$, $j = 1, 2, \cdots, D$.

Step 2: Randomly initialize the population X, according to the population size and the number of variables.

Step 3: The Teacher Phase: Calculate the mean of the population column wise, which will give the mean x_{mean} for the particular subject. The teacher will try to shift the mean from x_{mean} towards $x_{teacher}$, whose fitness is the best of the population. So the difference between two values is expressed as

$$Differencemean_i = r_i[x_{teacher} - (T_F \cdot x_{mean})]. \tag{10}$$

where T_F is calculated by (2). The resulting difference is added to the current solution as a new value, thereby improving the existing solution

$$X_{new,i} = X_{old,i} + Differencemean_i. \tag{11}$$

Step 4: The Learner Phase: As explained above, learners increase their knowledge with the help of their mutual interactions. The mathematical expression is explained in Section 2. Obtain $X_{new,i}$ after the student phase.

Step 5: If the maximum number of iterations N_1 is reached, TLBO algorithm is stopped; otherwise, the iteration is repeated from Step 3.

Step 6: The current global optimal individual $X_{new,i} = x^{(0)}$ is the initial point of Newton method, and the iteration is carried out in order to achieve the global optimal value.

Step 7: $x^{(1)} = x^{(0)} - \left[F'\left(x^{(0)}\right)\right]^{-1} F\left(x^{(0)}\right)$.

Step 8: If the maximum number of iterations N_2 is reached for Newton method, then output $x^{(1)}$; otherwise, $x^{(0)} \leftarrow x^{(1)}$, return back to Step 7.

5. Numerical Experiments and Result Analysis.

In this part, we choose three nonlinear equations to test the optimization performance of the TNHA. In order to verify its high-efficiency, we compare the TNHA with the algorithms in references, TLBO and Newton method. For the TNHA and TLBO, the population size is defined as 20. The maximum numbers of iterations N_1 and N_2 are 1000 and 20 respectively. In order to effectively reduce the influence of random disturbance, every algorithm runs independently 50 times. The mean value of solutions is viewed as the final result.

Example 1 [16] $\begin{cases} f_1(x) = x_1^2 - x_2 + 1 = 0 \\ f_2(x) = x_1 - \cos(0.5\pi x_2) = 0, \end{cases}$

where $-2 \leq x_1, x_2 \leq 2$. The exact solutions are $x_1^* = \left(-1/\sqrt{2}, 1.5\right)^T$, $x_2^* = (0, 1)^T$ and $x_3^* = (-1, 2)^T$.

Example 1 is a system of transcendental equations. The computational results are shown in Table 1. As seen from Table 1, the TNHA finds three families of solutions for Example 1, namely all the solutions, but [16] only finds two families of solutions. Success rate of the TNHA is 100%, but that of [16] is only 86%. The mean of solutions got by TNHA in this paper is closer to the exact solution than [16]. What is more, the numbers of every solution found are similar. Therefore, TNHA has higher precision and better stability than [16]. On the whole, the TNHA is greatly super to [16].

Example 2 [17] $\begin{cases} f_1(x) = x_1 + x_2 - 2x_3 = 0 \\ f_2(x) = x_1 x_2 - 1 = 0 \\ f_3(x) = x_1^2 + x_2^2 - 2 = 0, \end{cases}$

where $0 \leq x_1, x_2, x_3 \leq 2$. The exact solution is $x^* = (1, 1, 1)^T$.

Table 1. Comparison of the TNHA with [16]

Algorithm	Number of search solutions	Success rate	Mean of x_1	Mean of x_2	Number of solutions
[16]	43	86%	-0.77024	1.560608	26
			0.010104	0.989197	17
TNHA	50	100%	-0.7071	1.5000	20
			0.0000	1.0000	13
			-1.0000	2.0000	17
Exact solution			$-1/\sqrt{2}$	1.5	
			0	1	
			-1	2	

Table 2. Comparison of the TNHA with [17]

Algorithm	Mean of x_1	Mean of x_2	Mean of x_3	$\|x - x^*\|$
[17]	1.0175	0.9822	0.9999	0.0250
TNHA	1.0000	1.0000	1.0000	2.2360e-09
Exact solution	1	1	1	0

Table 3. Comparison of the TNHA with the TLBO

Algorithm	Success rate	Mean of x_1	Mean of x_2	Mean of x_3	$\|x - x^*\|$
TLBO	100%	1.0019	0.9976	0.9999	0.0031
TNHA	100%	1.000000	1.000000	1.000000	2.2360e-09
Exact solution		1	1	1	0

Example 2 is a system of algebraic equations. Table 2 is the comparison result of the TNHA with [17]. The mean of solutions got by TNHA in this paper is closer to the exact solution than [17], so the TNHA has very high precision.

Table 3 is the comparison result of the TNHA with single TLBO. Although the success rate is the same, the TNHA obtains a higher accuracy than the TLBO by deep iteration of Newton method. Thus, the hybrid algorithm in this paper gets more accurate solution than the TLBO alone.

Example 3 [16]
$$\begin{cases} f_1(x) = (x_1 - 5x_2)^2 + 40\sin^2(10x_3) = 0 \\ f_2(x) = (x_2 - 2x_3)^2 + 40\sin^2(10x_1) = 0 \\ f_3(x) = (3x_1 + x_3)^2 + 40\sin^2(10x_2) = 0, \end{cases}$$
where $-1 \le x_1, x_2, x_3 \le 1$. The exact solution is $x^* = (0, 0, 0)^T$.

