Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.18576/amis/110201

Conjugate Direction DE Algorithm for Solving Systems of Nonlinear Equations

Abdelmonem M. Ibrahim [1,2] *and Mohamed A. Tawhid* [3,4,*]

[1] Department of Mathematics, Faculty of Science, Al-Azhar University, Assiut Branch, Assiut, Egypt
[2] Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC, Canada V6T 1Z4
[3] Department of Mathematics and Statistics, Faculty of Science, Thompson Rivers University, Kamloops, BC, Canada V2C 0C8
[4] Department of Mathematics and Computer Science, Faculty of Science, Alexandria University, Moharam Bey 21511, Alexandria, Egypt

Received: 2 Sep. 2016, Revised: 19 Nov. 2016, Accepted: 22 Nov. 2016
Published online: 1 Mar. 2017

Abstract: In this paper, Differential Evolution with Powell conjugate direction method (DE-Powell) is proposed in order solve a system of nonlinear equations. A given system of nonlinear equations is formulated as an unconstrained optimization problem. Integrating Powell conjugate direction method into DE improves the performance of DE and enables DE to optimize effectively the system of nonlinear equations. For example, applying DE to solve our formulation of the system of nonlinear equations, in some iterations DE may get trapped in local minima, then Powell conjugate direction method is applied to help DE to overcome local minima by changing the initial solution for Powell with best obtained one by DE. Our proposed algorithm, DE-Powell, has superiority over Powell Conjugate Direction (CD) and Differential Evolution (DE), separately, it it overcomes the inaccuracy of Powell conjugate direction method and DE for solving systems of nonlinear equations. The DE-Powell is tested on nine well known problems and our numerical results show that the proposed algorithm is solving the highly nonlinear problems effectively and outperforms over many algorithms in literature.

Keywords: Optimization methods, Metaheuristics, Nonlinear system of equations, Powell conjugate direction method, Differential evolution.

1 Introduction

Consider the system of nonlinear equations, that there are n variables and m nonlinear equations system

$$\begin{aligned} f_1(x_1,x_2,x_3,\ldots,x_n) &= 0 \\ f_2(x_1,x_2,x_3,\ldots,x_n) &= 0 \\ &\vdots \\ f_n(x_1,x_2,x_3,\ldots,x_n) &= 0 \end{aligned} \tag{1}$$

where each function f_i maps a vector $X = (x_1,x_2,\cdots,x_n)$ of the n -dimensional space R^n into the real line R .

Many applications in sciences, engineering, economics, information security, dynamics and so on can be formulated as a system of nonlinear equations [9], [4]. Solving systems of nonlinear equations is considered one of the difficult problems in numerical computation. For most numerical methods such as the Newton' s method for solving systems of nonlinear equations, their convergence and performance may be sensitive to the initial solutions applied to the methods. Nonetheless, it is hard to select a good initial point for most systems of nonlinear equations. Many researchers have been interested in solving nonlinear equations and developed several kinds of methods that help to find the optimal solutions for these problems.

Some algorithms suggest a novel technique for solving nonlinear systems of equations by converting the problem into a global optimization problem.

Classical optimization techniques can be classified to two types: gradient search method and direct search method. In the gradient search method, the first order and/or second order derivatives are used for the search process, whereas the direct search method, only the objective function and constraints are used for the search process. Direct search methods require many function

* Corresponding author e-mail: Mtawhid@tru.ca

iterations which cause slowness of these methods, whereas gradient search methods are faster, but they are inadequate for discontinuous and non-differentiable functions. Furthermore, both methods search local optima, thus starting the search in the closeness of a local optima causes them to miss the global optima [32].

Metaheuristic algorithms overcome some of the afore-mentioned difficulties and are quickly replacing the classical methods in solving practical optimisation problems. Metaheuristic algorithms typically intend to find a good solution to an optimisation problem by trial-and-error in a reasonable amount of computational time. During the last few decades, several metaheuristic algorithms have been proposed. These algorithms include Particle Swarm Optimization Algorithm, Genetic Programming, Evolutionary Programming, Evolutionary Strategies, Genetic Algorithms, Differential Evolution, Harmony Search algorithm, Ant Colony Optimisation, Particle Swarm Optimisation, and Bee Algorithms [27, 33].

