Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2013, Article ID 757391, 18 pages
http://dx.doi.org/10.1155/2013/757391

Research Article

A Novel Differential Evolution Invasive Weed Optimization Algorithm for Solving Nonlinear Equations Systems

Yongquan Zhou,[1,2] Qifang Luo,[1] and Huan Chen[1]

[1] *College of Information Science and Engineering, Guangxi University for Nationalities, Nanning, Guangxi 530006, China*
[2] *Guangxi Key Laboratory of Hybrid Computation and IC Design Analysis, Nanning, Guangxi 530006, China*

Correspondence should be addressed to Yongquan Zhou; yongquanzhou@126.com

Received 4 September 2013; Revised 2 November 2013; Accepted 9 November 2013

Academic Editor: Chong Lin

In view of the traditional numerical method to solve the nonlinear equations exist is sensitive to initial value and the higher accuracy of defects. This paper presents an invasive weed optimization (IWO) algorithm which has population diversity with the heuristic global search of differential evolution (DE) algorithm. In the iterative process, the global exploration ability of invasive weed optimization algorithm provides effective search area for differential evolution; at the same time, the heuristic search ability of differential evolution algorithm provides a reliable guide for invasive weed optimization. Based on the test of several typical nonlinear equations and a circle packing problem, the results show that the differential evolution invasive weed optimization (DEIWO) algorithm has a higher accuracy and speed of convergence, which is an efficient and feasible algorithm for solving nonlinear systems of equations.

1. Introduction

Systems of nonlinear equations arise in many domains of practical importance such as engineering, mechanics, medicine, chemistry, and robotics. Solving such a system involves finding all the solutions (there are situations when more than one solution exists) of the polynomial equations contained in the mentioned system. The algorithms of solving nonlinear equations systems are worse than linear equations in convergence speed and ratio, especially solving nonconvex nonlinear equations. The traditional solutions of nonlinear equations include Newton-Raphson method, Quasi-Newton method, and homotopy method. Newton-Raphson method is a much more classical method, but it is sensitive to initial iteration value. In addition, not only it requires a large amount of calculation, but also sometimes it accompanied by difficulty calculation. Quasi-Newton method is to solve the difficult caused by Jacobi matrix. It has now become one of the most effective methods that solve nonlinear equations and optimization problems. While, its stability is poor and sometimes its iterative effect is not well. The basic idea of homotopy method is to start from easily solved equations and then gradually transit to the original equations and get the solution of problems. In recent years, with the rapid development of computational intelligence, computational intelligence techniques have also been used to solve nonlinear equations, such as genetic algorithm [1–3], particle swarm optimization algorithm [4], differential evolution algorithm [5], artificial fish-swarm algorithm [6], artificial bee colony algorithm [7] harmony search algorithm [8], and probabilistic-driven search (PDS) algorithm [9]. These swarm intelligent algorithm has several advantages when adopted for searching solutions for systems of nonlinear equations: their does not require of a "good" initial point to perform the search, and the search space can be bounded by lower and upper values for each decision variable. Additionally, no continuity or differentiability of the objective function is required. What can be considered as the main disadvantage of swarm intelligent algorithm in this sort of application is its relatively poor accuracy, which is caused by the coarse granularity of the search performed by the algorithm. This can, of course, be improved either by running the swarm intelligent algorithm for a larger number of iterations (although at a higher computational cost) or by postprocessing the solution produced by the swarm intelligent algorithm with a traditional numerical optimization technique.

In 2006, a novel stochastic optimization model, invasive weed optimization (IWO) algorithm [10], was proposed by Mehrabian and Lucas, which is inspired from a common phenomenon in agriculture: colonization of invasive weeds. The algorithm was inspired from colonizing weeds, which is used to mimic the natural behavior of weeds in colonizing and occupying suitable places for growth and reproduction. It has the robustness, adaptation, and randomness and is simple but effective with an accurate global search ability. So far, it has been applied in many engineering fields.

In this paper, a new hybrid algorithm based on the population diversity of IWO and the heuristic differential evolution (DE) is presented to solve nonlinear systems of equation. With the population diversity, which enhances the global searching ability of algorithm, and heuristic method, which improves the local mining capacity, the hybrid algorithm gets a higher optimization accuracy and faster convergence speed. The performance of the proposed approach is evaluated for several well-known benchmark problems from kinematics, chemistry, combustion, and medicine. Numerical results reveal the efficiency of the proposed approach and its flexibility to solve large-scale systems of equations.

2. Descriptions of Nonlinear Equation Systems

The general form of nonlinear equations systems can be described as follows:

$$\begin{aligned} f_1(X) &= 0, \\ f_2(X) &= 0, \\ &\vdots \\ f_n(X) &= 0, \end{aligned} \tag{1}$$

where $X = (x_1, x_2, \ldots, x_n) \in D \in R^n$ and $D = \{(x_1, x_2, \ldots, x_n) \mid a_i \le x_i \le b_i,\ i = 1, 2, \ldots, n\}$. The following fitness function is used to calculate conveniently

$$f(X) = \sum_{i=1}^{n} f_i^2(X), \tag{2}$$

where $f_i(X)$, $i = 1, 2, \ldots, n$ are nonlinear functions. Because there are many equality constraints, the system of equations usually has no solution X such that $f_i(X) = 0$, $i = 1, 2, \ldots, n$. Thus, we find an approximate solution of simultaneous equations such that $|f(X)| < \varepsilon$, where ε is an arbitrary small positive number. In order to do so, we define the fitness function error:

$$\varepsilon = \left|0 - f(X)\right|. \tag{3}$$

So, the smaller the fitness error is, the higher the solution quality is. The fitness error of theoretical solutions is 0. Thus, nonlinear equation systems problems can be formulated as solving function optimization

$$\begin{aligned} &\min \quad f(X) \\ &\text{s.t.} \quad X = (x_1, x_2, \ldots, x_n) \in D. \end{aligned} \tag{4}$$

In this way, the best value of formula (4) is the solution of (1).

3. Algorithm of Invasive Weed Optimization

In the basic IWO, weeds represent the feasible solutions of problems and population is the set of all weeds. A finite number of weeds are being dispread over the search area. Every weed produces new weeds depending on its fitness. The generated weeds are randomly distributed over the search space by normally distributed random numbers with a mean with a mean equal to zero equal to zero. This process continues until the maximum number of weeds is reached. Only the weeds with better fitness can survive and produce seed, others are being eliminated. The process continues until maximum iterations are reached or hopefully the weed with the best fitness is closest to the optimal solution. The process is addressed in detail as follows.

Step 1 (initialize a population). A population of initial solutions is being dispread over the D-dimensional search space with random positions.

Step 2 (reproduction). The higher the weed's fitness is, the more seeds it produces. The formula of weeds producing seeds is

$$\text{weed}_n = \frac{f - f_{\min}}{f_{\max} - f_{\min}} (s_{\max} - s_{\min}) + s_{\min}, \tag{5}$$

where f is the current weed's fitness. $f_{\max}$ and $f_{\min}$, respectively, represent the maximum and the least fitness of the current population. $s_{\max}$ and $s_{\min}$, respectively, represent the maximum and the least value of a weed.

Step 3 (spatial dispersal). The generated seeds are randomly distributed over the D-dimensional search space by normally distributed random numbers with a mean equal to zero but with a varying variance. This ensures that seeds will be randomly distributed so that they abide near to the parent plant. However, standard deviation (σ) of the random function will be reduced from a previously defined initial value (σ_{init}) to a final value (σ_{final}) in every generation. In simulations, a nonlinear alteration has shown satisfactory performance given as follows:

$$\sigma_{\text{cur}} = \frac{(\text{iter}_{\max} - \text{iter})^n}{(\text{iter}_{\max})^n} (\sigma_{\text{init}} - \sigma_{\text{final}}) + \sigma_{\text{final}}, \tag{6}$$

where $\text{iter}_{\max}$ is the maximum number of iterations, σ_{cur} is the standard deviation at the present time step, and n is the nonlinear modulation index. Generally, n is set to 3.

```
BEGIN
    Input: randomly initialize a population of DEIWO individuals.
  While (the termination criterion is not satisfied)
    Calculate every individual's fitness f(X);
    Calculate the number of every individual's offspring
        weed_n = (f − f_min)/(f_max − f_min) (s_max − s_min) + s_min;
    Update σ_cur by
        σ_cur = (iter_max − iter)^n/(iter_max)^n (σ_init − σ_final) + σ_final;
    Generate seeds over the search space by N(0, σ_cur),
    Add the generated seeds into the solution set;
      If (the number of all weeds and seeds equal the maximum population P_MAX)
          Sort the population in descending order by fitness;
          Truncate population of weeds with higher fitness until the number of set equal the
          maximum population;
      End If
    Mutate the individual by
        V_i^t = X_i^t + F × (X_best^t − X_i^t) + F × (X_r1^t − X_r2^t);
    Cross the individual by
        U_{i,j}^t = { V_{i,j}^t   if (rand_j[0, 1] ≤ CR or j = j_rand)     j = 1, 2, ..., n;
                    { X_{i,j}^t   otherwise,
    Select individual between U_i^t and X_i^t
        X_i^{t+1} = { U_i^t   if f(U_i^t) < f(X_i^t)
                    { X_i^t   otherwise;
  End While
    Output the best individual and its fitness;
END
```

Pseudocode 1: The pseudocode for DEIWO algorithm.

Step 4 (competitive exclusion). After passing some iteration, the number of weeds in a colony will reach its maximum (P_MAX) by fast reproduction. At this time, each weed is allowed to produce seeds. The produced seeds are then allowed to spread over the search area. When all seeds have found their position in the search area, they are ranked together with their parents (as a colony of weeds). Next, weeds with lower fitness are eliminated to reach the maximum allowable population in a colony. In this way, weeds and seeds are ranked together and the ones with better fitness survive and are allowed to replicate. The population control mechanism also is applied to their offspring to the end of a given run, realizing competitive exclusion.

4. Differential Evolution Invasive Weed Optimization

Reviewing the IWO, some initial weeds are dispread over the search space randomly and then produce new individuals (seeds). We select the better plants from the population consisting of weeds and seeds. The process continues until maximum number of plants is reached. In order to speed up the optimization process, differential evolution (DE) [11] is used to cooperate with IWO so that every weed in iteration can move towards the best individual of the current iteration. In this way, the algorithm designated as DEIWO not only ensures the individual diversity by IWO, but also improves the optimization accuracy and the speed by DE (see Pseudocode 1).

