Research Paper
Computational Methods for Differential Equations
http://cmde.tabrizu.ac.ir
Vol. 10, No. 1, 2022, pp. 274-297
DOI:10.22034/cmde.2020.37474.1658

Design of normal distribution-based algorithm for solving systems of nonlinear equations

Amir Khakbaz*
Department of Industrial Engineering, School of Engineering, Damghan University, Damghan, Iran.

Abstract

In this paper, a completely new statistical based approach is developed for solving the system of nonlinear equations. The developed approach utilizes the characteristics of the normal distribution to search the solution space. The normal distribution is generally introduced by two parameters, i.e., mean and standard deviation. In the developed algorithm, large values of standard deviation enable the algorithm to escape from a local optimum, and small values of standard deviation help the algorithm to find the global optimum. In the following, six benchmark tests and thirteen benchmark case problems are investigated to evaluate the performance of the Normal Distribution-based Algorithm (NDA). The obtained statistical results of NDA are compared with those of PSO, ICA, CS, and ACO. Based on the obtained results, NDA is the least time-consuming algorithm that gets high-quality solutions. Furthermore, few input parameters and simple structure introduce NDA as a user friendly and easy-to-understand algorithm.

Keywords. Normal distribution-based Algorithm (NDA); Nonlinear equations; Numerical optimization; Meta-heuristic.
2010 Mathematics Subject Classification. 90C30, 65H20, 65J15.

1. Introduction

A system of nonlinear equations consists of two or more equations with two or more variables, in which at least one equation is not linear, that is being solved simultaneously. Various areas of pure and applied sciences such as chemistry, physics, mechanics, robotics, aircraft control, engineering, statistics, management sciences, economics, biology, and medicine are related to the system of nonlinear equations [29]. Because of the computational complexity of the system of nonlinear equations, there are few methods for this system that the optimal solution often comes. A historic review shows that various approaches have been employed to solve the systems of nonlinear equations, which some of them are presented in Table 1. So far, several Evolutionary Algorithms (EAs), such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Differential Evolution (DE), Ant Colony Optimization (ACO), Imperialist Competitive Algorithm (ICA), Cuckoo Search (CS), Firefly Algorithm (FA), and others have been used for solving systems of nonlinear equations. EAs are known as population-based algorithms, in which a population of solutions is used in each iteration of the optimization process that uses operators to modify the solutions aiming at gradually evolving the solutions based on a fitness function. EAs resolve some of the major difficulties of exact methods and can effectively replace them in solving optimization problems. These algorithms typically attempt to find a high-quality solution to their optimization problems through an iterative trial-and-error procedure with reduced computational time. Some of them, such as PSO, ICA, and CS, have attracted more attention due to their simplicity, speed, and capability of global optimum search.

PSO was inspired by the behavior of social organisms in groups, such as birds and fishes. PSO has robust capability in global search and has been extensively used for solving systems of nonlinear equations, alone or as a hybrid with other methods, especially in recent years. Some research studies indicate that hybrids of PSO with other search algorithms, such as cuckoo search [11], NelderMead simplex search [1], Grey Wolf Optimizer [32] and others, get more

Received: 23 December 2019 ; Accepted: 06 December 2020.
∗ Corresponding author. Email: a.khakbaz@du.ac.ir.

TABLE 1. List of approaches for solving the system of nonlinear equations.

Authors	Approach
El-Emary and El-Kareem, 2008 [6]	Genetic algorithm
Li and Zeng, 2008 [16]	A neural-network algorithm
Hirsch et al., 2009 [8]	GRASP
Jaberipour et al., 2011 [13]	Particle swarm algorithm
Pourjafari and Mojallali, 2012 [27]	Invasive weed optimization algorithm
Abdollahi et al., 2013 [3]	Imperialist competitive algorithm
Oliveira and Petraglia, 2013 [25]	Fuzzy adaptive simulated annealing
Sharma and Arora, 2013 [30]	Weighted-Newton methods
Wang and Zhou, 2014 [35]	Pattern search firefly algorithm
Abdollahi et al. 2016 [2]	Cuckoo optimization algorithm
Raja et al., 2018 [28]	Particle swarm optimization hybrid with NelderMead method
Zhang et al., 2019 [37]	Niche cuckoo search algorithm
Pei et al., 2019 [26]	Continuous Variable Neighborhood search
Chen and Kelley, 2019 [5]	Energy Direct Inversion on the Iterative Subspace

efficient algorithms in comparison with the PSO. ICA is another evolutionary algorithm that uses a mathematical model of human social evolution. It was first represented by Atashpaz-Gargari and Lucas in 2007 and recently has been broadly applied to solve nonlinear optimization problems [20]. In ICA, generally, it is evident that competition among the empires will eventually lead to a global empire unless the algorithm terminates within a predetermined number of iterations before arriving at a global empire [9]. ICA has been tested on various standard test problems, which the obtained results demonstrate that ICA is a very efficient algorithm concerning both speed and accuracy [21]. In operations research, the CS algorithm is a recently developed evolutionary optimization algorithm [36]. CS is a nature-inspired algorithm, which is based on the brood parasitism of some cuckoo species laying their eggs in the nests of other birds. The applications of CS into nonlinear optimization problems have displayed its efficiency [11, 37]. Moreover, CS has some benefits over the other evolutionary algorithms that include simplicity, fewer input parameters compared with the other evolutionary algorithms, and ease of hybridization with other algorithms [31].

Although many EAs have been used to solve the system of nonlinear equations, each of them may fail to find the optimal solution due to their weaknesses. Some of these weaknesses are as follows. PSO suffers from a significant increase in search complexity with an increase in the dimension of the relevant problem. Besides the high number of input parameters, it is difficult for PSO to define initial design parameters [14]. Concerning ICA, it has more parameters than PSO, and thus it is more difficult to adjust the input parameters. Like many other EAs, ICA may result in premature convergence and gets the local optimum, especially in the multimodal optimization problems [9]. Generally, the CS algorithm has fewer drawbacks in comparison to the previous algorithms. The determination of CS input parameters is one of the drawbacks of the algorithm because the parameters should be changed with the increase of iterations. Also, as we know, the locations of some nests may be out of the boundary of the optimization problem. To cope with this problem, the CS algorithm replaces these unacceptable nests with the boundary values. As a consequence, a lot of nests can be made at the same location on the boundary, which is inefficient [18].

Besides of many advantages, EAs suffer from some disadvantages. First, they do not guarantee an optimal solution to specific optimization problems within predictable run time. Second, it may need much parameter tuning by trial-and-error procedure, and at last, EAs usually need lots of computational resources. To overcome some of these problems, we develop a new statistical-based approach, i.e., Normal Distribution-based Algorithm (NDA). To the best of our knowledge, statistical approaches have not been used to solve the system of nonlinear equations previously. In this research, the normal distribution is considered as the basis of the developed approach. NDA requires relatively few parameters that help the user to tune the algorithms parameters easily. NDA is a simple structure algorithm that facilitates coding procedure and computation. Due to its simplicity, NDA is introduced as a high-speed approach for solving large scale problems. Since NDA uses a contraction expansion coefficient, i.e., standard deviation, it can balance the local and global searches during the optimization process. The rest of this paper is structured as follows: In section 2, we describe the NDA in detail. In section 3, six test problems and thirteen well-known systems are used to evaluate the efficiency of the developed algorithm in comparison with some other EAs. Finally, the discussion and conclusion of this study are presented in sections 4 and 5, respectively.

2. Normal Distribution-based Algorithm (NDA)

In general, the system of nonlinear equations (2.1) is defined as:

$$\begin{cases} f_1(x_1, x_2, \dots, x_n) = 0 \\ f_2(x_1, x_2, \dots, x_n) = 0 \\ \vdots \\ f_n(x_1, x_2, \dots, x_n) = 0 \end{cases} \tag{2.1}$$

Above system can be transformed into an optimization problem with a single objective function by using the auxiliary function Eq. (2.2):

$$\min f(x) = \sum_{i=1}^{n} f_i^2(x), \quad x = (x_1, x_2, \dots, x_n) \tag{2.2}$$

Since $f_i^2(x) \geq 0$, it is clear that $f(x) \geq 0$. If $x^* : f(x_*) = 0$, then x^* is a global minimum of f and subsequently $f_1(x^*) = f_2(x^*) = \cdots = f_n(x^*) = 0$ and x^* is a root for Eq. (2.1).

In this study, a new search algorithm is developed based on the characteristics of normal distribution. NDA is applied to solve the system of nonlinear equations. The normal distribution, discovered by De Moivre in the 18^{th} century, is the most important and most widely known and used of all distributions in statistics. Many natural phenomena follow a normal distribution, or bell curve, pattern. Some of these phenomena are given below [22]:

- Human characteristics such as height, blood pressure level, cholesterol, shoe size, reaction times, and lung capacity
- Weight of products in continuous production lines such as food, steel, petroleum, and oil industry
- Weather patterns such as temperature and rainfall
- Delivery time in a shipping company
- Service time in cleaning companies

Normal distributions are introduced by two parameters, i.e., mean (μ) and standard deviation (σ). Because of the shape of the probability density function (pdf), the normal distribution is often called the bell curve. These distributions are symmetric about their mean. In other words, fifty percent of values are less than μ, and fifty percent of values are greater than μ. Another important property of the normal distribution is that the mean is equal to mode and median (Figure1). The pdf of the normal distribution is defined by Eq. (2.3). In this function, x is the random variable, μ is the mean and, σ is the standard deviation of the distribution. In the normal distribution, there is approximately a 68% probability that a value falls within one standard deviation from the mean, i.e., $\mu \pm \sigma$. Also, there are approximately 95% and 99.7% chance that a value falls within 2 and 3 standard deviations from the mean, respectively (Figure2).