Hybridization of algorithms is the one of success techniques that utilities to solve the nonlinear equations systems [11, 17, 34].

In this work, the equations system (1) is transformed into an optimization problem as follows.

$$\min \Psi(X) = \sqrt{\sum_{i=1}^{m} f_i^2(X)}. \tag{2}$$

Differential evolution with Powell conjugate direction method (DE-Powell) is proposed to solve nonlinear system as an optimization problem. DE-Powell is developed in which the advantages of Powell Conjugate Direction method and Differential Evolution (DE) are combined. The numerical computations show that DE-Powell could overcome the problem of Powell and DE of getting easily trapping into local minima. The method resulted gives accurate solution when it applies to solve systems of nonlinear equations. The DE-Powell is tested on nine well known problems and our numerical results show that the proposed algorithm is solving the highly nonlinear problems effectively and outperforms over many algorithms in literature such as Chaotic Quantum Particle Swarm Optimization (LQPSO) [28], Intelligent Tuned Harmony Search algorithm (ITHS) [32], Imperialist Competitive Algorithm (ICA) [1], Quantum behaved Particle Swarm Optimization (QPSO) [25], [26], multiobjective approach for nonlinear systems [4], fuzzy adaptive simulated annealing [18], Weighted-Newton method [23], Combined method based on Grobner bases [8], and filled function method [31], Gravitational Search Algorithm (GRAV) [22].

.

The paper is organized as follows: Section 2 outlines the differential evaluation algorithm and describes the conjugate direction method. Section 3 presents the proposed algorithm, DE-Powell. Sections 4 and 5 show the performance of DE-Powell on test problems nine case studies of systems of nonlinear equations and compare with other existing algorithms in the literature. Finally Section 6 gives the conclusions and future work.

2 Related works

In this section, some of the optimization algorithms such as differential evolution algorithm (DE) and Powell conjugate direction method are discussed which are use din the literature

2.1 The differential evolution algorithm

Storn and Price [24] proposed the differential evolution (DE) algorithm in order to solve complex continuous nonlinear functions. DE is known that it is a simple powerful evolutionary algorithm for various global optimization problems.

The classical DE algorithm starts with initializing a population of NP, target individuals $P^t = \{X_1^t, X_2^t, \ldots, X_{NP}^t\}$, where t denotes the current iteration, individual $X_i^t = (x_{i,1}^t, x_{i,2}^t, \ldots, x_{i,n}^t)$, $i = 1, 2, \ldots, NP$, is an n-dimensional vector with parameter values determined randomly and uniformly between predefined search ranges $[X_{min}, X_{max}]$, where $X_{min} = (x_{min,1}, x_{min,2}, \ldots, x_{min,n})$ and $X_{max} = (x_{max,1}, x_{max,2}, \ldots, x_{max,n})$. Then mutation and crossover operators are utilized to generate new candidate vectors, and a selection scheme is applied to determine whether the offspring or the parent survives to the next generation [2]. The above process is repeated until a termination criterion is reached.

2.1.1 Mutation

A mutant individual, denoted by $V_i^t = (v_{i,1}^t, v_{i,2}^t, \ldots, v_{i,n}^t)$, $i = 1, 2, \ldots, NP$, is generated by using a mutation operator. There are many mutation strategies in the literature [19]. Among them, the commonly used operator is 'DE/rand/1', which is described as

$$V_i^t = X_a^t + DW(X_b^t - X_c^t) \tag{3}$$

where a, b and c are three randomly chosen indices in the range $[1, NP]$ such that a, b, c and i are pairwise different $(a \neq b \neq c \neq i \in 1, \ldots, NP)$. $DW \in [0, 1]$ is a mutation scaling factor which affects the differential variation between two individuals.

2.1.2 Crossover

After the mutation phase, a crossover operator is applied to each mutant individual and its corresponding target individual to yield a trial vector, $U_i^t = (u_{i,1}^t, u_{i,2}^t, \ldots, u_{i,n}^t)$.