4.1. Differential Evolution. Differential evolution (DE), introduced by Storn and Price in 1997, is one of the most prominent new generation EAs for solving real-valued optimization problems. Using only a few control parameters, DE offers exclusive advantages, such as a simple and easy-to-understand concept, ease of implementation, powerful search capability, and robustness. The main procedure of DE includes mutation, crossover, and selection. These operators are based on natural evolution principles in order to keep the population diversity, as well as to avoid premature convergence. In this way, mutation and crossover operators are used to generate new vectors. Then, the selection operator determines which vectors will survive in the next generation. This procedure is repeated until a stopping condition is reached. The basic principle of DE is given as follows.

Suppose the objective function can be expressed as follow.

$$\text{Minimize } f(X). \tag{7}$$

The problem space is $S = \{X \mid X = (x_1, x_2, \ldots, x_i, \ldots, x_n)\}$, where $a_i \le x_i \le b_i$, $i = 1, 2, \ldots, n$; for each parent vector X_i^t a mutant vector is generated according to

$$V_i^t = X_i^t + F \times \left(X_{\text{best}}^t - X_i^t\right) + F \times \left(X_{r1}^t - X_{r2}^t\right), \tag{8}$$

Table 1: Coefficients a_{ki} for the kinematic application.

	$i = 1$	$i = 2$	$i = 3$	$i = 4$
$k = 1$	−0.249150680	+0.125016350	−0.635550077	+1.48947730
$k = 2$	+1.609135400	−0.686607360	−0.115719920	+0.23062341
$k = 3$	+0.279423430	−0.119228120	−0.666404480	+1.32810730
$k = 4$	+1.434801600	−0.719940470	+0.110362110	−0.25864503
$k = 5$	+0.000000000	−0.432419270	+0.290702030	+1.16517200
$k = 6$	+0.400263840	+0.000000000	+1.258776700	−0.26908494
$k = 7$	−0.800527680	+0.000000000	−0.629388360	+0.53816987
$k = 8$	+0.000000000	−0.864838550	+0.581404060	+0.58258598
$k = 9$	+0.074052388	−0.037157270	+0.195946620	−0.20816985
$k = 10$	−0.083050031	+0.035436896	−1.228034200	+2.68683200
$k = 11$	−0.386159610	+0.085383482	+0.000000000	−0.69910317
$k = 12$	−0.755266030	+0.000000000	−0.079034221	+0.35744413
$k = 13$	+0.504201680	−0.039251967	+0.026387877	+1.24991170
$k = 14$	−1.091628700	+0.000000000	−0.057131430	+1.46773600
$k = 15$	+0.000000000	−0.432419270	−1.162808100	+1.16517200
$k = 16$	+0.049207290	+0.000000000	+1.258776700	+1.07633970
$k = 17$	+0.049207290	+0.013873010	+2.162575000	−0.69686809

where parent vector X_i^t, X_{r1}^t, X_{r2}^t are mutually exclusive individuals. The parent vector X_{best}^t is the best individual of the current iteration. The scaling factor F is a positive constant which controls the amplification of the difference vector, and $F \in [0, 2]$. After mutation, the trial vector is generated using the parent and mutated vectors as follows:

$$U_{i,j}^t = \begin{cases} V_{i,j}^t & \text{if}\left(\text{rand}_j\,[0,1] \le \text{CR or } j = j_{\text{rand}}\right) \\ X_{i,j}^t & \text{otherwise,} \end{cases} \quad j = 1, 2, \ldots, n, \tag{9}$$

where $\text{CR} \in [0, 1]$ is a real-valued crossover rate constant and $j_{\text{rand}} \in [1, n]$ is a random integer.

The individual of the next generation is generated as the follows:

$$X_i^{t+1} = \begin{cases} U_i^t, & \text{if } f\left(U_i^t\right) < f\left(X_i^t\right), \\ X_i^t, & \text{otherwise,} \end{cases} \tag{10}$$

where $f(x)$ denotes the fitness function.

4.2. The Hybrid Strategy of DE and IWO. The hybrid strategy of DE and IWO can be summarized as follows. Every initial weed produces new seeds depending on its fitness. The generated seeds are randomly distributed over the search space by normally distributed random numbers. This process continues until the maximum number of weeds is reached. Only the weeds with better fitness can survive and produce seed, when the others are being eliminated. We can see that IWO ensures the diversity of individuals and the worse individuals also have the opportunity to reproduce, while individuals of DE move toward the best individual, using the information of distance and direction obtained from the current population. We can conclude that DE improves the convergence speed and searching precision. So, combining DE and IWO is useful to enrich the search behavior of optimization process and get high quality solutions.

Table 2: Parameters values used by the DEIWO.

G_SIZE	P_MAX	σ_{final}	$s_{\max}$	$s_{\min}$	n	F	CR
10	15	0.01	15	1	4	2	0.9

5. Experiments and Results

This section reports several experiments and comparisons using the proposed approach. Some well-known applications are also considered in the subsequent section.

5.1. Testing Platform. The experimental program testing platform included processor: CPU Intel Core i3-370, frequency: 2.40 GHz, memory: 4 GB, operating system: Windows 7, and run software: MATLAB 7.6.

5.2. Testing Nonlinear Equation Systems. In order to test the performance of DEIWO for solving nonlinear equation systems, 8 nonlinear equation systems in the literature are used and the testing results are compared with the literature [12]. Table 2 shows the experiment parameters. The search area and the number of iterations are as shown in Table 3. Table 4 to Table 11 is the results of 8 equation systems.

Example 1. Consider the following nonlinear system:

$$\begin{aligned} f_1\left(x_1, x_2\right) &= \cos\left(2x_1\right) - \cos\left(2x_2\right) - 0.4 = 0, \\ f_2\left(x_1, x_2\right) &= 2\left(x_2 - x_1\right) + \sin\left(2x_2\right) - \sin\left(2x_1\right) - 1.2 = 0. \end{aligned} \tag{11}$$

Table 3: Benchmarks used in the experiments.

Benchmark	Number of variables	Variables range	Number of iterations
(1) Interval arithmetic	10	$[-2, 2]$	2000
(2) Neurophysiology application	6	$[-10, 10]$	800
(3) Chemical equilibrium application	5	$[-10, 10]$	800
(4) Kinematic application	8	$[-10, 10]$	800
(5) Combustion application	10	$[-10, 10]$	800
(6) Economics modeling application	20	$[-10, 10]$	800

Table 4: Comparison of results for Example 1.

Method	Solution	Functions values	Error
Newton	(0.15, 0.49)	(−0.00168, 0.01497)	0.00022156
Secant	(0.15, 0.49)	(−0.00168, 0.01497)	0.00022156
Broyden	(0.15, 0.49)	(−0.00168, 0.01497)	0.00022156
Effati	(0.1575, 0.4970)	(−0.005455, 0.00739)	0.000084369
EA [12]	(0.157772, 0.49458)	(0.001264, 0.000969)	0.0000025366
PDS [9]	(0.156520, 0.493376)	(−0.0000005815, −0.0000008892)	0.0000003477
IWO			
500 iterations	(0.1564457783, 0.4933795366)	(0.000051022, 0.00029976)	0.00000009256
800 iterations	(0.1565388795, 0.4933910072)	(0.000012829, −0.000028008)	0.00000000009
DEIWO			
500 iterations	(0.1565200697, 0.4933763742)	(−0.0000000000, −0.000000000)	0.0000000000
800 iterations	(0.1565200697, 0.4933763742)	(−0.000000000, −0.000000000)	0.0000000000

Table 5: Comparison of results for Example 2.

Method	Solution	Functions values	Error
Effati	(0.0096, 0.9976)	(−0.019223, 0.016776)	0.00065095
EA [12]	(−0.00138, 1.0027)	(−0.00276, −6.37e − 005)	0.0000076216
PDS [9]	(0.0, 1.0)	(0, 0)	0
IWO			
500 iterations	(0.0000353012, 1.0000971972)	(0.000070606, 0.0001678)	0.0000000033
800 iterations	(0.0000535628, 0.9999772727)	(0.00010712, 0.00084396)	0.0000000186
DEIWO			
500 iterations	(0.0000000000, 1.0000000000)	(0.0000000000, 0.00000000000)	0.0000000000
800 iterations	(0, 1)	(0, 0)	0

The results obtained by applying Newton, Secant, Broyden, and Effati methods; evolutionary approach (EA); Probabilistic-driven search; IWO; and the proposed DEIWO method are presented in Table 4.

Example 2. Consider the following nonlinear system:

$$\begin{aligned} f_1(x_1, x_2) &= e^{x_1} + x_1 x_2 - 1 = 0, \\ f_2(x_1, x_2) &= \sin(x_1 x_2) + x_1 + x_2 - 1 = 0. \end{aligned} \tag{12}$$

The results obtained by Effati, evolutionary approach (EA), probabilistic-driven search, IWO, and the proposed DEIWO method are given in Table 5.

Example 3 (interval arithmetic benchmark). We consider one benchmark problem proposed from interval arithmetic [13]. The benchmark consists of the following system of equations:

$$\begin{aligned} 0 &= x_1 - 0.25428722 - 0.18324757 x_4 x_3 x_9, \\ 0 &= x_2 - 0.37842197 - 0.16275449 x_1 x_{10} x_6, \\ 0 &= x_3 - 0.27162577 - 0.16955071 x_1 x_2 x_{10}, \\ 0 &= x_4 - 0.19807914 - 0.15585316 x_7 x_1 x_6, \\ 0 &= x_5 - 0.44166728 - 0.19950920 x_7 x_6 x_3, \end{aligned}$$

Table 6: Comparison of results for interval arithmetic benchmark.