Table 4 compares the TNHA with [16], TLBO and Newton method. By view of success rate, the TNHA and TLBO is 100%, but that of [16] is only 54% and the lowest is Newton method. Because the Newton method is very strict with the selection of the initial value, whose success rate is only 4%. The TNHA and TLBO can find the approximate solution every time, so the reliability and stability are best. From another point of view, the TNHA has the highest precision, TLBO is the second and [16] is the worst in four algorithms. In general, the TNHA has the great stability, fast convergence speed and high precision. Therefore, the TNHA is a good selection for solving nonlinear equations.

6. **Conclusions.** The paper introduces the principles of TLBO and Newton method at first. Then, considering advantages and disadvantages of two methods above, this paper proposes a hybrid algorithm TNHA for solving nonlinear equations. The TNHA both gives full play to the global research ability of the TLBO and keeps the deep iteration of

Table 4. Comparison of the TNHA with [16], TLBO and Newton method

Algorithm	Number of search solutions	Success rate	Mean of x_1	Mean of x_2	Mean of x_3	$\|x - x^*\|$
[16]	27	54%	−0.000001	−0.000012	−0.000001	1.2083e-05
TLBO	50	100%	−0.3001e-39	0.5857e-39	−0.0485e-39	6.5991e-40
Newton method	2	4%	−0.1238e-7	−0.1904e-7	0.0784e-7	2.4026e-08
TNHA	50	100%	−0.2862e-45	0.0463e-45	0.558611e-45	6.2934e-46
Exact solution			0	0	0	0

Newton method. At last, we test the performance of the TNHA by numerical examples. The experiments show that the TNHA not only shows better accuracy and stability than the algorithms in the references and Newton method, but also can detect all the solution of equations in given the interval. Moreover, the TNHA has higher precision than the single TLBO because of the deep iteration of Newton method. Therefore, the TNHA is greatly effective for solving nonlinear equations, and its design strategy is a good reference for solving practical problems. Further practical engineering application of the TNHA in areas of computer engineering and material structure design can shed further light and is left for future research.

Acknowledgment. This work is funded by the National Natural Science Foundation of China (Grant No. 61403331), the National Natural Science Foundation of China (No. 6157 3306) and Teaching Research Project of Hebei Normal University of Science and Technology (No. JYZD201413).

REFERENCES

[1] Q. Y. Li, N. C. Wang and D. Y. Yi, *Numerical Analysis*, Tsinghua University Press, Beijing, 2008.
[2] J. Liu and L. Tang, A modified genetic algorithm for single machine scheduling, *Computers and Industrial Engineering*, vol.37, pp.43-36, 1999.
[3] H. C. Li, S. L. Xie and Z. S, Yi, Theory and application of genetic algorithm, *Mining Metallurgy*, vol.14, pp.87-90, 2005.
[4] Y. L. Lin, W. D. Chang and J. G. Hsieh, A particle swarm optimization approach to nonlinear rational filter modeling, *Expert Systems with Applications*, vol.34, no.2, pp.1194-1199, 2008.
[5] P. Chakraborty, S. Das, G. G. Roy and A. Abraham, On convergence of the multi-objective particle swarm optimizers, *Information Sciences*, vol.181, pp.1411-1425, 2011.
[6] L. Liu, S. Yang and D. Wang, Force-imitated particle swarm optimization using the near-neighbor effect for locating multiple optima, *Information Sciences*, vol.182, pp.139-155, 2011.
[7] E. Rashedi, H. N. Pour and S. Saryazdi, GSA: A gravitational search algorithm, *Information Sciences*, vol.179, pp.2232-2248, 2009.
[8] E. Rashedi, H. N. Pour and S. Saryazdi, BGSA: Binary gravitational search algorithm, *Natural Computing*, vol.9, no.3, pp.727-745, 2010.
[9] R. V. Rao, V. J. Savsani and D. P. Vakharia, Teaching-learning-based optimization: An optimization method for continuous non-linear large scale problems, *Information Sciences*, vol.183, pp.1-15, 2012.
[10] R. V. Rao, V. J. Savsani and D. P. Vakharia, Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems, *Computer-Aided Design*, vol.43, pp.303-315, 2011.
[11] R. V. Rao and V. Patel, An elitist teaching-learning-based optimization algorithm for solving complex constrained optimization problems, *Journal of Industrial Engineering Computations*, vol.3, pp.1153-1163, 2012.
[12] B. Amiri, Application of teaching-learning-based optimization algorithm on cluster analysis, *Journal of Basic and Applied Scientific Research*, vol.2, no.3, pp.11795-11802, 2012.
[13] P. Chakraborty, S. Das, G. G. Roy and A. Abraham, On convergence of the multi-objective particle swarm optimizers, *Information Sciences*, vol.181, pp.1411-1425, 2011.

[14] R. V. Rao and V. D. Kalyankar, Parameter optimization of modern machining processes using teaching-learning-based optimization algorithm, *Engineering Applications of Artificial Intelligence*, vol.26, pp.524-531, 2013.
[15] R. V. Rao and V. Patel, Multi-objective optimization of two stage thermoelectric cooler using a modified teaching-learning-based optimization algorithm, *Engineering Applications of Artificial Intelligence*, vol.26, pp.430-445, 2013.
[16] J. K. Zhang, X. Z. Wang and S. Y. Liu, Particle swarm optimization for solving nonlinear equation and system, *Computer Engineering and Applications*, vol.7, pp.56-58, 2006.
[17] Y. Zen, The application of improved genetic algorithms to solving non-linear equation group, *Journal of East China Jiaotong University*, vol.21, no.4, pp.132-134, 2004.