Binomial and exponential crossovers are two commonly used as crossover schemes [21]. The binomial crossover is represented as follows:

$$u_{i,j}^{t} = \begin{cases} v_{i,j}^{t}, & \text{if } r_j \le CR \text{ or } j = R; \\ x_{i,j}^{t}, & \text{otherwise.} \end{cases} \quad (4)$$

where the index R refers to a randomly chosen dimension in the set $1,2,\dots,n$, which is used to ensure that at least one dimension of the trial individual, $U_{i,j}^{t}$, differs from its target vector, X_i^t. CR is a crossover rate in the range $[0,1]$, and $r_j \in [0,1]$ is a uniform random number. If the parameter values of the obtained trial individuals exceed the pre-specified upper bound or lower bound, we can set them equal to upper bound or lower bound, respectively.

2.1.3 Selection

In order to decide whether or not the trial individual U_i^t should become a member of the target population in the next generation, a one-to-one greedy selection between a parent and its corresponding offspring is employed in DE. This strategy enhances diversity in comparison to other selection strategies such as tournament selection, rank based selection and fitness proportional selection. The one-to-one selection scheme is based on the survival of the fitter between the trial individual U_i^t and its target counterpart X_i^t. For minimization problem, it can be defined as follows [21]:

$$X_i^{t+1} = \begin{cases} U_i^t, & \text{if } F(U_i^t) \le F(X_i^t); \\ X_i^t, & \text{otherwise.} \end{cases} \quad (5)$$

where $F(U_i^t)$ and $F(X_i^t)$ are the objectives of U_i^t and X_i^t, respectively. X_{best}^t is the best of X_i^{t+1}, $i = 1,\dots,NP$, which has a minimum $F(X_i^{t+1})$.

2.2 Conjugate direction method

In 1964, Powell [20] proposed a conjugate direction (CD) method where the function does not need to be differentiable, and no derivatives are taken. Various conjugate gradient methods use different techniques for constructing conjugate directions. The so-called zero-order methods work with $\Psi(X)$ only, whereas the first-order methods utilize both $\Psi(X)$ and $\nabla\Psi$. Powell's method is a zero-order method because it requiring the evaluation of $\Psi(X)$ only [10]. If the problem involves n design variables, the basic algorithm is

Algorithm 1: Conjugate direction algorithm.

Step 1 Choose a point x_0 in the design space.
Step 2 Choose the starting vectors $v_i, i = 1,2,\dots,n$.
Step 3 Evaluate $\min \Psi(x)$ along the line through x_{i-1} in the direction of v_i for each element i. Let the minimum point be x_i.
Step 4 Calculate $v_{n+1} = x_0 - x_n$ (this vector is conjugate to v_{n+1} produced in the previous step) Minimize $\Psi(x)$ along the line through x_0 in the direction of v_{n+1}. Let the minimum point be x_{n+1}.
Step 5 Compare $x_{n+1} - x_0$, if $|x_{n+1} - x_0| < \varepsilon$ then exit.
Step 6 Update the position of vectors $v_{i+1} = v_i$ (v_1 is neglected and the other vectors are reused).
Step 7 Repeat Step 3 to Step 6 until termination criteria are met.

3 The proposed algorithm

CD gets a better solution for the optimization problem in (2) than the initial solution [10,16]. When CD has a good initial solution, it gives a good optimal solution and converges fast. It is known that it is hard to select a good initial solution. In general, initial solution is randomly given, so CD may take a long time. Indeed, If the intial solution is not goof, CD may fail to give a good optimization solution. DE is a stochastic algorithm, and is similar to hill climbing algorithm. The initial solutions are more than one solution, so the optimization value of the objective function will decline fast at first when DE is applied to optimization problem. But with the development of the calculation, the mutant individual $V_{i,G}$ is stable at the same solutions X_i, and sometimes it gets trapped into local minima. To overcome this problem, conjugate direction differential evaluation (DE-Powell) is proposed. When DE gets trapped into local minima, by the property of conjugate direction "Powell" method, conjugate direction method is applied on the initial solution X_{best}, then a new solution X_{best}^* is obtained which s better than X_{best} (it helps DE to jump over local minima). Replaced X_{best} by X_{best}^*, DE is applied again and into a new loop, and so on, until termination criterion is reached.