Algorithm	EA			DEIWO		
	Solution	Functions values	Error	Solution	Functions values	Error
Solution 1	0.0464905115	0.2077959240	0.5578712619	0.266267198585396	0.008394489588241	0.00009717937
	0.1013568357	0.2769798846		0.378860657321047	0.002282535415509	
	0.0840577820	0.1876863212		0.279521817014353	0.000606950630162	
	−0.1388460309	0.3367887114		0.201184236934515	0.00045403887095	
	0.4943905739	0.0530391321		0.443621074720278	0.001608785041108	
	−0.0760685163	0.2223730535		0.147345385930806	0.001919901952144	
	0.2475819110	0.1816084752		0.433560209877767	0.001477874245750	
	−0.0170748156	0.087896386		0.076147905678193	0.002678016787395	
	0.0003667535	0.3447200366		0.347936915389233	0.001950056578385	
	0.1481119311	0.2784227489		0.426164139956603	0.001222532385146	
Solution 2	0.1224819761	0.1318552790	0.4739431843	0.259070856110057	0.001390867407193	0.0000879354
	0.1826200685	0.1964428361		0.382754529706839	0.001670672347103	
	0.2356779803	0.0364987069		0.274163620083770	0.004670232165184	
	−0.0371150470	0.2354890155		0.194046880920087	0.006594528618355	
	0.3748181856	0.0675753064		0.446234352737533	0.001096002561125	
	0.2213311341	0.0739986588		0.147250545938791	0.001859451563865	
	0.0697813043	0.3607038292		0.430956789693803	0.000981853619901	
	0.0768058043	0.0059182979		0.070959777303814	0.002412801063046	
	−0.0312153867	0.3767487763		0.348016260950456	0.002088606106882	
	0.1452667120	0.2811693568		0.428727381578941	0.001450589369406	
Solution 3	0.0633944399	0.1908436653	0.6396343329	0.258177973482307	0.000279235911393	0.0000920236
	0.1017426933	0.2767897367		0.379453934641213	0.001675106786883	
	−0.1051842285	0.3769063436		0.282106464391719	0.003400908452008	
	−0.0477059943	0.2460900702		0.203187486961515	0.002444022146960	
	0.4149858326	0.0260337751		0.440886347480303	0.004507666000145	
	0.1215195321	0.0256054760		0.151149001651968	0.001912206205502	
	0.2539777159	0.1761486401		0.438073605204888	0.006015825363839	
	0.0843972823	0.1349869851		0.075807440998570	0.002323012002644	
	−0.0534132992	0.3986395691		0.343828213649516	0.002178696688097	
	0.0880998746	0.3383563536		0.426228284414650	0.001136405771158	
Solution 4	0.1939820199	0.0603335280	1.4462409770	0.256723977550822	0.000978898430504	0.0000663200
	0.0152114400	0.3633514726		0.384592485498424	0.003543546683658	
	0.1618654345	0.1097465792		0.276941817209508	0.001841321143614	
	0.0056985809	0.9114653768		0.197489408658780	0.004176118389889	
	0.1904538879	0.2502358229		0.440994124940054	0.002083758681279	
	−0.1623604033	0.3089460561		0.147051336865180	0.000846490460256	
	0.1864448178	0.2428992222		0.431136854794964	0.000640313677801	
	−0.0449302706	0.1144916285		0.072704222159548	0.005142454609879	
	0.1675935311	0.1774161896		0.340803111821165	0.000247320573489	
	−0.0274959004	0.4539962587		0.427549623883685		
Solution 5	0.1169663983	0.1376466161	0.7766780224	0.255903538082730	0.001871285966801	0.0000789255
	−0.0360324410	0.4148982728		0.382537175786251	0.001561689488107	
	−0.0517944631	0.3233536840		0.280411159251412	0.001688611428279	
	−0.1825907448	0.3816023122		0.195875663823431	0.004661705763413	
	0.0741902056	0.3669485262		0.443780436204940	0.001335014507774	
	0.25036046290	0.1038035643		0.143389617571427	0.005874675479176	
	0.2043019803	0.2250628007		0.429845943240599	0.002252674160797	
	0.0120607075	0.0595255950		0.075841793689244	0.002583241117790	
	0.18799376080	0.1571104516		0.346507927398647	0.00054690647042	
	0.09312965555	0.3333260708		0.427574364325872	0.000247274165109	
Solution 6	0.0600624922	0.1941520526	0.7079949322	0.255799163028461	0.002038827838389	0.0001078856
	0.0665034453	0.3118493104		0.377253755510825	0.003789082463356	
	0.1163378165	0.1553271371		0.277018794733188	0.001521763523132	
	−0.0456993775	0.2437021588		0.200978421031471	0.000354823423825	
	0.1649150798	0.2765628684		0.438492258899619	0.006702404370855	
	−0.1223771045	0.2690505556		0.148958902639442	0.00021411085140	
	0.0666559953	0.3628858561		0.428463690304868	0.003537608346351	
	0.0732866593	0.0028059669		0.075053838513460	0.001700781681794	
	0.0745961823	0.2703511071		0.348037273986552	0.002061545859033	
	0.0578573421	0.3686104960		0.422616629486848	0.004722682745247	

Table 6: Continued.

Algorithm	EA Solution	EA Functions values	EA Error	DEIWO Solution	DEIWO Functions values	DEIWO Error
Solution 7	0.2077500302	0.0464943050	0.6818033772	0.255930313799156	0.001810207471303	0.00007961456
	0.0299198492	0.3489889696		0.375799746775034	0.005285646592094	
	−0.0339491324	0.3058418474		0.277228800096808	0.001381138305712	
	−0.2027950317	0.4012915513		0.197664668158518	0.003001999551435	
	0.2131771707	0.2284027988		0.443579221165137	0.001676029968836	
	0.0568458067	0.0886970244		0.149296154125337	0.000058388800263	
	0.2267650517	0.2024745658		0.434509271234809	0.002374515657373	
	−0.0977041236	0.1687259437		0.078260058512172	0.004859797672831	
	−0.0339921200	0.3787652675		0.343897367370457	0.002133133043030	
	0.2532921324	0.1741025236		0.428288957805820	0.000928066676778	
Solution 8	−0.0364260444	0.2907604740	0.6614924105	0.258312377722712	0.000521318782360	0.0000970288
	0.1232874096	0.2550909534		0.373209790270427	0.007919092691069	
	−0.0349926786	0.3065546443		0.278669128269362	0.000010117982179	
	0.0959206680	0.1020362156		0.197734446091387	0.002933405895119	
	0.2474776135	0.1940393232		0.445854845752568	0.000612576903044	
	0.0877790534	0.0582777294		0.149636829778090	0.000546213255405	
	0.2453311373	0.1832428336		0.429718851893949	0.002127892775812	
	−0.1234286095	0.1938589990		0.070228985270279	0.003172573513494	
	−0.0767543100	0.4216253107		0.347004190872835	0.001068279667729	
	0.0837953112	0.3428082855		0.430286378019452	0.003005330807344	

$$
\begin{aligned}
0 &= x_6 - 0.14654113 - 0.18922793 x_8 x_5 x_{10},\\
0 &= x_7 - 0.42937161 - 0.21180486 x_2 x_5 x_8,\\
0 &= x_8 - 0.07056438 - 0.17081208 x_1 x_7 x_6,\\
0 &= x_9 - 0.34504906 - 0.19612740 x_{10} x_6 x_8,\\
0 &= x_{10} - 0.42651102 - 0.21466544 x_4 x_8 x_1.
\end{aligned} \tag{13}
$$

Parameters used by the DEIWO approach are listed in Tables 2 and 3. The results obtained by evolutionary approach (EA) and the proposed DEIWO method are given in Table 6.

Example 4 (neurophysiology application). We considered the example proposed in [14], which consisted of the following equations:

$$
\begin{aligned}
x_1^2 + x_3^2 &= 1,\\
x_2^2 + x_4^2 &= 1,\\
x_5 x_3^3 + x_6 x_4^3 &= c_1,\\
x_5 x_1^3 + x_6 x_2^3 &= c_2,\\
x_5 x_1 x_3^2 + x_6 x_4^2 x_2 &= c_3,\\
x_5 x_1^2 x_3 + x_6 x_2^2 x_4 &= c_4,
\end{aligned} \tag{14}
$$

where the constant c_i can be randomly chosen. In our experiments, we considered $c_i = 0$, $i = 1, 2, 3, 4$, as in the literature [12]. In [15], this problem is used to show the limitations of Newton's method for which the running time exponentially increases with the size of the initial intervals. We considered the following values for the parameters used by the DEIWO as given in Tables 2 and 3. Some of the solutions obtained by our approach as well as the values of the objective functions for these values are presented in Table 7.

Example 5 (chemical equilibrium application). We consider the chemical equilibrium system as given by the following [16]:

$$
\begin{aligned}
& x_1 x_2 + x_1 - 3x_5 = 0,\\
& 2x_1x_2 + x_1 + x_2x_3^2 + R_8 x_2 - R x_5\\
&\quad + 2R_{10} x_2^2 + R_7 x_2 x_3 + R_9 x_2 x_4 = 0,\\
& 2x_2x_3^2 + 2R_5 x_3^2 - 8x_5 + R_6 x_3 + R_7 x_2 x_3 = 0,\\
& R_9 x_2 x_4 + 2x_4^2 - 4R x_5 = 0,\\
& x_1\left(x_2+1\right) + R_{10} x_2^2 + x_2 x_3^2 + R_8 x_2 + R_5 x_3^2 + x_4^2 - 1\\
&\quad + R_6 x_3 + R_7 x_2 x_3 + R_9 x_2 x_4 = 0,
\end{aligned} \tag{15}
$$

where

$$
\begin{aligned}
& R = 10, \qquad R_5 = 0.193, \qquad R_6 = \frac{0.002597}{\sqrt{40}},\\
& R_7 = \frac{0.003448}{\sqrt{40}}, \qquad R_8 = \frac{0.00001799}{40},\\
& R_9 = \frac{0.0002155}{\sqrt{40}}, \qquad R_{10} = \frac{0.00003846}{40}.
\end{aligned} \tag{16}
$$

The parameters used by the DEIWO approach are presented in Tables 2 and 3. Some of the solutions obtained by the DEIWO approach for the chemical equilibrium application are depicted in Table 8.

Table 7: Comparison of results for neurophysiology application benchmark.