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}; \quad -\infty \leq X \leq +\infty \tag{2.3}$$

This study aims to develop a new statistical method for solving a system of nonlinear equations. The developed approach utilizes the properties of the normal distribution to solve the system. In the first step of the NDA, P individuals are generated randomly as the initial population. All generated individuals are evaluated by a fitness function, which corresponds to the objective function of the problem. After sorting the population, the top 100p% of individuals are selected to produce a new generation based on the local search factor, k, and the rest of individuals are used to produce a new generation based on the global search factor, σ_g. As said before, normal distributions are

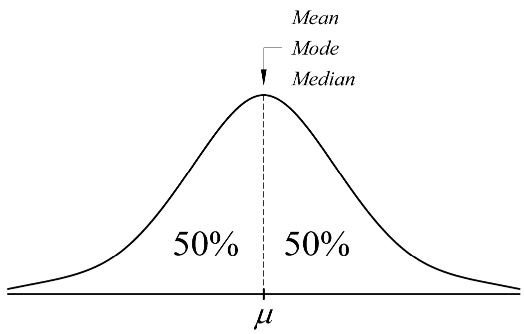


FIGURE 1. Shape of normal pdf.

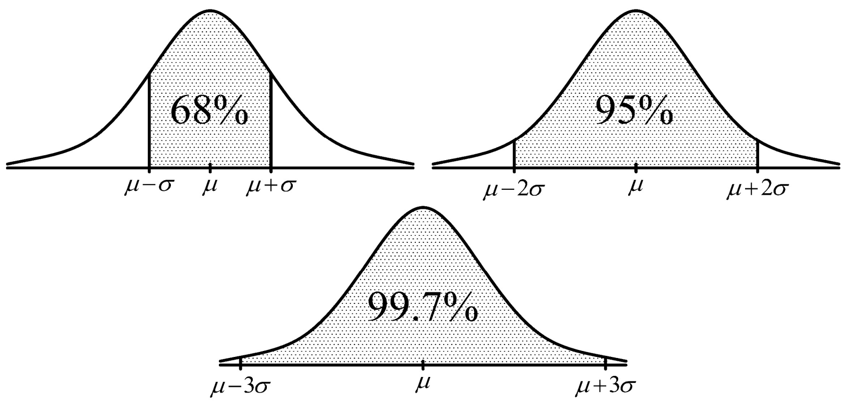


FIGURE 2. Standard normal characteristics.

defined by two parameters, the mean and the standard deviation. The selected individuals are used as the mean value of the normal distribution. Also, a predefined value is taken as the standard deviation of the normal distribution. Then, new individuals are produced by using the pdf of the normal distribution. These steps are repeated until a criterion is satisfied. Figure 3 shows the pseudo-code of NDA. The flowchart of the proposed algorithm is described in Figure4. In the following, each step of the NDA will be discussed in detail.

Step 1. Initialization

In this step, P individuals are generated randomly in the interval on which the X is defined.

Step 2. Evaluation

In this step, the fitness function is calculated for each individual that was generated in the previous step. The fitness function corresponds to the objective function of the relevant problem.

Step 3. Selection

All individuals are sorted in ascending/descending order of the values of their fitness function. Then, the top 100p% of individuals are selected to produce a new generation based on the local search factor, k, and the rest of individuals are used to produce a new generation based on the global search factor, σ_g. However, p plays an important role in the quality of the final solution. As p increases, NDA pays more attention to the local search area and vice versa. Also, σ_l is determined so that by increasing the number of iterations, NDA would be able to get closer to the optimal

```
Input:   Number of populations, P
         Global search factor, σ_g
         Local search factor, k
         Selection factor, p
Output:  The optimal solution, x*
         The optimal value of fitness function, f(x*)
Initialize a population P;
While termination conditions are not met do
    Calculate the fitness value of each individual;
    Sort top P individuals in a non-dominated order;
    Set f(x*) value for x*;
    Select the top 100p% of individuals to generate new individuals as:
        for i = 1 to the number of decision variables n do
            x_{t+1} = x_t + (σ_l. randn); where σ_l is obtained by "Eq (2.4)"
    Select the rest of individuals to generate new individuals as:
        for i = 1 to the number of decision variables n do
            x_{t+1} = x_t + (σ_g. randn);
    Combine new and old individuals into one population 2P;
Set the best fitness function value f(x*) as the best solution x*.
```

FIGURE 3. Pseudo-code of the NDA.

solution. Eq. (2.4) displays σ_l in terms of the number of iterations and k.

$$\sigma_l = 10^{-Iterations\backslash k} \tag{2.4}$$

Step 4. Generation

As mentioned previously, the normal distribution is defined with two parameters, μ and σ. The notation $X \sim N(\mu, \sigma)$ means that X is distributed as a normal random variable with the μ as its mean and the σ as its standard deviation. If $Z = \frac{(X-\mu)}{\sigma}$, then Z follows the standard normal distribution, i.e., $Z \sim N(0,1)$. Also, we have:

$$X = \mu + Z\sigma. \tag{2.5}$$

Based on Eq. (2.5). X is dependent on the mean, the standard deviation, and on the standard normal variable. In the NDA, all individuals from the selection step (X_{Old}) are considered as the mean, i.e., $\mu = X_{Old}$. Then, new individuals (X_{New}) are obtained by Eq. (2.6).

$$X_{New} = X_{Old} + Z\sigma. \tag{2.6}$$

In Eq. (2.6), the standard deviation is defined according to the range of X. For small values of standard deviation (σ_l), the obtained values of X_{New} are probably close to X_{Old}. Also, this probability is reduced for large values of standard deviation (σ_g). σ_l and σ_g, respectively, play the roles of crossover and mutation in the Genetic Algorithm (GA). In each iteration of NDA, some new individuals are produced by using σ_l and some others by using σ_g. A predefined portion of the population is selected to make new individuals by use of σ_l, we refer to as p, and the rest of the population is used to generate new individuals by use of σ_g. As mentioned previously, $Z \sim N(0,1)$, therefore:

$$f(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{Z^2}{2}}; \quad -\infty \leq X \leq +\infty \tag{2.7}$$

The cumulative distribution function of Z is given by Eq. (2.8).

$$F(z) = P(Z \leq z) = \int_{-\infty}^{z} \frac{1}{2\pi} e^{-\frac{z^2}{2}} dz \tag{2.8}$$

For ease of computation, $F(z)$ can be calculated using the standard normal table. Since $F(z)$ is the cumulative probability of Z, then $0 \leq F(z) \leq 1$. In Eq. (2.6), Z should be replaced by a random number, which has been generated by the standard normal distribution. It is done as follows. First, a random number is generated between 0 and 1. This number is considered as the cumulative probability of Z, i.e., $F(z)$. At last, the relevant Z is obtained by using the standard normal table. The following numerical example shows how it works in practice. Example: Let us assume, 0.95 was generated as a random number between 0 and 1. Therefore, $F(z) = 0.95$. Based on Eq. (2.8), $P(Z \leq z) = 0.95$. By using the standard normal table, we have $P(Z \leq 1.65) = 0.95$. Therefore, Eq. (2.6) can be rewritten as, $X_{New} = X_{Old} + 1.65\sigma$.

Step 5. Termination

The maximum number of iterations or running time can be considered as stop criteria. Also, no improvement in the best solution for a predefined number of iterations can be used as another stop criterion.

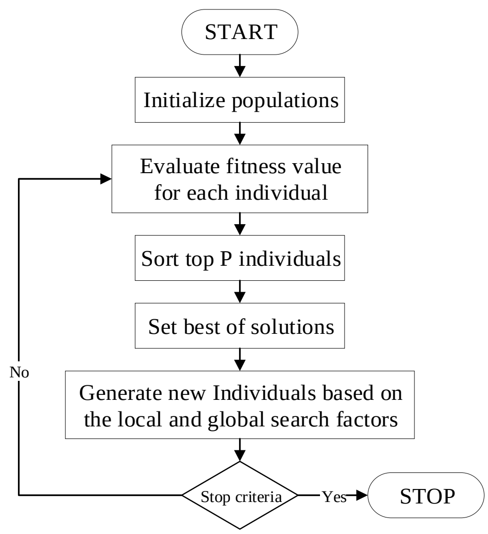


FIGURE 4. NDA flowchart.

3. Experiments and results

In this section, six tests functions and thirteen systems are used to demonstrate the performance of the NDA. The obtained results of the NDA are compared with those obtained by the well-known evolutionary algorithms, i.e., Particle Swarm Optimization (PSO), Imperialist Competitive Algorithm (ICA), Cuckoo search (CS) and Ant Colony Optimization (ACO). All algorithms were coded in MATLAB 8.5 and simulations were run on a laptop with Intel(R) Core (TM) i5-2450M CPU with the speed of 2.50GHz and the installed memory of 4 GB. The used parameters for solving the benchmark problems are listed in Table 2.

Table 2. Parameters setting for algorithms.

Algorithm	Parameters
PSO	Personal weight $(w) = 1$, reduction factor of personal weight $(w_{rf}) = 0.99$, personal best learning coefficient $(c_1) = 2$, global best learning coefficient $(c_2) = 2$, lower bound of velocity $(lb_v) = -0.8$, upper bound of velocity $(ub_v) = 0.8$, number of populations $(P) = 20$ and, maximum number of iterations $(Itr) = 8000$.
ICA	Number of empires/imperialists $(n_e mp) = 10$, selection pressure $(\alpha) = 1$, assimilation coefficient $(\beta) = 1.5$, revolution probability $(p_r) = 0.05$, revolution rate $(\mu) = 0.1$, colonies mean cost coefficient $(\zeta) = 0.2$, number of populations $(P) = 20$ and, maximum number of iterations $(Itr) = 8000$.
CS	Discovery rate of alien eggs/solutions $(p_a) = 0.25$, step size $(\alpha) = 1$, number of populations $(P) = 20$ and, maximum number of iterations $(Itr) = 8000$.
ACO	Sample size $(n_s) = 40$, intensification factor $(q) = 0.5$, deviation-distance ratio $(\zeta) = 1$, number of populations $(P) = 20$ and, maximum number of iterations $(Itr) = 8000$.
NDA	Local search factor $(\sigma_g) = 1$, global search factor $(k) = 500$, selection factor $(p) = 0.95$, number of populations $(P) = 20$ and, maximum number of iterations $(Itr) = 8000$.