During our experiments with DE-Powell, in some cases that there are no improvement happens after applying the best solution X_{best}^* "local minimum" obtained from Powell method in Section 2.2, because of the trapping local minima for the inability of the way to avoid it. To solve this problem, the parameter F in DE algorithm is changed from 0.6 (the better choice value [15]) to become a random number $F = rand$ at each iteration after applying Powell method. The improved results are shown in Table 1 and represented in Figures 1–2. In Algorithm **??**, the detailed algorithm explains the various steps involved in the DE-Powell algorithm and is written for easier implementation and understanding of the proposed algorithm.

Algorithm 2: Pseudo code for DE-Powell algorithm.

Step 1: Initialise DE-Powell Parameters
$\Psi(X)$: Objective function (2)
n: Number of variables
lb: Lower bound
ub: Upper bound
F: Differential weight = 0.6
CR: Crossover probability = 0.3
NP: Population size = 20
Maxiter: Maximum number of iterations = 8000.
Step 2: Initialise a population P
Initialize all individuals $X_i \in P$, $i = 1,\ldots,NP$ with random positions in the search-space $[lb,ub]$.
Step 3: Improvise DE-Powell
while *(not Maxiter or find the optimal solution)* **do**
  Update differential weight $F = rand$ after applying Powell method, Step 4 in Section 2.2.
  for $i = 1$ *to number of population size NP* **do**
    Randomly select three agents from the population P.
    if $F < 0.95$ **then**
      DE/best/1: $V_i = X_{best} + F(X_{r2} - X_{r3})$
    else
      DE/rand/1: $V_i = X_{r1} + F(X_{r2} - X_{r3})$
    Select a random index $R \in \{1,2,\ldots,n\}$.
    Pick a uniformly distributed number $r \in [0,1]$.
    for *j = 1 to number of decision variables n* **do**
      if $r < CR$ *or* $j = R$ **then**
        set $u_i^j = V_i^j$. Otherwise set $u_i^j = X_i^j$.
    Swap a randomly two variables in $\{1,2,\ldots,n\}$ of current u_i if $rand < 0.5$.
  Evaluate the objective function Ψ for a trial vector $U = \{u_i,\ldots,u_{NP}\}$.
  Replace the individual in the population P with the improved candidate solution if $\Psi(u_i) < \Psi(X_i)$.
  Compute the minimum fitness of $\Psi(X_i)$ as the best solution X_{best}.
  Step 4: Conjugate Direction Method (Powell)
  if $\Psi(X_{best})$ *does not change much over several iterations (generally it is 15 iterations)* **then**
    Set X_{best} as an initial solution for Powell method.
    Apply Powell method in Section 2.2.
    Update the position X_{best} by X^*_{best} which obtained from the Powell if $\Psi(X^*_{best}) < \Psi(X_{best})$.
Solution= X_{best} for $\Psi(X_{best})$.

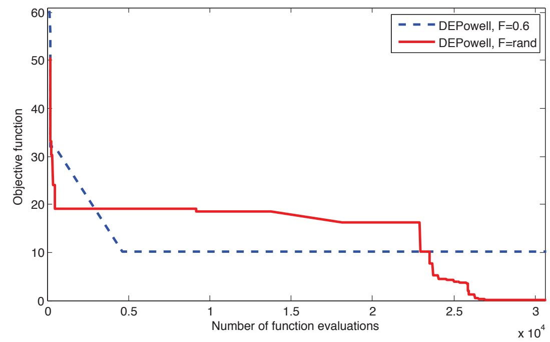

Fig. 1: This is an example for trapped into local minima. DE-Powell ($F = rand$) is able to overcome DE-Powell ($F = 0.6$) and jump over local minima in this test problem Eq. (10), the objective function stays on 10.1616 (local minimum) for many iteration without improving during using DE-Powell ($F = 0.6$), in contrast DE-Powell ($F = rand$) was able to achieve the objective function 9.39406E-06 after 30680 function evaluations.