Algorithm	EA			DEIWO		
	Solutions	Functions values	Error	Solutions	Functions values	Error
Solution 1	−0.8282192996	0.3139636069	0.1207940789	−0.513317629102986	6.476891252397143E − 004	0.0000016131
	0.5446434961	0.1206333341		0.494147884422256	1.764830741770584E − 004	
	−0.0094437659	0.0652332757		0.857821264907119	5.303409416579433E − 004	
	0.7633676230	0.0123681793		−0.869276357234471	5.954489839492686E − 004	
	0.0199325983	0.0465408323		0.035896065698353	3.768837806613808E − 004	
	0.1466452805	0.0330776356		0.035302984144262	6.201719809721604E − 004	
Solution 2	−0.6512719807	0.3607740323	0.1462019196	0.474689101876457	5.397782221636671E − 004	0.0000007971
	−0.6858609598	0.1134596029		−0.473073184862286	6.939643588305122E − 004	
	−0.4637572369	0.0143291397		−0.879846849365024	1.003432819985314E − 004	
	−0.6450853748	0.0412380343		0.880629205401128	4.515693669908784E − 005	
	0.1535909562	0.0204607154		0.021597922714450	8.794875317431894E − 005	
	−0.0036883801	0.0290928705		0.021393481167399	6.561341485605571E − 005	
Solution 3	0.0425943625	0.1489636110	0.1477907295	0.509450781724102	6.698088793095636E − 004	0.0000014868
	−0.1626952821	0.0049729625		0.513500001903592	6.164011496423427E − 005	
	−0.9215324786	0.3332320690		0.860888906816676	4.250176942812017E − 004	
	0.9841530788	0.0038536711		0.858125508396048	5.847818530452624E − 004	
	−0.6789794019	0.1183698936		−0.150516100248646	3.818324226455561E − 004	
	−0.9070329917	0.0224932754		0.151302300254598	6.049508942429435E − 004	
Solution 4	0.3269911198	0.2710537507	0.1340463271	−0.866626598784898	3.340535282903012E − 004	0.0000010566
	0.0266425162	0.0257807695		0.864802076273302	8.506808817729805E − 004	
	0.7886843835	0.1331679003		−0.499291890387587	1.347578366112058E − 005	
	0.9866658030	0.0083976429		0.502959292343981	4.419943585350222E − 004	
	−0.2403284017	0.0421023524		0.028130597707313	3.384546104441034E − 005	
	0.2613854687	0.2008350695		0.027625633057264	1.571583020721235E − 004	
Solution 5	0.8625703877	0.2043564483	0.1151261990	−0.999219226854328	6.382501370955085E − 004	0.0000006850
	−0.7176375053	0.2177684569		−0.999969649944819	4.308368076140212E − 004	
	−0.2271912801	0.0227051752		0.030375755917241	6.212509562937959E − 007	
	0.5169409578	0.1404211857		0.022170611106836	5.350559335882965E − 005	
	−0.1305290129	0.0352067233		0.036181758812701	1.564056883091377E − 005	
	0.1532817352	0.0628718325		−0.036046852107358	2.982004995065353E − 004	
Solution 6	0.7618711576	0.1608754444	0.0596389890	0.855393409271579	8.016421956744679E − 004	0.0000010225
	0.6775336796	0.0901846503		0.865828873424174	4.160424466184143E − 004	
	0.5086028850	0.0169111046		0.518752115726210	1.219573478839314E − 004	
	−0.6713892035	0.1306483985		0.500755833107956	3.030021758793175E − 004	
	0.2563543063	0.0674916558		−0.003420786702082	1.726192288883470E − 004	
	0.0555642759	0.0585551925		0.002831763536266	2.353934715277509E − 004	
Solution 7	0.6609930931	0.0837968094	0.4711746758	0.757633159666954	7.043063529591498E − 004	0.0000013232
	−0.4821043312	0.0055056402		0.737153307517733	1.344150494564378E − 004	
	0.8042915766	0.3405628319		−0.652140850599093	4.842380876615825E − 004	
	−0.8729660781	0.2381471738		−0.675824989376039	5.273519108908182E − 004	
	−0.8987020407	0.3141261109		0.015414698314102	2.245659170263979E − 004	
	−0.1909288931	0.2770687343		−0.015418973671849	1.077840862562709E − 004	
Solution 8	0.2739187054	0.0075566488	0.2152949830	0.643020583613510	5.874692574154850E − 004	0.0000011549
	0.1077541336	0.1344681018		−0.662772222883848	9.581664882496455E − 005	
	0.9656734396	0.3079429049		−0.766232339637761	5.491014584886340E − 005	
	0.9240784300	0.0065004171		0.748885036051881	6.779650279244136E − 004	
	−0.3143660356	0.0831983095		−0.015577794653013	2.724638197823467E − 004	
	−0.0314940456	0.0231155825		−0.016554865087646	5.105665891994952E − 004	
Solution 9	−0.0838634907	0.2102336348	0.1303553523	−0.943535194739874	4.018227192950086E − 004	0.0000014553
	−0.1437222650	0.2606391818		−0.856135906960048	2.566469115463654E − 004	
	0.8847221485	0.0953305060		0.330665259088240	2.922768011053473E − 004	
	0.8477645497	0.0000303505		0.516998990062110	9.610052402960732E − 004	
	0.1777227339	0.0069629557		0.003385954254385	3.374094591558359E − 004	
	−0.0455327341	0.0003085023		−0.003000958247795	1.404462148891126E − 004	

Table 7: Continued.

Algorithm	EA Solutions	EA Functions values	EA Error	DEIWO Solutions	DEIWO Functions values	DEIWO Error
Solution 10	0.4612359064 0.2783584687 0.9360523694 −0.9009125260 −0.1421082154 −0.2759388163	0.0889325997 0.1108731832 0.0852199228 0.0198955321 0.1197727508 0.0090365095	0.0495497431	−0.594047159308489 −0.642823012010247 −0.804411318752719 −0.765987085952347 0.004090936206997 −0.004733324210377	$3.040278002663932E-005$ $4.235938430474651E-005$ $2.090792562180113E-006$ $3.997031908528915E-004$ $2.127225008508227E-004$ $3.369070905144755E-004$	0.0000003212
Solution 11	−0.6907741758 0.8565963646 −0.5428400528 0.5465986672 −0.2327716625 −0.0607828078	0.2281557150 0.0325274349 0.0273082414 0.0385212104 0.0318257635 0.0359158033	0.0583912523	0.818188591099612 −0.819888086423616 −0.574896979975124 0.572533090358964 0.007674059020755 0.007253138354942	$6.089180991386556E-005$ $1.061381536349160E-005$ $9.690837191759410E-005$ $2.057404583100777E-004$ $1.258799064935345E-004$ $1.619105484420356E-004$	0.0000001070
Solution 12	−0.8078668904 −0.9560562726 0.5850998782 −0.2219439027 0.0620152964 −0.0057942792	0.0050092197 0.0366973076 0.0124852708 0.0276342907 0.0168784849 0.0248569233	0.0033499526	0.728516375898463 0.710260686224578 0.684731209007270 0.704073008291695 0.022870698246558 −0.022612099542858	$4.070614592122102E-004$ $1.890434011242448E-004$ $5.496957312890105E-004$ $7.409173419877374E-004$ $1.495429352398985E-004$ $2.800232682831274E-004$	0.0000014555

Example 6 (kinematic application). We consider the kinematic application kin2 as introduced in [17], which describes the inverse position problem for a six-revolute-joint problem in mechanics. The equations describe a denser constraint system and are given as follows:

$$\begin{gathered} x_i^2 + x_{i+1}^2 - 1 = 0, \\ \begin{aligned} a_{1i}x_1x_3 &+ a_{2i}x_1x_4 + a_{3i}x_2x_3 \\ &+ a_{4i}x_2x_4 + a_{5i}x_2x_7 + a_{6i}x_5x_8 \\ &+ a_{7i}x_6x_7 + a_{8i}x_6x_8 + a_{9i}x_1 \\ &+ a_{10i}x_2 + a_{11i}x_3 + a_{12i}x_4 \\ &+ a_{13i}x_5 + a_{14i}x_6 + a_{15i}x_7 \\ &+ a_{16i}x_8 + a_{17i} = 0, \end{aligned} \\ 1 \le i \le 4. \end{gathered} \tag{17}$$

The coefficients, a_{ki}, $1 \le k \le 17$, $1 \le i \le 4$, are given in Table 1. The parameters used by the DEIWO approach are presented in Tables 2 and 3. Some of the solutions obtained by the DEIWO approach for the kinematic example kin2 are presented in Table 9.

Example 7 (combustion application). We consider the combustion problem for a temperature of 3000°C as proposed in [18]. The problem is described by the following sparse system of equations:

$$\begin{gathered} x_2 + 2x_6 + x_9 + 2x_{10} = 10^{-5}, \\ x_3 + x_8 = 3 \times 10^{-5}, \\ x_1 + x_3 + 2x_5 + 2x_8 + x_9 + x_{10} = 5 \times 10^{-5}, \\ x_4 + 2x_7 = 10^{-5}, \\ 0.5140437 \times 10^{-7} x_5 = x_1^2, \\ 0.1006932 \times 10^{-6} x_6 = 2x_2^2, \\ 0.7816278 \times 10^{-15} x_7 = x_4^2, \\ 0.1496236 \times 10^{-6} x_8 = x_1x_3, \\ 0.619441 \times 10^{-7} x_9 = x_1x_2, \\ 0.2089296 \times 10^{-14} x_{10} = x_1x_2^2. \end{gathered} \tag{18}$$

The parameters used by the DEIWO approach are presented in Tables 2 and 3. Some of the solutions obtained by the DEIWO approach are presented in Table 10.

Example 8 (economics modeling application). The following modeling problem is considered as difficult and can be scaled up to arbitrary dimensions [18]. The problem is given by the following system of equations:

$$\begin{gathered} \left(x_k + \sum_{i=1}^{n-k-1} x_i x_{i+k} \right) x_n - c_k = 0, \quad 1 \le k \le n-1, \\ \sum_{l=1}^{n-1} x_l + 1 = 0. \end{gathered} \tag{19}$$

The constant c_k can be randomly chosen. We considered the value 0 for the constants in our experiments and the case of 20 equations as in literature [12].

The parameters used by the DEIWO approach are presented in Tables 2 and 3. Some of the solutions obtained by the DEIWO approach are presented in Table 11.

Table 2 gives the parameters values used by the DEIWO.

Table 8: Comparison of results for chemical equilibrium application benchmark.