3.1. **Simulation results on six benchmark test functions.** Here, six benchmark test functions are used to evaluate the performance of the NDA. Table 3 represents the details of these functions, including mathematical formulation, dimension, range, and optimal solution. Benchmark functions include 20-dimension Rastrigin function [24], Hartmans function [23], Six-Hump camelback [19], a 30-dimension function of [13], 30-dimension Ackley function, and 30-dimension sphere function, respectively. The obtained results are compared with PSO, ICA, CS, and ACO (Table 4). As shown in this table, some statistical data such as mean, standard deviation, and rank are computed for all algorithms. The average ranks determined by Friedman test [17] are also used to assess the success of each algorithm. In other words, the performance of an algorithm increases as rank decreases. Best results are highlighted in bold font. Based on the obtained results, NDA reaches the best solution in 3 out of 6 functions, i.e., $f_2, f_3, andf_4$. In all tests, NDA converges the optimal solution in the shortest time. As indicated in Table 4, NDA has the best average rank, i.e., 2.29. For further investigation, the convergence history of all mentioned algorithms is illustrated in Fig.5. The convergence curves indicate that NDA has strong and steady convergence in terms of the number of iterations.

3.2. **Benchmark cases and simulation results.** In this section, the performance of NDA is studied on 13 benchmarks of nonlinear systems (as seen in Tables 5 and 6). To investigate the performance of the NDA, 50 consecutive runs are performed on each system for each of the above-mentioned algorithms, i.e., NDA, PSO, ICA, CS, and ACO. As said above, the maximum number of iterations and population size for all algorithms are set to 8000 and 20, respectively, as in [33]. The obtained statistical results, including best, average and worst solutions, standard deviation and solution time are listed for each system in Tables 7-17, 19, and 20. The best values are shown in bold font in all tables. The obtained results indicate that the NDA is efficient in most of the benchmark systems. In addition, Fig.7 displays the convergence history of all mentioned algorithms. It is noted that the values for parameters used in the algorithms are as shown in Table 2. In the following, each case is investigated in detail.

Case1 is known as the Geometry size of the thin wall rectangle girder section [33]. In this case, x_1 is the width, x_2 is the height and x_3 is the thickness of the section. Table 7 represents the obtained results from the NDA and other algorithms. As can be seen, CS gets the optimal solutions among all algorithms. However, NDA has the minimum

TABLE 3. Benchmark test function.

Function	Dim	Range	Optimal solution
$f_1(x) = \sum_{i=1}^{n}[x_i^2 - 10\cos(2\pi x_i + 10]$	20	[-5.2,5.2]	$x^* = (0,\ldots,0)^T$ $f(x^*) = 0$
$f_2(x) = -\sum_{i=1}^{n} c_i \exp[-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2]$	4	[0,1]	$0.14 < x_i^* < 0.66$ $f(x^*) = -3.3220$
c=[1 1.2 3 3.2]			
$p_{ij} = \begin{bmatrix} 0.1312 & 0.1696 & 0.5569 & 0.0124 & 0.8283 & 0.5886 \\ 0.2329 & 0.4135 & 0.8307 & 0.3736 & 0.1004 & 0.9991 \\ 0.2348 & 0.1415 & 0.3522 & 0.2883 & 0.3047 & 0.6650 \\ 0.4047 & 0.8828 & 0.8732 & 0.5743 & 0.1091 & 0.0381 \end{bmatrix}$			
$a_{ij} = \begin{bmatrix} 10 & 3 & 17 & 3.5 & 1.7 & 8 \\ 0.05 & 10 & 17 & 0.1 & 8 & 14 \\ 3 & 3.5 & 1.7 & 10 & 17 & 8 \\ 17 & 8 & 0.05 & 10 & 0.1 & 14 \end{bmatrix}$			
$f_3(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	[-3,3]	$x^* = (0.089842012773979, -0.712656402251958)^T$ $f(x^*) = -1.031628453489878$
$f_4(x) = \sum_{i=1}^{n}[\sin(x_i + \sin(\frac{(2x_i)}{3})]$	30	[3,13]	$5.361 < x_i^* < 5.364$ $f(x^*) = -1.21598n$
$f_5(x) = -20\exp(-0.2\sqrt{(\frac{1}{n}\sum_{i=1}^{n} x_i^2)} - \exp(\frac{1}{n}\sum i = 1^n \cos 2\pi x_i)$	30	[-32,32]	$x^* = (0,\ldots,0)^T$ $f(x^*) = 0$
$+20 + \exp(1)$			
$f_6(x) = \sum_{i=1}^{n} x_i^2$	30	[-100,100]	$x^* = (0,\ldots,0)^T$ $f(x^*) = 0$

average, standard division, and worst solution. For *Case2* [3], the optimal solution is (4, 3, 1). As shown in Table 8, all the algorithms except the ICA succeed to reach the optimal solution. Moreover, NDA and CS meet the global optimum in all 50 runs. In other words, for NDA and CS, average solution, standard division, and worst solution equal zero. The next case, *Case3*, is minimized at (-0.2905146, 1.0842151) or (1.0842151, -0.2905146). Based on Table 9, all algorithms can find the optimal solution. Besides, CS acts better than the other ones and gets the optimal solution in all 50 runs. It can be seen easily that NDA tracks the optimal solution almost as well as the other algorithms.

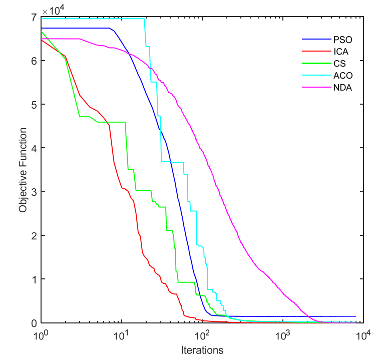
Case4 is known as Neurophysiology application [25]. As indicated in Table 10, this case has more than one optimal solution. All the algorithms except the CS succeed to reach one of the optimal solutions. In terms of the other factors, i.e., minimum average, standard division, and worst solution, PSO displays the best efficiency among these algorithms. According to the obtained results, NDA is ranked as the second-best algorithm. Table 11 depicts the statistically obtained results for *Case5* [34]. All algorithms get the optimal solution, i.e., (0.5, 3.141592654). By considering all factors, CS and NDA are ranked as best and second-best algorithms, respectively. It is observed that the results obtained by NDA are so close to those obtained by CS, where the difference is less than 5.0E-34. For *Case6*, the minimum value of the objective function is approximately 3.0967E-33 that is obtained by NDA and CS. Based on Table 12, NDA performs the best among all five algorithms. As shown in this table, ACO gets solutions that are relatively close to those of NDA. Also, there are large differences between results from PSO, ICA, and CS and those from NDA.

Table 13 compares the obtained results of the five algorithms for *Case7* [10]. As indicated in this table, CS reaches the optimal solution, i.e., (0.5000000, 0.0000000, -0.5235988), in all runs. Besides, the obtained results of the other four algorithms are relatively close to each other. *Case8* is known as Browns almost linear system [33]. Table 14 shows that NDA and CS obtain the optimal solution, i.e., (0.9163546, 0.9163546, 0.9163546, 0.9163546, 1.4182271), in all runs. ACO acts best among the other algorithms. Similar to *Case8*, both NDA and CS are ranked as the best ones, and ACO is ranked as the second-best for solving *Case9* [15]. For this case, as indicated in Table 15, the optimal solution is (-1, 1, -1, 1, -1, 1). Table 16 displays the obtained results of the algorithms for *Case10* [15]. According to this table, NDA is accepted as the best algorithm for solving *Case10*. Here, the optimal solution is (0.5149333, 0.5149333, 0.5149333, 0.5149333), which has been obtained by all the algorithms.

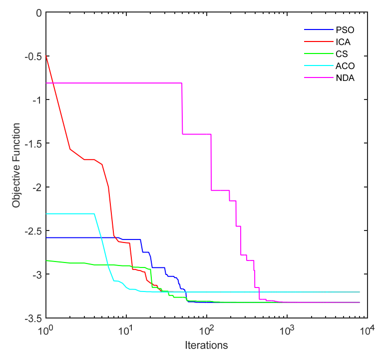
TABLE 4. Statistical results of test functions.

Function	Algorithm	NDA	PSO	ICA	CS	ACO
f_1	min	9.6820E-01	6.0393E+02	**3.5527E-14**	1.0001E+01	1.8439E+02
	mean	2.9286E+00	1.4277E+03	**1.7251E+00**	1.6540E+01	2.0645E+02
	std	1.3147E+00	4.2638E+02	**1.0877E+00**	4.8124E+00	1.0720E+01
	max	6.2574E+00	2.3817E+03	**4.9749E+00**	2.8935E+01	2.2615E+02
	Time(s)	**2.622**	6.642	7.513	5.811	65.959
	Rank	1.73	5.00	**1.27**	3.00	4.00
f_2	min	**-3.3220E+00**	**-3.3220E+00**	**-3.3220E+00**	**-3.3220E+00**	**-3.3220E+00**
	mean	**-3.3220E+00**	-3.2327E+00	-3.3022E+00	**-3.3220E+00**	-3.2863E+00
	std	**1.3323E-15**	1.1084E-01	4.4309E-02	**1.3323E-15**	5.4487E-02
	max	**-3.3220E+00**	-2.8685E+00	-3.2031E+00	-3.3220E+00	-3.2031E+00
	Time(s)	**1.427**	6.727	7.985	5.016	16.230
	Rank	2.88	4.10	3.17	2.88	**1.97**
f_3	min	**-1.0316E+00**	**-1.0316E+00**	**-1.0316E+00**	**-1.0316E+00**	**-1.0316E+00**
	mean	**-1.0316E+00**	**-1.0316E+00**	**-1.0316E+00**	**-1.0316E+00**	**-1.0316E+00**
	std	**0.0000E+00**	6.6613E-16	**0.0000E+00**	**0.0000E+00**	6.6613E-16
	max	**-1.0316E+00**	**-1.0316E+00**	**-1.0316E+00**	**-1.0316E+00**	**-1.0316E+00**
	Time(s)	**2.885**	6.007	8.271	4.482	7.910
	Rank	**2.00**	4.50	**2.00**	**2.00**	4.50
f_4	min	**-3.6479E+01**	-3.1516E+01	-3.3502E+01	**-3.6479E+01**	-2.8405E+01
	mean	**-3.6479E+01**	-2.7182E+01	-2.9376E+01	-3.6248E+01	-2.5622E+01
	std	**6.7092E-04**	2.4648E+00	2.7347E+00	4.1984E-01	1.3776E+00
	max	**-3.6477E+01**	-2.2582E+01	-2.1590E+01	-3.5487E+01	-2.2976E+01
	Time(s)	**3.903**	6.603	7.464	5.845	62.752
	Rank	1.77	4.08	3.32	**1.23**	4.60
f_5	min	7.9936E-15	1.1546E-14	3.2863E-14	**4.4409E-15**	**4.4409E-15**
	mean	1.3678E-14	1.6234E+00	4.9679E-14	8.4632E-01	**6.9278E-15**
	std	3.3829E-15	1.5368E+00	9.0313E-15	6.5488E-01	**1.6281E-15**
	max	2.2204E-14	5.5623E+00	6.8390E-14	2.0119E+00	**7.9936E-15**
	Time(s)	**4.196**	6.899	8.608	6.332	68.225
	Rank	2.23	4.37	3.70	3.33	**1.37**
f_6	min	2.3756E-43	**1.7487E-81**	8.1800E-36	7.4433E-51	2.0992E-62
	mean	3.2182E-43	1.1905E+01	3.4483E-32	7.0278E-49	**1.9500E-58**
	std	5.2928E-44	4.5465E+01	1.3111E-31	1.3154E-48	**5.4424E-58**
	max	4.3573E-43	2.1429E+02	7.0443E-31	6.6005E-48	**2.6311E-57**
	Time(s)	**2.193**	6.813	7.823	5.369	68.225
	Rank	3.10	4.57	4.20	2.10	**1.03**
Average rank		**2.29**	4.44	2.94	2.43	2.91