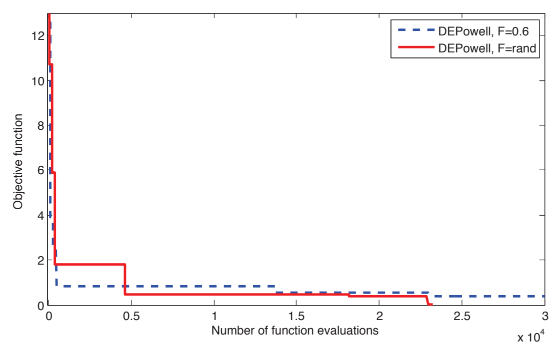

Fig. 2: This is another example for trapped into local minima. DE-Powell ($F = rand$) is able to overcome DE-Powell ($F = 0.6$) and jump over local minima in this test problem Eq. (12), the objective function stays on 0.4004 (local minimum) for many iteration without improving during using DE-Powell ($F = 0.6$), in contrast DE-Powell ($F = rand$) is able to achieve the objective function 7.482E-09 after 23260 function evaluations.

4 Test problems and performance

Our aim is to improve the quality of the solution by combine DE algorithm and Powell method introduced in Section 3. The success rate is compare on well known test functions of each algorithm Powell, DE and DE-Powell and the best results are recorded to show best performance for our algorithm. These functions are listed in the following subsections. The algorithms are run for 50 times and maximum iteration is set to 3000 with population size 20. Successful run criterion is considered when the fitness function tends to 10^{-6}. To evaluate performance of each algorithm, the success rate is defined as the number of solutions which are found to be successful out of the number of all trials (50 for this study).

In order to evaluate performance, a minimum, mean, maximum and stranded division for successful runs are calculated and recorded.

The performance of the algorithms Powell, DE, DE-Powell ($F = 0.6$) and DE-Powell ($F = rand$) are presented in Table 1.

Test 1. The nonlinear system of dimension 10 [16] is defined by

$$\begin{cases} f_1(x) = (3-5x_1)+x_1+1-2x_2, \\ f_i(x) = (3-5x_i)x_i - x_{i-1} - 2x_{i+1},\ \text{i=2,...,9}; \\ f_{10}(x) = (3-5x_{10})x_{10}+1-x_9. \end{cases} \quad (6)$$

where $x \in [-5,5]$. In this test the solution is unknown.

Test 2. Consider the following nonlinear system [13]

$$\begin{cases} x_1+x_2-2x_3 = 0, \\ x_1x_2 = 0, \\ x_1^2+x_2^2 = 2. \end{cases} \quad (7)$$

where $0 \leq x_1,x_2,x_3 \leq 10$. The best result is $x^* = (1,1,1)^T$.

Test 3. Three dimension system of nonlinear equations [13] is given as

$$\begin{cases} 821y-263yz+661 = 0, \\ 613xz-977xy-268 = 0, \\ 977xz+373x-647yz-811 = 0. \end{cases} \quad (8)$$

where $0 \leq x,y,z \leq 10$. The best result is $x^* = (2,3,5)^T$.

Test 4. The nonlinear system of dimension three is given as

$$\begin{cases} 3x^2+2yz-\sqrt{2}x-6 = 0, \\ xz-y-\sqrt{2}+1 = 0, \\ xy-z-\sqrt{2}+1 = 0. \end{cases} \quad (9)$$

where $0 \leq x,y,z \leq 2$. The best result is $x^* = (\sqrt{2},1,1)^T$.

Test 5 Consider the following nonlinear system [13]

$$\begin{cases} -94x^{15}-64+90x^2-38xy^3 = 0, \\ 64-22y^2z^{20}-37xy = 0, \\ -20x-7y+4y^{20}+1+z = 0. \end{cases} \quad (10)$$

where $-5 \leq x,y,z \leq -5$. The best result is unknown.

Test 6. The following nonlinear system is defined by

$$\begin{cases} f_1(x) = x_1^2-x_2+1, \\ f_2(x) = x_1-\cos(0.5\pi x_2). \end{cases} \quad (11)$$

where $x \in [-2,2]$. The best result is $x^* = (-1/\sqrt{2},1.5)^T$, $x^* = (0,1)^T$, $x^* = (-1,2)^T$.