Algorithm	EA			DEIWO		
	Solutions	Functions values	Error	Solutions	Functions	Error
	−0.0163087455	0.1525772447		0.047039050817101	0.026769403348726	
	0.2613604709	0.3712483549		1.869415636872731	0.006747169898629	
Solution 1	0.5981559224	0.0265535280	0.2537179573	−0.265064210870296	$8.805250733959802E-004$	0.0007637704
	0.8606983883	0.2784694066		−0.849394548238717	$1.522979575456773E-004$	
	0.0440020125	0.1168649339		0.036068394869839	$9.201328624012376E-004$	
	0.3357311285	0.3491354953		0.026474719105915	0.009748439199621	
	0.1015972384	0.3388481591		3.527094917732416	0.013690244877239	
Solution 2	0.1959807715	0.0324919199	0.4345218367	0.199080066736952	0.001726981982017	0.0003118371
	0.5298149584	0.2853430985		−0.855305091111645	0.005077554838563	
	0.0069016628	0.3380473798		0.036701709037720	$7.862731788029286E-004$	
	0.3273318676	0.3206895328		0.043701369103126	0.024510445238943	
	0.0396552907	0.2986524101		2.049069927756282	0.008581129313918	
Solution 3	0.5208586308	0.0741335715	0.3799218328	0.252703525295020	0.003229156423991	0.0007169003
	−0.4442729860	0.1331193703		−0.852671096585522	0.004195330599570	
	0.0065409250	0.3989667326		0.036246028365058	0.003804521422204	
	0.1626252165	0.0298611415		0.041608285134930	0.018067590257058	
	−0.3017126041	0.2240794988		2.045381329230565	0.015129175003949	
Solution 4	−0.1783889066	0.2301576033	0.3113039788	0.256879114658442	0.006078177707939	0.0006.929642
	0.7137275251	0.0971666758		−0.848553325138083	0.008580373137608	
	0.0278993324	0.3805446921		0.036215168144720	0.005202632121528	
	0.3221051215	0.3510711865		0.041994672912187	0.020960255065471	
	0.2525051386	0.3544681189		2.105373855163727	0.009880976362258	
Solution 5	−0.4189363818	0.0165037731	0.2740228596	−0.254482330009030	0.005430820903461	0.0007143232
	−0.6331604264	0.1035503586		0.850700424653915	0.011875271190618	
	0.0174557111	0.1177149436		0.036482968084062	0.002615595117843	
	0.0429158354	0.0446360606		0.039142614022070	0.020928314409356	
	0.0811055468	0.2478304056		2.315919558093960	0.006812894119697	
Solution 6	−0.2662203512	0.2040211938	0.2625846864	−0.242200440749196	0.003638624072727	0.0005034014
	−0.7711670069	0.0243731377		0.851329829612683	0.001947129057743	
	0.0303442027	0.3396013173		0.036288481660450	0.001400229470722	
	0.7276812579	0.3738656386		0.029777144983889	0.012756030203747	
	−0.4551167619	0.0304080605		3.130317377778047	0.013574181713828	
Solution 7	−0.2113909448	0.0838221369	0.3273224789	−0.210038553008973	0.001175210980581	0.0003990879
	−0.4895999565	0.1776147414		−0.855206644040595	0.007108131624504	
	0.0075452271	0.3755303113		0.036744343061275	$4.544958713665148E-004$	
	0.1296088399	0.0660183067		0.034568611394593	0.018414022774285	
	0.2275206857	0.1191351931		2.701986140890615	0.005877945460186	
Solution 8	−0.1061140447	0.2387977164	0.1698830307	0.225725393195049	0.003279597720941	0.0004157540
	−0.8124975178	0.0792417966		−0.853903291996822	0.002556918408524	
	0.0310264084	0.1760779901		0.036519499172778	0.004983364513138	
	0.3367208030	0.2796624109		0.029700290568578	0.017030072621489	
	0.1420207287	0.0856442848		3.233972423385453	$8.550186286035441E-004$	
Solution 9	−0.1427721429	0.2660909007	0.2383566164	0.207655335416109	0.006078174688298	0.0003695853
	−0.8435618534	0.0247927741		−0.849852570852197	0.005196711511919	
	0.0349599086	0.1028940828		0.036240046204135	0.003857570695716	
	0.4224008806	0.2712921597		0.039076419522626	0.014604592480938	
	−0.1889079092	0.0011063024		2.195407554156290	0.023351748126232	
Solution 10	0.3561384679	0.1890226787	0.1772971294	−0.246825556242421	0.003407541582840	0.0007996812
	−0.7390069308	0.1414158627		0.857523650896189	$6.178556688816173E-004$	
	0.0237712845	0.1106294577		0.036753497883680	0.005393500809564	
	0.2395706253	0.2975483261		0.029989577399948	0.011074163193089	
	0.4637567881	0.2847994096		3.042439401620619	0.016575368131918	
Solution 11	0.0154833612	0.1413409742	0.3034471293	0.212865462882654	$1.046629449772713E-004$	0.0004476638
	−0.5861394961	0.02122094995		−0.854930903089161	0.007033411425349	
	0.0177082676	0.3056084889		0.036718962042137	$8.967043275154037E-004$	

Table 8: Continued.

Algorithm	EA Solutions	EA Functions values	EA Error	DEIWO Solutions	DEIWO Functions	DEIWO Error
Solution 12	0.1671662105 0.2358035522 −0.1121274369 −0.7953186041 0.0273318802	0.1245889560 0.0243717563 0.2079331799 0.1717817639 0.1555592488	0.1562964598	0.026264550497174 3.679702397787328 0.193832890273980 0.850241727008394 0.036012370926070	0.014873167160221 0.001793762023353 0.003373540589790 0.005433755376894 0.008086975223007	0.0003307343
Solution 13	0.4586478321 −0.1011456067 −0.0115635220 −0.75891034603 0.02454099949	0.3386346204 0.1204471133 0.1963075419 0.1702524624 0.0117867162	0.2353790429	0.039763793276441 2.261109646335777 0.245179433647380 0.851709222649360 0.036466833083980	0.020273590576762 0.008783261436782 0.003715808978656 0.007790504369955 0.003080933278725	0.0005721555
Solution 14	0.4064810686 0.0073246701 0.0846953560 0.6726843741 0.0234023812	0.3392512646 0.1784649874 0.1843099545 0.0310865463 0.1365650475	0.2344978766	0.042494575252538 2.060377057920538 −0.253929806014199 0.846954259610168 0.035785075844889	0.022694195654279 0.007609342603815 0.003926958981918 0.003319461816077 0.007645558185311	0.0006578230

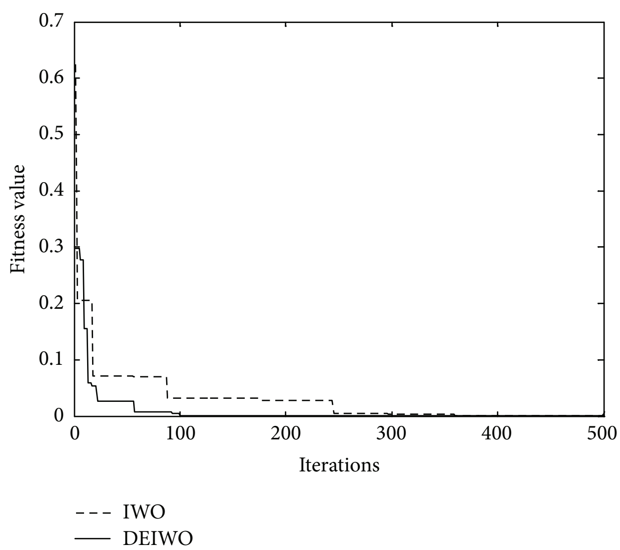

Figure 1: Results of 500 iterations.

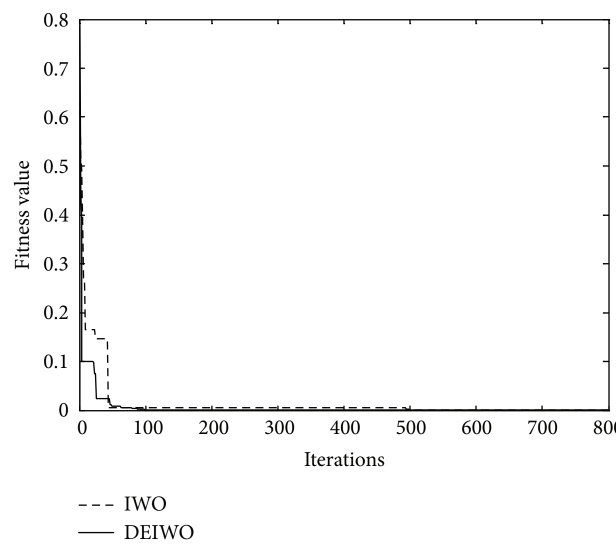

Figure 2: Results of 800 iterations.

5.3. Experimental Results and Discussion. For Example 1, we can see from Table 4 that the error of DEIWO is very small. When iteration number is 500, the error of DEIWO is E-020, compared to the traditional numerical method has high precision 16, compared to 14 EA high accuracy, precision is higher than IWO 12. What is more, all the accuracy for functions values is very high simultaneously. Thus, the solutions acquired can be approximately equal to the theoretical value. Figures 1 and 2 show that the convergence speed of DEIWO is faster than that of IWO, and the optimization precision higher than that of IWO. For Example 2, Table 5 shows the error of DEIWO is E-024, compared with traditional methods, and high precision of 20, 18 accuracy is higher than the EA, and precision is higher than IWO 16; almost equal to 0 when iteration number is 500. When iteration number reaches 800, the error is 0. Figures 3 and 4 show that IWO easy to fall into local optimum, while DEIWO can effectively jump out of local optimal solution. In addition, DEIWO can effectively jump out of local optimal, fewer than 200 generations has been largely convergence. All of this indicates that DEIWO is more effective than traditional numerical method and IWO.

For interval arithmetic benchmark, we can see the errors of all solutions obtained by DEIWO are almost in E-005. Moreover, the given 8 solutions are stable. Phenomenon of fluctuation does not appear. Thus, DEIWO can effectively resolve high-dimensional equation systems.

For neurophysiology application benchmark, 12 solutions are given in Table 7. The solutions' errors of DEIWO are between E-006 and E-007, EA solving accuracy in E-001 and E-003. This also indicates that DEIWO is more effective than EA.

Table 8 shows 14 solutions of DEIWO and EA for chemical equilibrium application. The errors of DEIWO are all E-004, while those of EA are E-001.

Table 9: Comparison of results for kinematic application benchmark.