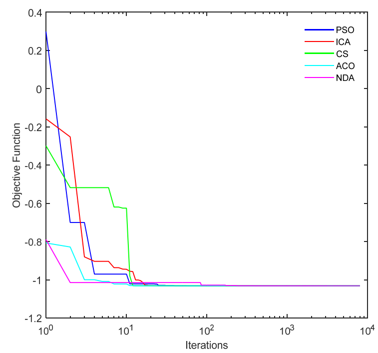
Tables 17, 19, and 20 represent the statistical results of the five algorithms for *case11* [10], *case12* [7], and *case13* [25], respectively. NDA reaches the minimum best value of the objective function in *case11* and *case12*. In *case11*, both NDA and ICA can be determined as the best algorithms that converge to 2.2720E-33 in all 50 runs. Besides, CS is ranked as the second-best algorithm, with an insignificant difference in standard deviation. Given the Table 17, NDA and CS get similar results. Here, it is difficult to identify the top-ranked one. PSO and ICA find weakest solutions among all the five algorithms. Based on Table 20, in *case13*, like many other cases, CS performs efficiently and gets the best values of all factors. Here, NDA can be introduced as the third-best algorithm for solving the *case13*.



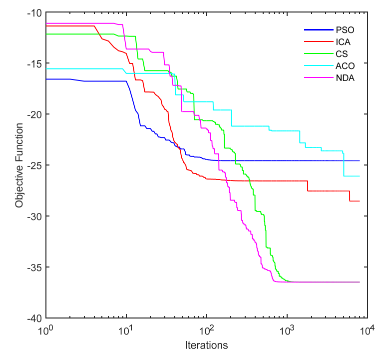
(a) f_1



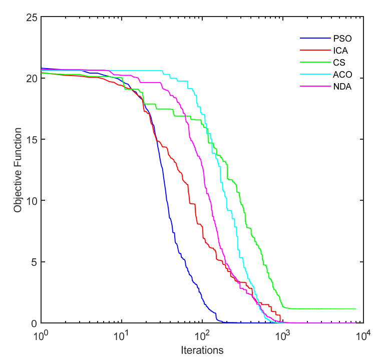
(b) f_2



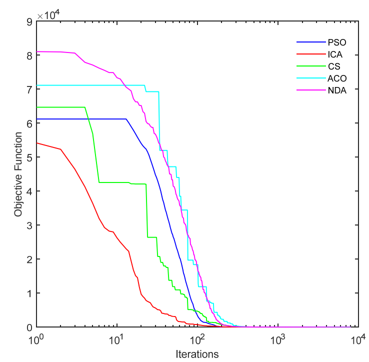
(c) f_3



(d) f_4



(e) f_5



(f) f_6

FIGURE 5. Comparision of the convergence of PSO, ICA, CS, ACO, and NDA for tests functions 1-6.

As shown in Tables 7-17, 19, and 20, for all cases, NDA has the least run time. For NDA, PSO, ICA, CS, and ACO, average solution times (in terms of second) are 2.06, 6.46, 7.78, 5.14, and 14.84, respectively. These data show that NDA and ACO are the fastest and slowest algorithms, respectively. Fig.6 depicts the mean ranks of thirteen of nonlinear systems with 50 independent runs. According to this figure, NDA has the minimum average rank in 5

Table 5. Benchmark cases (1-7).

Case	
Case 1	$\begin{cases} x_1x_2-(x_1-2x_3)-165=0 \\ (x_1x_2^3/12)-[(x_1-2x_3)(x_2-2x_3)^3/12]-9369=0 \\ 2t(x_2-x_3)^2(x_1-x_3)^2/(x_2+x_1-2x_3)-6835=0 \end{cases}$ $x_i \in [-30,30]$
Case 2	$\begin{cases} x_1^{x_2}+x_2^{x_1}-5x_1x_2x_3-85=0 \\ x_1^3-x_2^{x_3}-x_3^{x_2}-60=0 \\ x_1^{x_3}+x_3^{x_1}-x_2-2=0 \end{cases}$ $x_1 \in [3,5], \quad x_2 \in [2,4], \quad x_3 \in [0.5,2]$
Case 3	$\begin{cases} x_1^3-3x_1x_2^2-1=0 \\ 3x_1^2x_2-x_2^3+1=0 \end{cases}$ $x_i \in [-10,10]$
Case 4	$\begin{cases} x_1^2+x_3^2-1=0 \\ x_2^2+x_4^2-1=0 \\ x_5x_3^3+x_6x_4^3=0 \\ x_5x_1^3+x_6x_2^3=0 \\ x_5x_1x_3^2+x_6x_2x_4^2=0 \\ x_5x_3x_1^2+x_6x_4x_2^2=0 \end{cases}$ $x_i \in [-10,10]$
Case 5	$\begin{cases} 0.5\sin(x_1x_2)-0.25x_2\pi-0.5x_1=0 \\ (1-0.25/\pi)(\exp(2x_1)-e)+ex_2/\pi-2ex_1=0 \end{cases}$ $x_1 \in [0.25,1], \quad x_2 \in [1.5,2\pi]$
Case 6	$\begin{cases} 4.731\times10^{-3}x_1x_3-0.3578x_2x_3-0.1238x_1+x_7-1.637\times10^{-3}x_2 \\ -0.9338x_4-0.3571=0 \\ 0.2238x_1x_3+0.7623x_2x_3+0.2638x_1-x_7-0.07745x_2-0.6734x_4 \\ -0.6022=0 \\ x_6x_8+0.3578x_1+4.731\times10^(-3)x_2=0 \\ -0.7623x_1+0.2238x_2+0.3461=0 \\ x_1^2+x_2^2-1=0 \\ x_3^2+x_4^2-1=0 \\ x_5^2+x_6^2-1=0 \\ x_7^2+x_8^2-1=0 \end{cases}$ $x_i \in [-1,1]$
Case 7	$\begin{cases} 3x_1-\cos(x_2x_3)-0.5=0 \\ x_1^2-625x_2^2-0.25=0 \\ \exp(-x_1x_2)+20x_3+(10\pi-3)/3=0 \end{cases}$ $x_i \in [-10,10]$

cases, i.e., *case2*, *case6*, *case8*, *case9*, and *case10*. Fig.7 displays the convergence history of all mentioned algorithms. The convergence curves indicate that NDA has strong and steady convergence in terms of the number of iterations. According to this figure, the convergence trend of NDA follows a uniform pattern in all cases. It ensures that NDA combines local and global search in a reasonable manner.

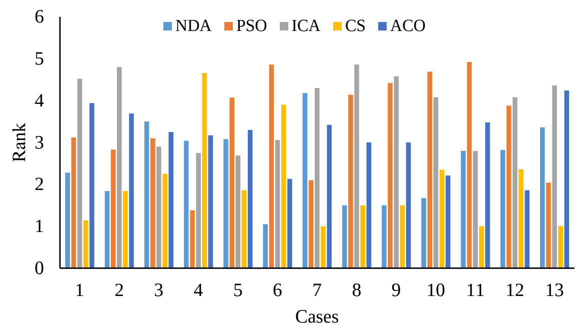


Figure 6. The mean ranks of nonlinear systems with 50 runs.

4. Discussion

In the previous section, we considered six benchmark test functions and thirteen benchmark systems of nonlinear equations for evaluating the performance of NDA. The obtained statistical results of the five algorithms, i.e., NDA, PSO, ICA, CS, and ACO, including the best minimum, maximum and average values of the objective functions, standard deviation, solution time, and ranks have been listed in the relevant tables. Convergence curves illustrate the behavior of each algorithm to reach the optimal solution in terms of the number of iterations. It can be concluded from the benchmark tests functions that NDA gets the optimal solution in half of the tests. In the rest half, NDA obtains solutions that are very close to the best ones. Regarding the cases, NDA finds the optimal solutions in 7 out of 13 cases (i.e., 54% of them), and similar to the tests, for the rest 46%, the obtained solutions are very close to the best ones. Analyzing the mean of standard deviations, it is observed that NDA has the lowest mean value for both benchmark tests and cases, i.e., 0.2192 and 6.7658E-05, respectively. It indicates that the obtained solutions of NDA are more likely to be close to the best ones than those of the other four algorithms. Moreover, we compared the solution times of the algorithms for all tests and case problems. As a result, it can be seen easily that NDA consumes less time in comparison with the other algorithms. As another result, the outcomes of NDA indicate that the solution times are very stable across the tests and case problems. As another important factor, ranks could be used to evaluate the performance of NDA. As mentioned previously, the average ranks of NDA, 2.29 for tests and 2.51 for case problems, ensure that it can be introduced as an efficient algorithm in solving the benchmark problems.