Algorithm	EA			DEIWO		
	Solutions	Functions values	Error	Solutions	Functions values	Error
Solution 1	−0.0625820337	0.3911967824	2.8055186567	0.045507371706493	0.001754595515567	0.0001004601
	0.7777446281	0.3925758963		0.998085408972999	0.003231810708989	
	−0.0503725828	0.8526542737		0.024366076545306	0.001642317536650	
	0.3805368959	0.5424213097		−0.998881362713880	0.002169368838492	
	−0.5592587603	0.7742116224		−0.008164213640613	0.004282027104326	
	−0.6988338865	0.1537105718		−0.773486615357711	0.001214133281959	
	0.3963927675	0.9116019977		0.001737164451486	0.003982242223259	
	0.0861763643	0.1519175234		−1.189751083986727	0.006623094167720	
Solution 2	−0.1564353525	0.7723864643	1.8197999485	−0.209908402344501	0.000822682189556	0.0001426091
	0.4507122320	0.5832167209		0.978141679315801	0.001868888002580	
	0.4622139796	0.0087255337		0.203396084482184	0.004862472339110	
	−0.8818348503	0.2031050697		0.981576540651021	0.001672810091532	
	−0.6522824284	0.6056929403		−0.195397812001838	0.006669856089477	
	0.4082826235	0.3663682493		0.225356569439925	0.004274595104825	
	0.4718261386	0.3532359802		−0.346358155642600	0.003392827053353	
	−0.5070478474	0.4646334692		−1.054154642146118	0.006142247670554	
Solution 3	−0.5618177814	0.0336943748	1.7384318314	−0.202282750875349	0.002202615985165	0.0001181619
	0.8473813517	0.2323541264		0.980451072049731	0.003098994555294	
	0.2226897354	0.9432510244		0.204486405102704	0.004087922439548	
	−0.0846064846	0.3663913629		0.980955265324429	0.002656303324497	
	−0.7914861841	0.0081327333		−0.200955394943197	0.007030249990859	
	−0.4111014166	0.2206688453		0.226603999519539	0.000351392074988	
	0.3056098314	0.624967041		−0.346215357036744	0.002940170368879	
	0.4290046184	0.4690397335		−1.055887922168507	0.004663375717434	
Solution 4	0.0608363294	0.7409305497	2.3539965598	0.089422621197712	0.003572640293149	0.0000791293
	0.5053398770	0.6356638946		0.997785665917926	0.000259082400491	
	0.3301025811	0.7726033233		0.068431332656990	0.000907927865151	
	−0.3441350935	0.8133740699		−0.997200694366805	0.004354292165639	
	−0.2611454909	0.0176109130		0.035163660058698	0.003188494681972	
	−0.1335439441	0.0929407653		−0.842053211908446	0.001119794486023	
	0.7518856650	0.2565663633		−0.019839283580881	0.003566152462141	
	−0.3711959678	0.260158013		−1.151839767619862	0.004730361789871	
Solution 5	−0.7461742647	0.1552859413	2.3132511111	0.005618623394285	0.012038481762346	0.0001565654
	0.5365985698	0.4712662401		0.993946653150363	0.001991177524501	
	0.4907094198	0.7461140527		0.118580049821602	0.001098498042427	
	0.1144124666	0.6917033188		−0.993497493618748	0.000219260012879	
	−0.5433290610	0.9909917470		−0.114812848523787	0.001036835150341	
	0.3443841680	0.0271473496		−0.699491795013951	0.001125049094542	
	0.6758924483	0.0900355495		0.036947466879746	0.000179375216701	
	−0.3341809018	0.20247922410		−1.208913136968972	0.002011941060148	
Solution 6	−0.3064809352	0.9060378884	3.8308029184	0.077779987853934	0.000692862137470	0.0000905605
	−0.0056167467	0.7492384561		0.997317971174144	0.002111917197364	
	−0.5007294639	0.7403485971		0.056964437816277	0.000121608449425	
	−0.0944531990	0.4782413752		−0.998437109323217	0.001744806752775	
	−0.7161265376	0.8882665877		0.037128587012228	0.001103067513220	
	0.6371096100	0.2081000785		−0.830589842164460	0.003814260073128	
	0.4792262814	0.7690562864		−0.026418733632420	0.005523418787933	
	−0.4680908930	0.4979341279		−1.159254624161276	0.006023917232275	
Solution 7	−0.5739275815	0.5286935399	1.9576485695	0.062451694147746	0.002479214228837	0.0000526779
	0.3767142036	0.8412047222		0.999289247478883	0.002927150217360	
	0.1299295442	0.3390733979		−0.065940504171922	0.001665746962001	
	−0.8025240903	0.2643422170		−0.998657897816642	0.001658680908898	
	−0.3026761756	0.5162506493		0.031995659386056	0.000472802926307	
	0.1226779594	0.2659449070		−0.777915484740598	0.000768554472374	
	0.7855891471	0.4029986127		0.011189710814883	0.004544515610755	
	−0.1240877403	0.5347985046		−1.188813754784150	0.003312159443232	

Table 9: Continued.

Algorithm	EA Solutions	EA Functions values	EA Error	DEIWO Solutions	DEIWO Functions values	DEIWO Error
Solution 8	0.7822939914	0.1443884130	0.6032920001	0.129736374626430	0.000932222022973	0.0000690265
	−0.4935865657	0.1116189082		0.991078327417070	0.004618875277069	
	0.8029653752	0.3487762429		0.114651095272193	0.000203409138437	
	0.0804385674	0.1428116670		−0.993508195985984	0.003973280158754	
	0.9223437373	0.3802921343		0.094700498150336	0.001204313453836	
	0.0296919251	0.5311414488		−0.914637593030690	0.001429540419579	
	0.7980078255	0.0338066582		−0.052275570550158	0.003053228353648	
	−0.8324923146	0.0083023855		−1.115505051433812	0.004263676032891	
Solution 9	−0.7461742647	0.0762392460	1.3773369662	0.028424442196598	0.004476461376883	0.0000566727
	0.6057926381	0.2265885749		0.997354294976930	0.002834673688489	
	0.9271482376	0.0694226055		0.049494813895825	0.000769440136329	
	0.2664085959	0.6991957509		−0.999159498545526	0.000760152784294	
	−0.4794066217	0.0290725466		−0.049400903337364	0.000030652634633	
	−0.3302191010	0.1921690687		−0.740573888955816	0.001888111028099	
	0.7692157072	0.4459273853		0.016727719945304	0.003940336679685	
	−0.1642217317	0.7592540224		−1.200998766591111	0.002887323459874	
Solution 10	−0.6205399028	0.1461818771	2.5250914998	0.270276389119342	0.007641330839746	0.0001467558
	0.6846519932	0.4555568728		0.966742987729601	0.002119116346448	
	−0.9933817672	0.0102591212		0.251572811188323	0.001683133324140	
	−0.0541621937	0.9914691316		−0.968707517259438	0.006884785205684	
	−0.0748152730	0.2920414356		0.233925117931725	0.002237943072133	
	0.37913950333	0.5909270772		−1.122930726803439	0.002719402992536	
	−0.4826393335	0.8366810215		−0.122906457137246	0.004568280905804	
	−0.7830381952	0.7918662138		−1.014959836413780	0.000607655156908	

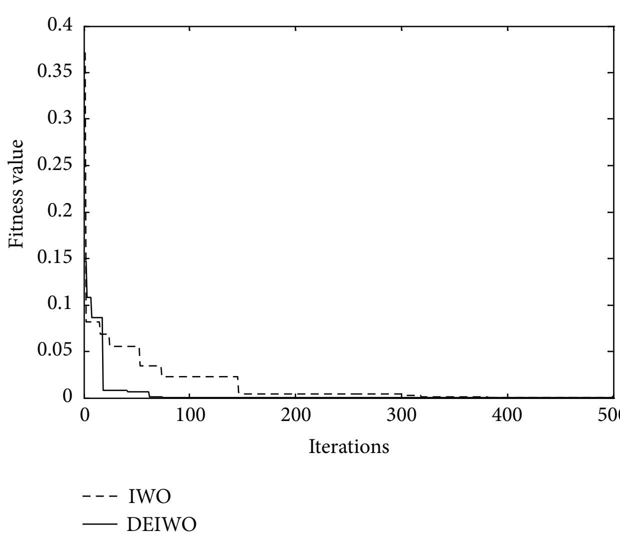

Figure 3: Results of 500 iterations.

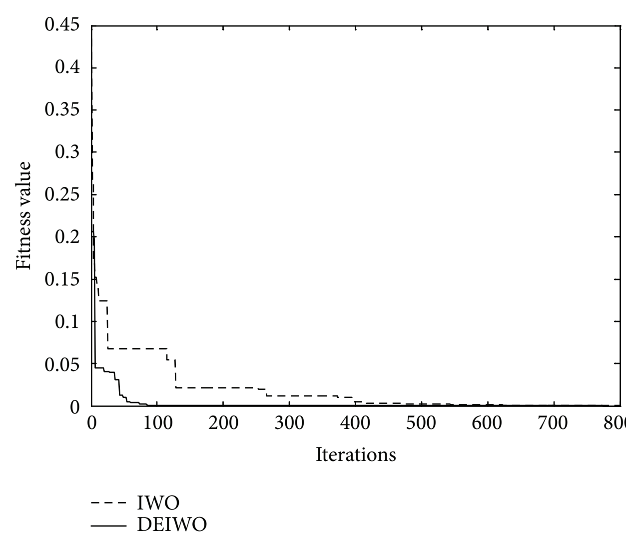

Figure 4: Results of 800 iterations.

For kinematic application benchmark, we can see that the errors of EA are very large, such as Solution 6; the error is 3.8, while the errors of DEIWO are between E-004 and E-005. Solutions for combustion application are given in Table 10. It is obvious that DEIWO can effectively resolve the nonlinear equation systems, and the error is much smaller than that of EA.

The number of equations for economics modeling application is variable. In our experiment, the number is 20. As can be seen from the Table 11, even if the equations of variable number is 20, DEIWO by solving the error precision can achieve the fitness of E-006, and even E-007. Than the error of high precision of the solutions of EA for five orders of magnitude. At the same time, precision of functions values obtained by DEIWO is between E-004 and E-006. No functions values are too large or too small.

The running time required for our algorithm to converge is presented in Table 12. It is measured in seconds using

Table 10: Comparison of results for combustion application benchmark.

Algorithm	EA			DEIWO		
	Solutions	Functions values	Error	Solutions	Functions values	Error
Solution 1	−0.0552429896	0.0274133878	0.0371466690	0.003672929585885	0.002043691593620	0.0000304137
	−0.0023377533	0.0841848522		0.025839710713681	0.004159565646108	
	0.0455880930	0.1482418893		0.521599141528195	0.000758296565400	
	−0.1287029472	0.0839188567		0.004984881426735	0.001702002448575	
	0.0539771728	0.0030517851		0.628058027576599	0.000013458126816	
	−0.0151036079	0.0000109317		1.337360448010958	0.001335246636430	
	0.1063159019	0.0165644486		−0.001636439489080	0.000024849042839	
	0.0386267592	0.0025184283		−0.525728707174303	0.001915875580313	
	−0.1144905135	0.0001291515		1.237228898173834	0.000094830798928	
	0.0872294353	0.0000003019		−1.967867906657906	0.000002452380746	
Solution 2	−0.0338378558	0.0008794626	0.0794603736	0.000170916648864	0.000847476006004	0.0000044657
	0.0185669333	0.1035086837		−0.007197346940591	0.001240932558196	
	0.0534924988	0.0955626197		−0.063594314222215	0.001348023865310	
	−0.0392784517	0.2441423777		−0.012650838988592	0.000594884188514	
	0.0183882247	0.0011449995		1.737695584563462	0.000000060112646	
	0.0005245892	0.0006894619		−0.347952826255259	0.000103638642450	
	−0.1024269629	0.0015427967		0.006032977400039	0.000160043727115	
	0.0500461848	0.0018100789		0.064865246780411	0.000010859621702	
	−0.1013361102	0.0006282589		−7.783877291359830	0.000001712311771	
	0.0404252678	0.0000116649		4.243918883408473	0.000000008853799	
Solution 3	−0.0103209333	0.1170689295	0.0601546074	0.012819319246173	0.001067920987079	0.0000247769
	0.0021201108	0.0726549501		−0.019071587101257	0.000826374747992	
	0.1207182825	0.2023013696		0.081305679904288	0.002833219569578	
	−0.0263026679	0.0155834694		0.029533158589909	0.003530328678861	
	0.0044219824	0.0001065214		0.443752747560672	0.000164312135105	
	−0.0850838579	0.0000089983		−1.815950119305152	0.000727633722950	
	0.0053645992	0.0006918303		−0.016526743634385	0.000872207456297	
	−0.0480333324	0.0012459181		−0.082102054652280	0.001042295751625	
	0.0732269061	0.0000218860		−5.292468955772288	0.000244156926303	
	0.1059498141	0.0000000463		4.472259351235464	0.000004662712454	
Solution 4	0.0177198747	0.0262522791	0.1058284540	0.015469554753745	0.003486775086451	0.0000195602
	0.0030100424	0.0644480331		−0.027948524788676	0.000509018636612	
	0.0676669725	0.1379528266		−0.031112943145748	0.001919773451644	
	−0.0408039903	0.286269853		0.000991240275462	0.000734539635309	
	0.0852565598	0.0003139895		−3.984894922917517	0.000239511965292	
	0.0536056660	0.0000181153		−4.439885695760647	0.001562687142025	
	0.1635419218	0.0016649656		−0.000857889955385	0.000000982557284	
	−0.0031889394	0.0011990507		0.030633924509136	0.000481307961101	
	−0.1390794276	0.0000533461		6.948027176097301	0.000432781623864	
	0.0275601661	0.0000001605		0.978107982563108	0.000012083579193	
Solution 5	0.0348357700	0.0289865877	0.0768661970	−0.001269492482752	0.000878131789857	0.0000162755
	0.1092386108	0.2167589209		0.013623874571987	0.002563519769053	
	0.1250085306	0.0089588165		−1.175677942247783	0.001329398835772	
	−0.0107958218	0.1683941428		−0.030597939250755	0.001980751318993	
	−0.1399310251	0.0012135380		−2.528288495278615	0.000001741576241	
	−0.0177642059	0.0238661499		−4.876718138697484	0.000371710969061	
	−0.0787941605	0.0001165497		0.016294345284874	0.000936233886393	
	0.0917803902	0.0043547546		1.173144422478730	0.001492338779729	
	−0.0422169082	0.0038054137		−1.968788171623771	0.000017173451524	
	−0.0302349392	0.0004156978		5.854744353118305	0.000000235630459	
Solution 6	0.0172348545	0.0202656999	0.0042879841	−0.001858395184513	0.001820431874319	0.0000300866
	−0.0049839785	0.0113645412		0.023721236556167	0.004719151003933	
	−0.0036835674	0.0052992119		0.653216998921955	0.000379406194652	
	−0.0401647761	0.0609700462		−0.018603147142958	0.001223765779024	
	0.0334303826	0.0002970384		3.776633185434082	0.000003259497212	
	−0.0049589041	0.0000496805		4.088647518759656	0.001124982428505	
	0.0505724112	0.0016132092		0.008694690681967	0.000346077083623	
	−0.0076509738	0.0000634846		−0.657906149925889	0.001213836886952	
	−0.1141678724	0.0000858910		−5.577580694917199	0.000043737933515	
	0.0544069796	0.0000004281		−1.310802573641980	0.000001045713511	

Table 10: Continued.

Algorithm	EA Solutions	EA Functions values	EA Error	DEIWO Solutions	DEIWO Functions values	DEIWO Error
Solution 7	−0.1443475355	0.0460704291	0.0134575843	−0.002441213851907	0.001484619742542	0.0000291561
	0.0137124749	0.0928317803		−0.007212781373596	0.003755227488414	
	0.0532523778	0.0470659155		−0.632390997817911	0.001910664271513	
	−0.0407593315	0.0014507164		0.021860340733110	0.002564589021807	
	0.0053340672	0.0208362107		−1.135708922430422	0.000006017905472	
	0.0390841261	0.0003760600		3.176958936133236	0.000103728532125	
	0.0196593075	0.0016613231		−0.009642875855651	0.000477874496968	
	0.0396094025	0.0076868554		0.636176225306324	0.001543706476717	
	0.0463257617	0.0019793648		9.619896251014470	0.000017012045888	
	−0.0921334590	0.0000271419		−7.984037980824944	0.000000127002218	
Solution 8	0.1612054472	0.0966899590	0.0231646270	−0.013880682509557	0.001750147547270	0.0000254721
	0.1001108591	0.0844303598		0.015121949074709	0.000131319731513	
	−0.0303525758	0.0520483724		−0.275566456809759	0.000908744309037	
	0.0015541591	0.0511277742		−0.040530104159630	0.001986519079324	
	0.0464169709	0.0259871938		−1.600723894610612	0.000192755631134	
	0.0906816701	0.0200443591		−2.981776673353555	0.000457646932271	
	−0.0263359667	0.0000024154		0.021263311619477	0.001642689343190	
	−0.0540477839	0.0048929924		0.275465137078247	0.003825009281174	
	0.0577884947	0.0161384122		−0.071979725029653	0.000209898515312	
	−0.121281367	0.0016156306		3.011085635104662	0.000003174142090	

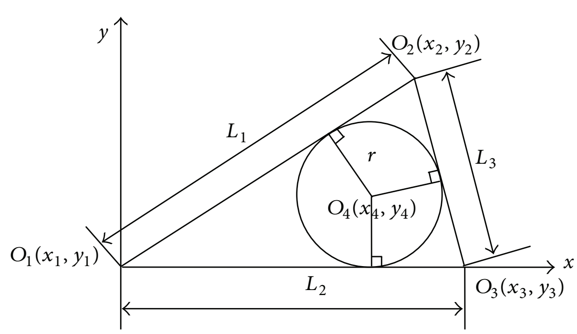

Figure 5: Single circle packing problem model.

processor: CPU Intel Core i3-370 using frequency: 2.40 GHz, memory: 4 GB, operating system: Windows 7, run software: MATLAB 7.6. The comparison of the number of function evaluations required by the DEIWO for all the examples is shown in Table 13.

5.4. A Practical Example for Geometric Constraint. The geometric constraint problem is generally divided into three categories: integrity constraint, underconstraint, and overconstraint [19]. Solving geometric constraint problems generally uses the divide and conquer method; the problem is decomposed into a number of small problems, we use numerical methods to solve. This paper uses the example in literature [20]: a circle packing problem. First, transform the geometric constraints into equations, and then the equations are transformed into an optimization problem; finally solve optimization equation. In this way, we do not need to consider if the number of variables and the number of equations are equal; thus, this problem can be effectively solved.

Single circle packing problem model is as shown in Figure 5, where, the Point O_1 at the origin of coordinates, the line segment O_1O_3 at the x-axis of positive direction, the length of line segment O_1O_2 is L_1, the length of line segment O_1O_3 is L_2, and the length of line segment O_2O_3 is L_3. There is an inscribed circle in triangle $O_1O_2O_3$, center is O_4, and radius is r. The geometric constraint problem can be transformed and simplified, can obtain the following equations:

$$\begin{gathered} f_1 = x_2^2 + y_2^2 - L_1^2 = 0, \\ f_2 = \left(L_2 - x_2\right)^2 + y_2^2 - L_3^2 = 0, \\ f_3 = \frac{\left(y_2 x_4 - x_2 r\right)^2}{\left(x_2^2 + y_2^2\right)} - r^2 = 0, \\ f_4 = \frac{\left(\left(-y_2\right)\left(x_4 - x_2\right) + \left(L_2 - x_2\right)\left(y_2 - r\right)\right)^2}{\left(\left(L_2 - x_2\right)^2 + y_2^2\right)} - r^2 = 0. \end{gathered} \tag{20}$$

Solve the equations, and then get the coordinate values of points $O_1(x_1, y_1)$, $O_2(x_2, y_2)$, $O_3(x_3, y_3)$, $O_4(x_4, y_4)$, and r. Parameters are set as $G\_SIZE = 100$, $P\_MAX = 150$, $F = 1.06$, and $CR = 0.1$. The remaining parameters are the same as in Table 2. Unknown quantity needed to be solved is x_2, y_2, x_4, r, with range is $(0, 40)$. Input parameters are L_1, L_2, and L_3. For the first set of parameters, the fitness function value obtained by DEIWO is 1 precision higher than that of PSO. For the second set of parameters, accuracy is the same. From the third and the fourth set of parameters, we can see that there is conflict. The results of solving geometric constraint are shown in Table 14.

The DEIWO algorithm is very efficient for solving equations systems. The algorithm has the abilities to overcome local optimal solutions and to obtain global optimal solutions.

Table 11: Comparison of results for economics modeling application benchmark.

Algorithm	EA			DEIWO		
	Solutions	Functions values	Error	Solutions	Functions values	Error
Solution 1	−0.1639324	$1.94E-005$	0.4094911811	−6.219110636449961	0.000206621288237	0.0000004629
	−0.3813209	$9.73E-005$		−2.286125269050309	0.000055179798466	
	0.2242448	0.0001201		−0.190906907089769	0.000179369758855	
	−0.0755094	$2.39E-005$		3.293847780216458	0.000153340171517	
	0.1171098	$5.61E-005$		0.226175997267840	0.000012764866830	
	0.0174083	$3.89E-005$		6.431110110640911	0.000366739260435	
	−0.0594358	$3.90E-005$		0.873513075442108	0.000027744616667	
	−0.2218284	$9.31E-005$		−1.765644489553301	0.000025655636832	
	0.1856304	0.0001294		−6.288552449032001	0.000211906268785	
	−0.2653962	$5.01E-005$		−1.366427157490679	0.000005300945657	
	−0.3712114	0.0001009		3.124344370134706	0.000160285427757	
	−0.3440810	0.0001601		−7.620830976635871	0.000008959715481	
	−0.1060168	$6.32E-006$		3.649665491921485	0.000036544006572	
	0.0218564	$7.96E-006$		−4.964622122726966	0.000213079172405	
	−0.2028748	$7.66E-006$		8.520732370320570	0.000028339070859	
	0.0533728	$2.35E-005$		4.694681906366633	0.000079961116369	
	−0.0587111	$2.21E-005$		−3.567582193815149	0.000045923286818	
	0.0057098	$3.34E-006$		1.165936019206677	0.000022862241718	
	−0.0149290	$6.12E-006$		1.289481831200988	0.000004301520459	
	−0.0004102	0.4110599		0.000003335851933	0.000313249125625	
Solution 2	−0.2071340	0.0050496	0.1866532809	−8.258178201520460	0.000112448421209	0.0000023157
	−0.2251718	0.0003158		3.820226326345229	0.000508863338639	
	−0.0910972	0.0035983		0.896792437864452	0.000443065499734	
	−0.0028412	0.0044357		−1.378246977599881	0.000118951963880	
	−0.2110337	0.0117217		−1.977980138918008	0.000139208990183	
	0.4501557	0.0153466		9.317994920873035	0.000243214789175	
	−0.0263800	0.0005013		7.625886190069895	0.000139948558457	
	0.0086212	0.0001027		−9.854325674137856	0.000253303825953	
	−0.2065700	0.0076698		−8.451314609323713	0.000004024795816	
	0.1663536	0.0045637		−1.288302214965556	0.000387682069229	
	−0.1450036	0.0009652		−4.806635040632184	0.000182576533214	
	−0.0743482	$8.36E-006$		8.714568113773245	0.000073658228323	
	−0.2007066	0.0018678		−0.240230209985321	0.000016739957752	
	−0.1451752	0.0016314		−2.242967577834238	0.000163845735074	
	−0.2078702	0.0044618		−3.026321149977592	0.000222130681599	
	−0.2750080	0.0084773		7.848935266531751	0.000103377255053	
	−0.0422618	0.0014519		−0.856054140099465	0.000140635835620	
	−0.0602186	0.0024115		4.362368405116176	0.000048579832185	
	0.0643765	0.0021106		−1.207387470433794	0.000004092228149	
	−0.0327867	0.2107177		−0.000003389324678	0.001171744854285	
Solution 3	0.0936580	0.0004909	0.2694715119	0.410175655605525	0.000174067214160	0.0000057950
	−0.1113572	$1.66e-005$		−2.945267796294842	0.000070546833656	
	−0.0652960	0.0001413		−3.835549781983572	0.000015906476143	
	−0.0274100	0.0001387		−2.443847819769871	0.000141821985185	
	−0.0515078	$1.50e-005$		−5.748058015027992	0.000226460376758	
	−0.0525712	0.0001687		−1.590111925088804	0.000111802212012	
	0.1674281	0.0006289		2.813985490565141	0.000123909026492	
	−0.0284058	$7.11e-006$		−4.171269624265078	0.000270653873401	
	−0.1587341	0.0002428		−0.630245452070986	0.000001910012918	
	−0.1284569	0.0001981		8.113507792718970	0.000074219143303	
	−0.1326800	0.0002817		4.439827399023531	0.000015270345745	
	−0.1138290	0.0002442		4.827513156671341	0.000060127527268	
	−0.1430544	0.0001696		7.390298377478004	0.000025057749856	
	0.0521726	0.0001335		−8.776390912646111	0.000081474905001	
	−0.2608338	0.0005227		−3.473157113136982	0.000034111315945	
	−0.1602811	0.0003038		0.231251460267028	0.000007504624843	
	−0.1141750	0.0002455		5.048928765222839	0.000000658443143	
	−0.1677992	0.0003751		−2.003672367032949	0.000003576887608	
	−0.1159721	0.0002454		1.344442609750887	0.000003311438718	
	0.0020995	0.3975329		−0.000002463056953	0.002359899986076	