Table 6. Benchmark cases (8-13).

Case	
Case 8	$\begin{cases} 2x_1 + x_2 + x_3 + x_4 + x_5 - 6 = 0 \\ x_1 + 2x_2 + x_3 + x_4 + x_5 - 6 = 0 \\ x_1 + x_2 + 2x_3 + x_4 + x_5 - 6 = 0 \\ x_1 + x_2 + x_3 + 2x_4 + x_5 - 6 = 0 \\ x_1 x_2 x_3 x_4 x_5 = 1 \end{cases}$ $x_i \in [-2, 2]$
Case 9	$\begin{cases} x_1 + 0.25x_2^2 x_4 x_6 + 0.75 = 0 \\ x_2 + 0.405exp(1 + x_1 x_2) - 1.405 = 0 \\ x_3 - 0.5x_4 x_6 + 1.5 = 0 \\ x_4 - 0.605exp(1 - x_3^2) - 0.395 = 0 \\ x_5 - 0.5x_2 x_6 + 1.5 = 0 \\ x_6 - x_1 x_5 = 0 \end{cases}$ $x_i \in [-2, 2]$
Case 10	$\begin{cases} x_i - \cos(2x_i - \sum_(j = 1)^4 x_j) = 0; \qquad 1 \leq i \leq 4 \end{cases}$ $x_i \in [0, 2\pi$
Case 11	$\begin{cases} x_1 - 0.25428722 - 0.18324757x_3 x_4 x_9 = 0 \\ x_2 - 0.37842197 - 0.16275449x_1 x_6 x_1 0 = 0 \\ x_3 - 0.27162577 - 0.16955071x_1 x_2 x_1 0 = 0 \\ x_4 - 0.19807914 - 0.15585316x_1 x_6 x_7 = 0 \\ x_5 - 0.44166728 - 0.19950920x_3 x_6 x_7 = 0 \\ x_6 - 0.14654113 - 0.18922793x_5 x_8 x_1 0 = 0 \\ x_7 - 0.42937161 - 0.21180486x_2 x_5 x_8 = 0 \\ x_8 - 0.07056438 - 0.17081208x_1 x_6 x_7 = 0 \\ x_9 - 0.34504906 - 0.19612740x_6 x_8 x_1 0 = 0 \\ x_1 0 - 0.42651102 - 0.21466544x_1 x_4 x_8 = 0 \end{cases}$ $x_i \in [-2, 2]$
Case 12	$\begin{cases} x_i^2 + x_{i+1}^2 - 1 = 0 \\ a_{1i}x_1 x_3 + a_{2i}x_1 x_4 + a_{3i}x_2 x_3 + a_{4i}x_2 x_4 + a_{5i}x_2 x_7 + a_{6i}x_5 x_8 + a_{7i}x_6 x_7 + \\ a_{8i}x_6 x_8 + a_{9i}x_1 + a_{10i}x_2 + a_{11i}x_3 + a_{12i}x_4 + a_{13i}x_5 \\ +a_{14i}x_6 + a_{15i}x_7 + a_{16i}x_8 + a_{17i} = 0; \qquad 1 \leq i \leq 4 \end{cases}$ $x_i \in [-10, 10]$
Case 13	$\begin{cases} x_2 + 2x_6 + x_9 + 2x_1 0 - 10^(- 5) = 0 \\ x_3 + x_8 - 3 \times 10^(- 5) = 0 \\ x_1 + x_3 + 2x_5 + 2x_8 + x_9 + x_1 0 - 5 \times 10^(- 5) = 0 \\ x_4 + 2x_7 - 10^(- 5) = 0 \\ 0.5140437 \times 10^(- 7)x_5 - x_1^2 = 0 \\ 0.1006932 \times 10^(- 6)x_6 - 2x_2^2 = 0 \\ 0.7816278 \times 10^(- 15)x_7 - x_4^2 = 0 \\ 0.1496236 \times 10^(- 6)x_8 - x_1 x_3 = 0 \\ 0.6194411 \times 10^(- 7)x_9 - x_1 x_2 = 0 \\ 0.2089296 \times 10^(- 14)x_1 0 - x_1 x_2^2 = 0 \end{cases}$ $x_i \in [-20, 20]$

Table 7. Optimum results for Case 1.

Function	NDA	PSO	ICA	CS	ACO
x_1	12.2565196	-12.2565196	-12.2568621	-12.2565196	-12.2565206
x_2	22.8949380	-22.8949386	-22.9118650	-22.8949386	-22.8949932
x_3	2.7898183	-2.7898179	-2.7797279	-2.7898179	-2.7897853
f_1	1.6249E-05	-7.3129E-11	-3.8856E-01	0.0000E+00	-1.2556E-03
f_2	2.5032E-05	0.0000E+00	3.7464E-03	0.0000E+00	1.1628E-05
f_3	-2.4320E-05	1.8190E-12	-2.1183E-03	0.0000E+00	-1.0003E-05
F_{best}	1.4821E-09	5.3512E-21	1.5100E-01	0.0000E+00	1.5768E-06
F_{avg}	**3.0358E-04**	1.1259E+06	1.4411E+04	9.1972E-02	8.3196E+03
F_{worst}	**5.0167E-03**	2.8090E+07	5.0856E+04	9.1919E+00	2.8960E+04
Std	**8.5611E-04**	5.5041E+06	1.5115E+04	9.1457E-01	1.0200E+04
$Time(s)$	**1.941**	6.501	7.015	4.614	9.941

Table 8. Optimum results for Case 2.

Function	NDA	PSO	ICA	CS	ACO
x_1	4	4	4	4	4
x_2	3	3	3	3	3
x_3	1	1	1	1	1
f_1	0.0000E+00	0.0000E+00	-2.6816E-11	0.0000E+00	0.0000E+00
f_2	0.0000E+00	0.0000E+00	-5.06688E-11	0.0000E+00	0.0000E+00
f_3	0.0000E+00	0.0000E+00	3.8192E-09	0.0000E+00	0.0000E+00
F_{best}	**0.0000E+00**	**0.0000E+00**	1.4590E-17	**0.0000E+00**	**0.0000E+00**
F_{avg}	**0.0000E+00**	1.5639E+00	1.5273E-12	**0.0000E+00**	2.2598E-24
F_{worst}	**0.0000E+00**	7.8194E+00	3.5000E-11	**0.0000E+00**	7.2724E-24
Std	**0.0000E+00**	3.1278E+00	5.2979E-12	**0.0000E+00**	2.4547E-24
$Time(s)$	**2.762**	6.314	7.052	5.345	9.822

Table 9. Optimum results for Case 3.

Function	NDA	PSO	ICA	CS	ACO
x_1	-0.2905146	-0.2905146	-0.2905146	-0.2905146	-0.2905146
x_2	1.0842151	1.0842151	1.0842151	1.0842151	1.0842151
f_1	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
f_2	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
F_{best}	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
F_{avg}	9.8608E-32	6.7053E-32	5.1276E-32	**0.0000E+00**	9.0719E-32
F_{worst}	1.9722E-31	1.9722E-31	1.9722E-31	**0.0000E+00**	1.9722E-31
Std	9.8608E-32	9.3423E-32	8.6505E-32	**0.0000E+00**	9.8292E-32
$Time(s)$	**1.607**	6.184	8.599	4.415	7.839

TABLE 10. Optimum results for Case 4.

Function	NDA	PSO	ICA	CS	ACO
x_1	-0.9293683	0.9760155	-0.3971934	-0.9978328	0.8714294
x_2	-0.9293683	0.9760155	0.3971934	0.9978328	-0.8714294
x_3	0.3691539	-0.2177008	0.9177349	-0.0658011	0.4905209
x_4	0.3691539	-0.2177008	-0.9177349	0.06580113	-0.4905209
x_5	0.2791445	0.2568353	0.7090235	0.34639726	-0.1475619
x_6	-0.2791445	-0.2568353	0.7090235	0.34639726	-0.1475619
f_1	0.0000E+00	0.0000E+00	0.0000E+00	1.6735E-10	0.0000E+00
f_2	0.0000E+00	0.0000E+00	0.0000E+00	3.8574E-12	0.0000E+00
F_3	0.0000E+00	0.0000E+00	0.0000E+00	-2.3286E-12	0.0000E+00
F_4	0.0000E+00	0.0000E+00	0.0000E+00	-3.7086E-11	0.0000E+00
F_5	0.0000E+00	0.0000E+00	0.0000E+00	-2.3595E-11	0.0000E+00
F_6	0.0000E+00	0.0000E+00	0.0000E+00	-1.8012E-10	0.0000E+00
F_best	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	6.2402E-20	**0.0000E+00**
F_avg	1.2635E-10	**4.3568E-22**	1.6660E-06	1.4765E-09	3.3145E-09
F_{worst}	2.0113E-09	**2.1784E-20**	8.3233E-05	2.5806E-08	8.5111E-08
Std	3.9700E-10	**3.0497E-21**	1.1652E-05	3.8618E-09	1.4012E-08
$Time$	**2.227**	6.419	8.015	5.144	16.019

TABLE 11. Optimum results for Case 5.

Function	NDA	PSO	ICA	CS	ACO
x_1	0.5	0.5	0.5	0.5	0.5
x_2	3.141592654	3.141592654	3.141592654	3.141592654	3.141592654
f_1	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
f_2	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
F_{best}	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
F_{avg}	5.4696E-34	2.2279E-04	2.3533E-25	**4.6222E-35**	3.0614E-30
F_{worst}	**7.7037E-34**	7.4264E-04	1.1694E-23	**7.7037E-34**	4.1763E-29
Std	3.4957E-34	3.4032E-04	1.6370E-24	**1.8295E-34**	6.7191E-30
$Time(s)$	**2.250**	6.158	7.441	4.577	7.774

Table 12. Optimum results for Case 6.

Function	NDA	PSO	ICA	CS	ACO
x_1	0.6715543	0.6715543	0.1644317	0.6715543	0.6715543
x_2	0.7409554	0.7409554	-0.9863885	0.7409554	0.7409554
x_3	-0.6515906	-0.6515906	0.7184526	0.9518927	-0.6515906
x_4	-0.7585708	-0.7585708	-0.6955759	-0.3064314	-0.7585708
x_5	-0.9625450	-0.9625450	-0.9979644	-0.9638108	0.9625450
x_6	0.2711219	-0.2711219	0.0637737	0.2665873	-0.2711219
x_7	-0.4375776	-0.4375776	-0.5278091	0.4046414	-0.4375776
x_8	-0.8991807	0.8991807	-0.8493630	-0.9144754	0.8991807
f_1	0.0000E+00	-1.1102E-16	1.1102E-16	0.0000E+00	0.0000E+00
f_2	0.0000E+00	1.1102E-16	-1.1102E-16	0.0000E+00	0.0000E+00
f_3	-3.9031E-18	-3.1659E-17	-4.3368E-18	-3.9031E-18	-3.1659E-17
f_4	-5.5511E-17	-5.5511E-17	0.0000E+00	-5.5511E-17	-5.5511E-17
f_5	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
f_6	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
f_7	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
f_8	0.0000E+00	2.2204E-16	0.0000E+00	0.0000E+00	0.0000E+00
F_{best}	**3.0967E-33**	7.8039E-32	2.4671E-32	**3.0967E-33**	4.0838E-33
F_{avg}	**5.8616E-33**	7.6967E-02	1.2089E-02	2.5392E-12	7.1887E-32
F_{worst}	**1.8504E-32**	3.4509E-01	2.0184E-01	8.3763E-11	4.2529E-31
Std	3.8990E-33	9.6374E-02	4.7848E-02	1.2158E-11	9.6176E-32
$Time(s)$	**2.365**	6.233	7.900	5.189	19.890

TABLE 13. Optimum results for Case 7.

Function	NDA	PSO	ICA	CS	ACO
x_1	0.5000000	0.5000000	0.5000000	0.5000000	0.5000000
x_2	0.0000001	0.0000000	0.0000002	0.0000000	0.0000002
x_3	-0.5235988	-0.5235988	-0.5235988	-0.5235988	-0.5235988
f_1	1.2355E-12	1.0891E-13	8.7710E-12	0.0000E+00	1.1545E-11
f_2	-3.7081E-12	-3.286E-13	-2.6312E-11	0.0000E+00	-3.4637E-11
f_3	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
F_{best}	1.5276E-23	1.1984E-25	7.6926E-22	**0.0000E+00**	1.3330E-21
F_{avg}	2.8038E-13	1.8994E-20	3.3658E-14	**0.0000E+00**	1.0887E-15
F_{worst}	2.3105E-12	9.3809E-20	1.9288E-13	0.0000E+00	2.7747E-15
Std	5.3773E-13	2.2653E-20	3.9585E-14	**0.0000E+00**	1.0709E-15
$Time(s)$	**1.721**	6.524	7.779	4.834	9.741

TABLE 14. Optimum results for Case 8.

Function	NDA	PSO	ICA	CS	ACO
x_1	0.9163546	1.0000000	0.9162502	0.9163546	0.9163546
x_2	0.9163546	1.0000000	0.9162415	0.9163546	0.9163546
x_3	0.9163546	1.0000000	0.9162459	0.9163546	0.9163546
x_4	0.9163546	1.0000000	0.9162426	0.9163546	0.9163546
x_5	1.4182271	1.0000000	1.4187915	1.4182271	1.4182271
f_1	0.0000E+00	8.8818E-16	2.1927E-05	0.0000E+00	0.0000E+00
f_2	0.0000E+00	0.0000E+00	1.3164E-05	0.0000E+00	0.0000E+00
f_3	0.0000E+00	8.8818E-16	1.7560E-05	0.0000E+00	0.0000E+00
f_4	0.0000E+00	0.0000E+00	1.4293E-05	0.0000E+00	0.0000E+00
f_5	0.0000E+00	-2.6645E-15	-8.0265E-05	0.0000E+00	-4.4409E-16
F_{best}	**0.0000E+00**	8.6775E-30	7.6092E-09	**0.0000E+00**	1.9722E-31
F_{avg}	**0.0000E+00**	2.8552E-02	1.2504E-05	**0.0000E+00**	1.1581E-29
F_{worst}	**0.0000E+00**	1.2602E+00	3.1293E-04	**0.0000E+00**	8.2547E-29
Std	**0.0000E+00**	1.7619E-01	4.8082E-05	**0.0000E+00**	1.8148E-29
Time	**1.827**	6.536	7.760	5.138	14.157

Table 15. Optimum results for Case 9.

Function	NDA	PSO	ICA	CS	ACO
x_1	-1	-0.9998809	-0.9999280	-1	-1.0000000
x_2	1	0.9999070	0.9999467	1	1.0000000
x_3	-1	-1.0001231	-1.0000845	-1	-1.0000000
x_4	1	0.9998398	0.9998892	1	1.0000000
x_5	-1	-1.0000682	-1.0000405	-1	-1.0000000
x_6	1	0.9999413	0.9999627	1	1.0000000
f_1	0.0000E+00	1.7928E-05	8.3776E-06	0.0000E+00	-7.8604E-14
f_2	0.0000E+00	-7.0790E-06	-2.5291E-06	0.0000E+00	2.9754E-14
f_3	0.0000E+00	-1.3648E-05	-1.039E-05	0.0000E+00	1.128E-13
f_4	0.0000E+00	-1.1259E-05	-8.652E-06	0.0000E+00	7.8604E-14
f_5	0.0000E+00	7.6686E-06	4.8038E-06	0.0000E+00	-6.217E-14
f_6	0.0000E+00	-7.7441E-06	-5.717E-06	0.0000E+00	6.4837E-14
F_{best}	**0.0000E+00**	8.0336E-10	3.1524E-10	**0.0000E+00**	3.4035E-26
F_{avg}	**0.0000E+00**	2.0635E-02	1.7433E-03	**0.0000E+00**	7.6415E-23
F_{worst}	**0.0000E+00**	3.9144E-01	8.2054E-02	**0.0000E+00**	1.6305E-21
Std	**0.0000E+00**	6.0278E-02	1.1475E-02	**0.0000E+00**	2.2360E-22
Time	**1.810**	6.527	7.808	5.170	15.705

Table 16. Optimum results for Case 10.

Function	NDA	PSO	ICA	CS	ACO
x_1	0.5149333	0.5149333	0.5149333	0.5149333	0.5149333
x_2	0.5149333	0.5149333	0.5149333	0.5149333	0.5149333
x_3	0.5149333	0.5149333	0.5149333	0.5149333	0.5149333
x_4	0.5149333	0.5149333	0.5149333	0.5149333	0.5149333
f_1	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
f_2	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
f_3	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
f_4	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
F_{best}	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
F_{avg}	**0.0000E+00**	4.7051E-01	4.1454E-01	4.3449E-33	3.2047E-33
F_{worst}	**0.0000E+00**	1.0363E+01	1.0363E+01	1.2326E-32	1.2326E-32
Std	**0.0000E+00**	2.0196E+00	2.0308E+00	5.1396E-33	5.1916E-33
Time	**1.807**	6.261	7.616	5.157	12.016

TABLE 17. Optimum results for Case 11.

Function	NDA	PSO	ICA	CS	ACO
x_1	0.2578348	0.2578348	0.2578348	0.2578348	0.2578348
x_2	0.3810972	0.3810972	0.3810972	0.3810972	0.3810972
x_3	0.2787451	0.2787451	0.2787451	0.2787451	0.2787451
x_4	0.2007453	0.2007453	0.2007453	0.2007453	0.2007453
x_5	0.4453571	0.4453571	0.4453571	0.4453571	0.4453571
x_6	0.1491876	0.1491876	0.1491876	0.1491876	0.1491876
x_7	0.4447334	0.4447334	0.4447334	0.4447334	0.4447334
x_8	0.0734865	0.0734865	0.0734865	0.0734865	0.0734865
x_9	0.3459679	0.3459679	0.3459679	0.3459679	0.3459679
x_10	0.4273275	0.4273275	0.4273275	0.4273275	0.4273275
f_1	2.4720E-17	2.4720E-17	2.4720E-17	2.4720E-17	2.4720E-17
f_2	-2.1250E-17	-2.1250E-17	-2.1250E-17	-2.1250E-17	-2.1250E-17
f_3	1.4745E-17	1.4745E-17	1.4745E-17	1.4745E-17	1.4745E-17
f_4	7.3726E-18	7.3726E-18	7.3726E-18	7.3726E-18	7.3726E-18
f_5	2.0383E-17	2.0383E-17	2.0383E-17	2.0383E-17	2.0383E-17
f_6	4.3368E-19	4.3368E-19	4.3368E-19	4.3368E-19	4.3368E-19
f_7	1.2143E-17	1.2143E-17	1.2143E-17	1.2143E-17	1.2143E-17
f_8	8.6736E-19	8.6736E-19	8.6736E-19	8.6736E-19	8.6736E-19
f_9	-7.5894E-18	-7.5894E-18	-7.5894E-18	-7.5894E-18	-7.5894E-18
f_10	1.7781E-17	1.7781E-17	1.7781E-17	1.7781E-17	1.7781E-17
F_{best}	**2.2720E-33**	**2.2720E-33**	**2.2720E-33**	**2.2720E-33**	**2.2720E-33**
F_{avg}	**2.2720E-33**	3.2457E-14	**2.2720E-33**	**2.2720E-33**	2.7632E-33
F_{worst}	**2.2720E-33**	1.6202E-12	**2.2720E-33**	**2.2720E-33**	7.5772E-33
Std	**0.0000E+00**	2.2683E-13	**0.0000E+00**	4.7896E-48	9.1794E-34
$Time(s)$	**1.388**	6.594	7.713	5.784	23.556

TABLE 18. Parameters for Case 12.