Table 11: Continued.

Algorithm	EA Solutions	EA Functions values	EA Error	DEIWO Solutions	DEIWO Functions values	DEIWO Error
Solution 4	−0.2686292	0.0015673	0.0336841112	0.309460292219735	0.000526241325241	0.0000017369
	0.3391340	0.0028117		−9.464274256657685	0.000697399646506	
	0.0732562	$2.58e-005$		3.672220916230188	0.000524135889717	
	0.0797120	0.0013986		3.501423117721803	0.000291514335086	
	−0.1109362	0.0003991		4.263588177101166	0.000511322143838	
	0.0177894	0.0014187		3.919464724708721	0.000111237908905	
	0.4220681	0.0023095		5.299860346560947	0.000002321170209	
	−0.0583526	$4.67e-006$		4.570456826901647	0.000357073934313	
	−0.2610232	0.0017655		−9.419089944307890	0.000119999377139	
	−0.2838340	0.0014891		4.270196782723423	0.000161166520658	
	−0.3579828	0.0028633		−9.794235483541998	0.000247081697216	
	0.0214270	0.0014185		3.575029940105042	0.000079293991687	
	−0.6282558	0.0035229		−9.670595944106060	0.000272037260975	
	0.2185146	0.0010139		4.067150477104734	0.000003123855339	
	−0.0897853	0.0008279		−9.508380909500426	0.000049007635690	
	−0.0178795	0.0001595		2.638382397598715	0.000037946306022	
	−0.2514783	0.0010944		2.824986452927806	0.000020643833165	
	−0.0546466	0.0002984		2.617827805414225	0.000007010036976	
	0.0275084	0.0001323		1.326365492565140	0.000003070341910	
	0.0048114	0.2261284		−0.000002314853582	0.000162788230767	

Table 12: Comparison of results for CPU time required by the DEIWO for all the examples.

	Example 1	Example 2	Example 3	Example 4	Example 5	Example 6	Example 7	Example 8
EA [12]	5.14	5.09	39.07	28.90	32.71	221.09	151.12	640.92
PDS [9]	5.00	5.00	30	30	30	30	30	20
DEIWO	2.27	2.27	15.05	13.11	12.25	16.05	17.25	18.14

Table 13: Comparison of the number of function evaluations required by the DEIWO for all the examples.

	Example 1	Example 2	Example 3	Example 4	Example 5	Example 6	Example 7	Example 8
EA [12]	1	1	8	12	14	10	8	4
PDS [9]	1	1	1	2	2	2	2	3
DEIWO	2	2	8	12	14	10	8	4

Table 14: Results of solving geometric constraint.

L_1, L_2, L_3	Algorithm	x_2	y_2	x_4	r	$F(x)$
20, 20, 20	PSO	10.0000	17.3205	10.0000	5.7735	$1.446E-5$
	DEIWO	10.0000000215	17.3205081063	10.0000000126	5.7735026797	$1.4937E-6$
10, 20, 15	PSO	6.8750	7.2618	7.5000	3.2275	$9.482E-6$
	DEIWO	6.8750000954	7.2618437417	7.5000003760	3.2274862446	$6.4021E-6$
13.6, 8.25, 16.3	PSO	−0.7677	13.5783	2.7750	2.9363	$8.626E-5$
	DEIWO	0	13.7395290033	2.9816440198	2.9817236110	12.66797
10, 20, 40	PSO	−20.0294	−0.0169	−6.2739	1.9062	303.5985
	DEIWO	0	10.9711605462	4.0800150911	4.0797379306	$1.1000E+3$

6. Conclusions

In this paper, based on the IWO algorithm and DE algorithm, we present a new hybrid DEIWO algorithm for solving nonlinear equation systems. The proposed DEIWO approach seems to be very efficient for solving equations systems. We analyzed the case of nonlinear equations systems. We first compared our approach for some simple equations systems having only two equations that were recently used for analyzing the performance of a newly proposed method. The results obtained using the proposed DEIWO optimization approach are very promising. In optimization process, useful information obtained by DE from population is used to guide evolution direction of weeds. Experiments show that the DEIWO is effective and the proposed method could be extended for more higher dimensional systems, although this will also involve increased computational complexity. In a similar manner, we can also solve inequality systems and systems of differential equations, which are part of our future research work.

Acknowledgments

This work is supported by the National Science Foundation of China under Grant no. 61165015, Key Project of Guangxi Science Foundation under Grant no. 2012GXNSFDA053028, Key Project of Guangxi High School Science Foundation under Grant no. 20121ZD008, the Funded by Open Research Fund Program of Key Lab of Intelligent Perception and Image Understanding of Ministry of Education of China under Grant no. IPIU01201100, and the Innovation Project of Guangxi Graduate Education under Grant no. YCSZ2012063.

References

[1] C. L. Karr, B. Weck, and L. M. Freeman, "Solutions to systems of nonlinear equations via a genetic algorithm," *Engineering Applications of Artificial Intelligence*, vol. 11, no. 3, pp. 369–375, 1998.

[2] G. P. Rangaiah, "Evaluation of genetic algorithms and simulated annealing for phase equilibrium and stability problems," *Fluid Phase Equilibria*, vol. 187-188, pp. 83–109, 2001.

[3] N. Chakraborti and P. K. Jha, "Pb-S-O vapor system re-evaluated using genetic algorithms," *Journal of Phase Equilibria and Diffusion*, vol. 25, no. 5, pp. 421–426, 2004.

[4] J. Zhang and X. Wang, "Particle swarm optimization for solving nonlinear equation and system," *Computer Engineering and Applications*, vol. 7, pp. 56–58, 2006.

[5] Z. Wang, "Solving nonlinear systems of equations based on differential evolution," *Computer Engineering and Applications*, vol. 46, no. 4, pp. 54–55, 2010.

[6] D. Wang and Y. Zhou, "Artificial fish-swarm algorithm for solving nonlinear equations," *Application Research of Computers*, vol. 24, no. 6, pp. 242–244, 2007.

[7] J. Zhang, "Artificial bee colony algorithm for solving nonlinear equation and system," http://www.cnki.net/kcms/detail/11.2127.TP.20111114.0947.046.html.

[8] C. Li, H. Lai, and J. Zhou, "Solution of nonlinear equations based on maximum entropy harmony search algorithm," *Computer Engineering*, vol. 37, no. 20, pp. 189–190, 2011.

[9] T. Nguyen Huu and H. Tran Van, "A new probabilistic algorithm for solving nonlinear equations systems," *Journal of Science*, vol. 30, pp. 1–14, 2011.

[10] A. R. Mehrabian and C. Lucas, "A novel numerical optimization algorithm inspired from weed colonization," *Ecological Informatics*, vol. 1, no. 4, pp. 355–366, 2006.

[11] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces ," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[12] C. Grosan and A. Abraham, "A new approach for solving nonlinear equations systems," *IEEE Transactions on Systems, Man, and Cybernetics A*, vol. 38, no. 3, pp. 698–714, 2008.

[13] R. E. Moore, *Methods and Applications of Interval Analysis*, vol. 2 of *SIAM Studies in Applied Mathematics*, SIAM, Philadelphia, Pa, USA, 1979.

[14] J. Verschelde, P. Verlinden, and R. Cools, "Homotopies exploiting newton polytopes for solving sparse polynomial systems," *SIAM Journal on Numerical Analysis*, vol. 31, no. 3, pp. 915–930, 1994.

[15] P. van Hentenryck, D. Mcallester, and D. Kapur, "Solving polynomial systems using a branch and prune approach," *SIAM Journal on Numerical Analysis*, vol. 34, no. 2, pp. 797–827, 1997.

[16] K. Meintjes and A. P. Morgan, "Chemical equilibrium system as numerical test problems," *ACM Transactions on Mathematical Software*, vol. 16, no. 2, pp. 143–151, 1990.

[17] A. Morgan and A. Sommese, "Computing all solutions to polynomial systems using homotopy continuation," *Applied Mathematics and Computation*, vol. 24, no. 2, pp. 115–138, 1987.

[18] A. P. Morgan, *Solving Polynomial Systems Using Continuation for Scientific and Engineering Problems*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1987.

[19] X. Gao and K. Jiang, "Survey on geometric constraint solving," *Journal of Computer Aided Design & Computer Graphics*, vol. 16, no. 4, pp. 385–396, 2004.

[20] S. Wang and Z. Wang, "Study of the application of PSO algorithms for nonlinear problems," *Journal of Huazhong University of Science and Technology*, vol. 33, no. 12, pp. 4–7, 2005.