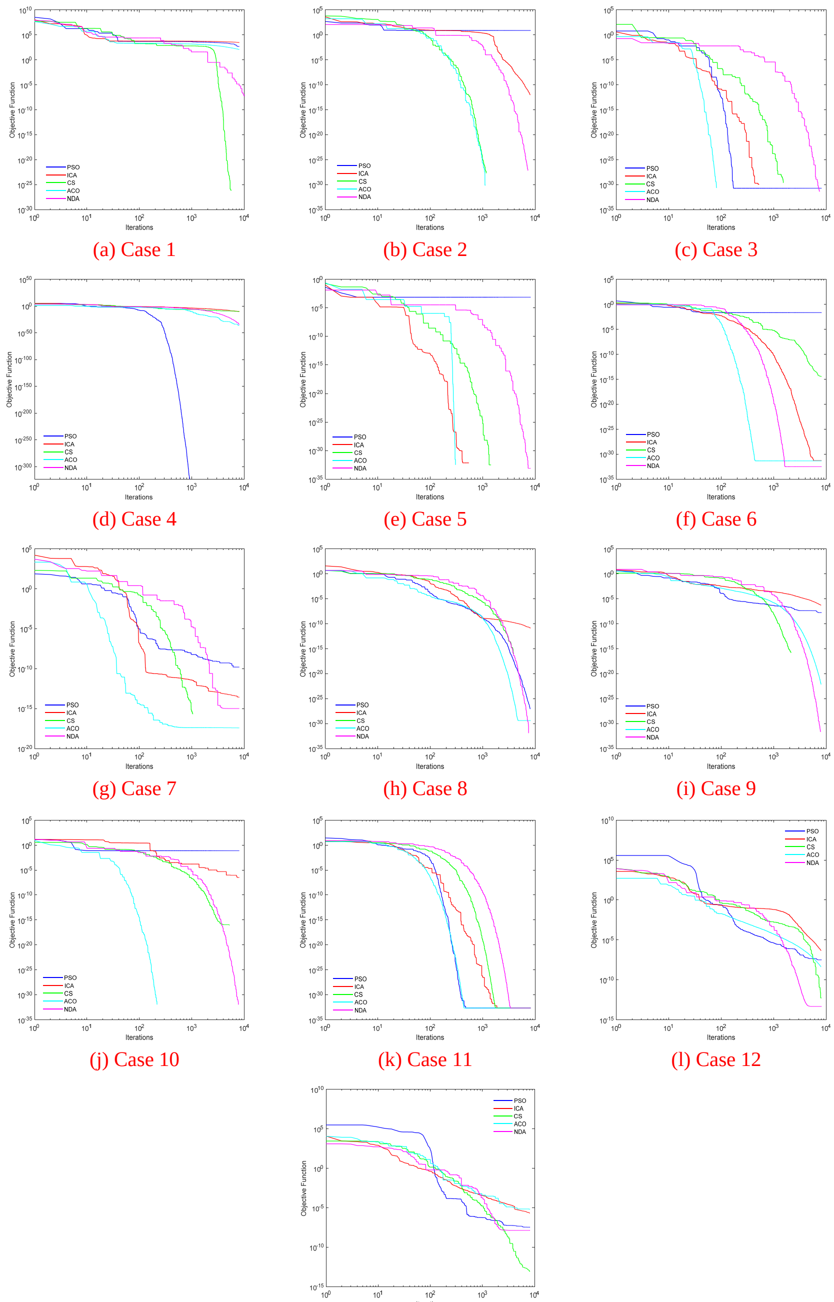
a_{ji}	1	2	3	4
1	-0.249150680	0.125016350	-0.635550070	1.489477300
2	1.609135400	-0.686607360	-0.115719920	0.230623410
3	0.279423430	-0.119228120	-0.666404480	1.328107300
4	1.434480160	-0.719940470	0.110362110	-0.258645030
5	0.000000000	-0.432419270	0.290702030	1.165172000
6	0.400263840	0.000000000	1.258776700	-0.269084940
7	-0.800527680	0.000000000	-0.629388360	0.538169870
8	0.000000000	-0.864838550	0.581404060	0.582585980
9	0.074052388	-0.037157270	0.195946620	-0.208169850
10	-0.083050031	0.035436896	-1.228034200	2.686832000
11	-0.386159610	0.085383482	0.000000000	-0.699103170
12	-0.755266030	0.000000000	-0.079034221	0.357444130
13	0.504201680	-0.039251967	0.026387877	1.249911700
14	-1.091628700	0.000000000	-0.057131430	1.467736000
15	0.000000000	-0.432419270	-1.162808100	1.165172000
16	0.049207290	0.000000000	1.258776700	1.076339700
17	0.049220729	0.013873010	2.162575000	-0.696868090

Table 19. Optimum results for Case 12.

Function	NDA	PSO	ICA	CS	ACO
x_1	-0.9290548	-0.7990614	-0.7990614	0.9945078	0.7729415
x_2	-0.3699422	-0.6012494	-0.6012494	0.1046625	0.6344772
x_3	-0.9290548	-0.7990614	-0.7990614	0.9945078	0.7729415
x_4	0.3699422	0.6012494	0.6012495	0.1046625	0.6344772
x_5	-0.9290548	0.7990614	0.7990613	0.9945078	-0.7729415
x_6	-0.5257808	-0.6786384	-0.6786383	-0.3843076	0.5002711
x_7	1.7263456	0.9436221	0.9436231	-0.4832660	0.0187910
x_8	0.1681780	-0.6348997	-0.6348993	-0.8702357	-1.2010347
f_1	0.0000E+00	-3.9968E-15	-3.2826E-08	0.0000E+00	-5.5511E-16
f_2	0.0000E+00	7.2387E-14	-9.5577E-08	0.0000E+00	2.2204E-16
f_3	0.0000E+00	-4.74065E-14	9.2267E-08	0.0000E+00	2.2204E-16
f_4	0.0000E+00	-4.74065E-14	9.2267E-08	0.0000E+00	2.2204E-16
f_5	1.3878E-17	-6.87853E-14	7.9699E-08	1.1102E-16	5.0654E-16
f_6	3.6429E-17	-1.43349E-13	2.5228E-07	8.6736E-18	5.6205E-16
f_7	0.0000E+00	-8.88178E-15	-6.3999E-08	0.0000E+00	0.0000E+00
f_8	0.0000E+00	9.99201E-16	-6.0876E-09	0.0000E+00	3.3307E-16
F_{best}	**1.5197E-33**	3.2902E-26	9.3101E-14	1.2401E-32	1.1025E-30
F_{avg}	1.3411E-05	4.3893E-03	5.1761E-03	**3.7631E-06**	5.8562E-06
F_{worst}	8.2093E-05	6.6469E-02	6.6469E-02	**5.8180E-05**	1.0649E-04
Std	2.2097E-05	1.4497E-02	1.5864E-02	**1.1106E-05**	1.9067E-05
$Time(s)$	**2.886**	7.334	8.505	6.041	22.938

Table 20. Optimum results for Case 13.

Function	NDA	PSO	ICA	CS	ACO
x_1	-0.0000001	-0.0000001	-0.0000002	0.0006745	0.0000190
x_2	-0.0016248	0.0008110	0.0194582	-0.0008281	0.0030550
x_3	0.2888647	3.1439916	-9.2054784	-0.0000013	-0.7771021
x_4	0.0009906	0.0001748	0.0053118	0.0000532	-0.0036210
x_5	-1.8858060	0.4189938	-13.7403258	9.0550510	-15.2917490
x_6	-5.2386197	-2.7210099	-12.0499629	13.6325414	-16.7629329
x_7	-0.0004903	-0.0000824	-0.0026505	-0.0000216	0.0017389
x_8	-0.2888347	-3.1439615	9.2055024	0.0000313	0.7771094
x_9	-2.3579406	-0.8292310	12.4699675	-8.9573310	6.0898931
x_10	6.4184074	3.1352249	5.8052212	-9.1534568	3.7165353
f_1	4.1902E-09	2.2586E-08	-6.7834E-05	2.4652E-09	1.4307E-04
f_2	1.3347E-09	3.6900E-08	-5.9185E-06	2.2491E-09	-2.2668E-05
f_3	-1.7409E-09	-7.6660E-15	1.3434E-05	3.3685E-09	1.6114E-05
f_4	-8.5102E-10	-2.7908E-08	7.8414E-07	1.3236E-08	-1.5311E-04
f_5	-9.6939E-08	2.1538E-08	-7.0631E-07	1.0480E-08	-7.8642E-07
f_6	-5.8072E-06	-1.5896E-06	-7.5846E-04	1.3315E-09	-2.0355E-05
f_7	-9.8127E-07	-3.0563E-08	-2.8215E-05	-2.8268E-09	-1.3112E-05
f_8	-7.7326E-09	-4.2211E-08	-3.4552E-07	8.5542E-10	1.4875E-05
f_9	-1.4626E-07	-5.1256E-08	7.7608E-07	3.6977E-09	1.5581E-06
f_10	3.3769E-13	9.6138E-14	7.0874E-11	-4.6253E-10	-1.7725E-10
F_{best}	3.4717E-11	2.5351E-12	5.8087E-07	**3.3190E-16**	4.5496E-08
F_{avg}	9.4762E-07	2.1991E-08	2.9448E-06	**2.3173E-12**	1.0729E-04
F_{worst}	6.0118E-06	2.1562E-07	5.8016E-06	**1.7253E-10**	2.0885E-03
Std	1.3443E-06	4.3751E-08	1.0157E-06	**1.7327E-11**	3.6690E-04
$Time(s)$	2.143	6.404	7.937	**5.362**	23.475



(a) Case 1 (b) Case 2 (c) Case 3
(d) Case 4 (e) Case 5 (f) Case 6
(g) Case 7 (h) Case 8 (i) Case 9
(j) Case 10 (k) Case 11 (l) Case 12
(m) Case 13

FIGURE 7. Comparision of the convergence of PSO, ICA, CS, ACO and NDA for cases studies 1-13.

5. Conclusions

So far, a wide range of various mathematical and evolutionary methods has been used to solve the system of nonlinear equations. In this study, a completely new statistical approach has been developed for solving the systems of nonlinear equations. This approach utilizes the benefits of the normal distribution, one of the most important statistical distributions, to find the optimal solution. Some benchmark tests and case problems have been used to compare the obtained results of NDA with those of PSO, ICA, CS, and ACO. These comparisons clearly demonstrate the advantages of NDA. The advantages include (1) NDA requires relatively few parameters, i.e., σ_g, k, and p, to make a successful and smart search. As a consequence, it helps the user to tune the algorithms parameters easily. (2) NDA is an easy to understand algorithm with a simple structure that facilitates the coding procedure and computation. (3) Due to its simplicity, NDA is suggested as a high-speed approach for solving large scale problems. In other words, it gets high-quality solutions in a reasonable amount of time for practical size problems. (4) NDA is an efficient algorithm for solving nonlinear functions. The effectiveness of NDA is because it uses a contraction expansion coefficient, i.e., standard deviation, to balance the local and global searches during the optimization process.

Besides all mentioned advantages, future work is needed to improve NDA for solving other mathematical programming problems such as linear programming and constrained nonlinear programming problems. Furthermore, other statistical distributions can be used instead of the normal distribution to solve the system of nonlinear equations.

Compliance with ethical standards.

Conflict of interest the author declares that they have no conflict of interests.

References

[1] A. Abdelhalim, K. Nakata, M. El-Alem, and A. Eltawil, *A hybrid evolutionary-simplex search method to solve nonlinear constrained optimization problems*, Soft Computing, *23*(22) (2019), 12001-12015, Doi: 10.1080/0305215x.2017.1340945.

[2] M. Abdollahi, A. Bouyer, and D. Abdollahi, *Improved cuckoo optimization algorithm for solving systems of nonlinear equations*, The Journal of Supercomputing, *72*(3) (2016), 1246-1269, Doi:10.1007/s11227-016-1660-8.

[3] M. Abdollahi, A. Isazadeh, and D. Abdollahi, *Imperialist competitive algorithm for solving systems of nonlinear equations*, Computers and Mathematics with Applications, *65*(12) (2013), 1894-1908, Doi:10.1016/j.camwa.2013.04.018.

[4] E. Atashpaz-Gargari and C. Lucas, *Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition*, IEEE congress on evolutionary computation, (2007), 4661-4667, Doi: 10.1109/CEC.2007.4425083.

[5] X. Chen and C. T. Kelley, *Convergence of the EDIIS algorithm for nonlinear equations*, SIAM Journal on Scientific Computing, *41*(1) (2019), A365-A379, Doi: doi.org/10.1137/18M1171084.

[6] I.M. El-Emary and M.A. El-Kareem, *Towards using genetic algorithm for solving nonlinear equation systems*, World Applied Sciences Journal, *5*(3) (2008), 282-289.

[7] C. Grosan and A. Abraham, *A new approach for solving nonlinear equations systems*, IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, *38*(3) (2008), 698-714, Doi: 10.1109/TSMCA.2008.918599.

[8] M.J. Hirsch, P.M. Pardalos, and M. G. Resende, *Solving systems of nonlinear equations with continuous GRASP*, Nonlinear Analysis: Real World Applications, *10*(4) (2009), 2000-2006, Doi: 10.1016/j.nonrwa.2008.03.006.

[9] S. Hosseini and A. Al Khaled, *A survey on the imperialist competitive algorithm metaheuristic: implementation in engineering domain and directions for future research*, Applied Soft Computing, *24* (2014), 1078-1094, Doi:10.1016/j.asoc.2014.08.024.

[10] A. M. Ibrahim and M. A. Tawhid, *A hybridization of differential evolution and monarch butterfly optimization for solving systems of nonlinear equations*, Journal of Computational Design and Engineering, *6*(3) (2019a), 354-367, Doi:10.1016/j.jcde.2018.10.006.

[11] A. M. Ibrahim and M. A. Tawhid, *A hybridization of cuckoo search and particle swarm optimization for solving nonlinear systems*, Evolutionary Intelligence, *12*(4) (2019b), 541-561, Doi:10.1016/j.camwa.2006.12.081.

[12] A. F. Izmailov, A. S Kurennoy, and M. V. Solodov, *Critical solutions of nonlinear equations: stability issues*, Mathematical Programming, (2016), 1-33, Doi: 10.1007/s10107-016-1047-x.

[13] M. Jaberipour, E. Khorram, and B. Karimi, *Particle swarm algorithm for solving systems of nonlinear equations*, Computers and Mathematics with Applications, *62*(2) (2011), 566-576, Doi: 10.1016/j.camwa.2011.05.031.

[14] M. Juneja and S. K. Nagar, *Particle swarm optimization algorithm and its parameters: A review*, International Conference on Control, Computing, Communication and Materials (ICCCCM), IEEE, (2016), 1-5, Doi: 10.1109/ICCCCM.2016.7918233.

[15] JA. Koupaei and S. Hosseini, *A new hybrid algorithm based on chaotic maps for solving systems of nonlinear equations*, Chaos Solitons Fractals, *81* (2015), 233245, Doi: 10.1016/j.chaos.2015.09.027

[16] G. Li and Z. Zeng, *A neural-network algorithm for solving nonlinear equation systems*, International Conference on Computational Intelligence and Security (IEEE), *1* (2008), 20-23, Doi: 10.1109/CIS.2008.65.

[17] MD. Li, H. Zhao, XW. Weng, and T. Han, *A novel nature-inspired algorithm for optimization: virus colony search*, Advances in Engineering Software, *92* (2016), 6588, Doi: 10.1016/j.advengsoft.2015.11.004.

[18] L. Liu, M. X. Liu, N. Wang, and P. Zou, *Modified cuckoo search algorithm with variational parameters and logistic map*, Algorithms, *11*(3) (2018), 30, Doi: 10.3390/a11030030.

[19] M.A. Luersen and R. Le Riche, *Globalized NelderMead method for engineering optimization*, Computers and structures, *82*(23) (2004), 2251-2260, Doi: 10.1016/j.compstruc.2004.03.072.

[20] A. Majd, M. Abdollahi, G. Sahebi, D. Abdollahi, M. Daneshtalab, J. Plosila, and H. Tenhunen, *Multi-population parallel imperialist competitive algorithm for solving systems of nonlinear equations*, International Conference on High Performance Computing and Simulation (HPCS), IEEE, (2016), 767-775, Doi:10.1109/HPCSim.2016.7568412.

[21] A. Majd, G. Sahebi, M. Daneshtalab, J. Plosila, S. Lotfi, and H. Tenhunen, *Parallel imperialist competitive algorithms*, Concurrency and Computation: Practice and Experience, *30*(7) (2018), e4393, Doi: 10.1002/cpe.4393.

[22] E.L. Melnick and A. Tenenbein, *Misspecifications of the normal distribution*, The American Statistician, *36*(4) (1982), 372-373.

[23] Y. Mo, H. Liu, and Q. Wang, *Conjugate direction particle swarm optimization solving systems of nonlinear equations*, Computers and Mathematics with Applications, *57*(11) (2009), 1877-1882, Doi: 10.1016/j.camwa.2008.10.005.

[24] H. Mhlenbein, M. Schomisch, and J. Born, *The parallel genetic algorithm as function optimizer*, Parallel computing, *17*(6-7) (1991), 619-632, Doi:10.1016/S0167-8191(05)80052-3.

[25] H.A. Oliveira and A. Petraglia, *Solving nonlinear systems of functional equations with fuzzy adaptive simulated annealing*, Applied Soft Computing, *13*(11) (2013), 4349-4357, Doi: 10.1016/j.asoc.2013.06.018.

[26] J. Pei, Z. Drai, M. Drai, N. Mladenovi, and P.M. Pardalos, *Continuous variable neighborhood search (C-VNS) for solving systems of nonlinear equations*, INFORMS Journal on Computing, *31*(2) (2019), 235-250, Doi: 10.1287/ijoc.2018.0876.

[27] E. Pourjafari and H. Mojallali, *Solving nonlinear equations systems with a new approach based on invasive weed optimization algorithm and clustering*, Swarm and Evolutionary Computation, *4* (2012), 33-43, Doi: 10.1016/j.swevo.2011.12.001.

[28] M. A. Z. Raja, M. A. Zameer, A.K. Kiani, A. Shehzad, and M. A. R. Khan, *Nature-inspired computational intelligence integration with NelderMead method to solve nonlinear benchmark models*, Neural Computing and Applications, *29*(4) (2018), 1169-1193, Doi: 10.1007/s00521-016-2523-1.

[29] J. R. Sharma, I. K. Argyros, and D. Kumar, *On a general class of optimal order multipoint methods for solving nonlinear equations*, Journal of Mathematical Analysis and Applications, (2016), Doi: 10.1016/j.jmaa.2016.12.051.

[30] J. R. Sharma and H. Arora, *On efficient weighted-Newton methods for solving systems of nonlinear equations*, Applied Mathematics and Computation, *222* (2013), 497-506, Doi: 10.1016/j.amc.2013.07.066.

[31] M. Shehab, A.T. Khader, and M. A. Al-Betar, *A survey on applications and variants of the cuckoo search algorithm*, Applied Soft Computing, *61* (2017), 1041-1059, Doi: 10.1016/j.asoc.2017.02.034.

[32] N. Singh and S. B. Singh, *Hybrid algorithm of particle swarm optimization and grey wolf optimizer for improving convergence performance*, Journal of Applied Mathematics, (2017), Doi: 10.1016/j.asoc.2017.02.034.

[33] OE. Turgut, MS. Turgut, and MT. Coban, *Chaotic quantum behaved particle swarm optimization algorithm for solving nonlinear system of equations*, Computers and Mathematics with Applications, *68*(4) (2014), 508530, Doi: 10.1016/j.camwa.2014.06.013.

[34] C. Wang, R. Luo, K. Wu, and B. Han, *A new filled function method for an unconstrained nonlinear equation*, Journal of Computational and Applied Mathematics, *235*(6) (2011), 1689-1699, Doi: 10.1016/j.cam.2010.09.010.

[35] X. Wang and N. Zhou, *Pattern Search Firefly Algorithm for Solving Systems of Nonlinear Equations*, Seventh International Symposium on Computational Intelligence and Design, IEEE, *2* (2014), 228-231, Doi: 10.1109/ISCID.2014.222.

[36] X. S. Yang and S. Deb, *Cuckoo search via Lvy flights*, World congress on nature and biologically inspired computing (NaBIC), IEEE, (2009), 210-214, Doi: 10.1109/NABIC.2009.5393690.

[37] X. Zhang, Q. Wan, and Y. Fan, *Applying modified cuckoo search algorithm for solving systems of nonlinear equations*, Neural Computing and Applications, *31*(2) (2019), 553-576, Doi: 10.1007/s00521-017-3088-3.