Test 7. Consider the following nonlinear system

$$\begin{cases} (x_1-5x_2)^2+40(\sin^2(10x_3)) = 0, \\ (x_2-2x_3)^2+40(\sin^2(10x_1)) = 0, \\ (3x_1+x_3)^2+40(\sin^2(10x_2)) = 0. \end{cases} \quad (12)$$

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

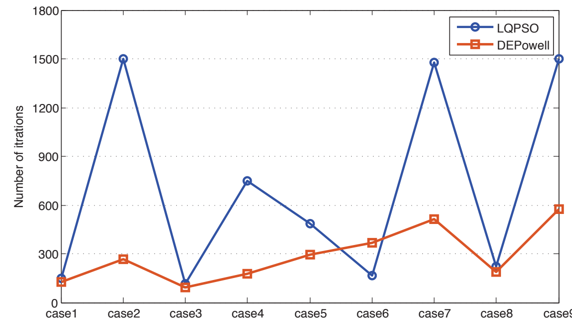

Fig. 3: The convergence to the optimum solution for our algorithm and LQPSO [28] within number iterations

5 Illustrative examples and simulation results

In this section, the performance of DE-Powell is investigated on nine case studies of systems of non linear equations. 100 consecutive algorithm runs are performed for each case and best results are compared with the obtained by Chaotic Quantum Particle Swarm Optimization (LQPSO) [28], Intelligent Tuned Harmony Search algorithm (ITHS) [32], imperialist competitive algorithm (ICA) [1], Quantum behaved Particle Swarm Optimization (QPSO) [25], [26], multiobjective approach for nonlinear systems [4], fuzzy adaptive simulated annealing [18], weighted-Newton method [23], combined method based on Grobner bases [8], and filled function method[31], Gravitational Search algorithm (GRAV) [22]. In order to be consistent and fair in our computations and comparisons, the parameters in these algorithms are adopted the same in this paper. For example, the values of the parameters used in ITHS, harmony memory size = 20 and harmony memory consideration rate = 0.95 [28], are considered in our experiments. The statistical results in terms of minimum value, standard deviation value, mean deviation value and maximum value for each algorithm for each problem are reported in Table 2. Numerical results show the accuracy and robustness of the proposed algorithm. Also, the proposed algorithm is efficient and outperforms other algorithms in most of cases with regard to the minimum objective function values.

In addition, our algorithm converges to the optimum solution within minimum numbers of iterations for all the nine case studies of systems of non linear equations compared to LQPSO which was better than the other algorithms during the experiments as shown in Figure 3. The systems of non linear equations problems are described in the following subsections.

Table 1: Statistical results obtained after 50 runs

	algorithms	min	mean	max	Std	Suc
Test 1	Powell	3553	5422.194	7601	812.7172	31
	DE	-	-	-	-	0
	DE-Powell ($F = 0.6$)	10380	16479.07	24875	3256.237	28
	DE-Powell ($F = rand$)	16046	17934.57	19942	1054.275	**37**
Test 2	Powell	**254**	**793.6**	**1204**	184.4521	**50**
	DE	1060	1828.4	2640	305.2223	**50**
	DE-Powell ($F = 0.6$)	624	1194.5	1613	225.9470	**50**
	DE-Powell ($F = rand$)	571	1011	1431	**178.9323**	**50**
Test 3	Powell	544	1064.43	2701	456.8366	42
	DE	12760	**17122.4**	23620	1994.558	**50**
	DE-Powell ($F = 0.6$)	**1027**	2880.08	16920	2987.294	**50**
	DE-Powell ($F = rand$)	1131	1963.64	**5873**	**962.4363**	**50**
Test 4	Powell	406	772.229	1080	148.7187	48
	DE	3120	5649.6	9320	1328.631	**50**
	DE-Powell ($F = 0.6$)	922	1544.98	5001	567.7985	**50**
	DE-Powell ($F = rand$)	**764**	**1149.68**	**1630**	**196.6570**	**50**
Test 5	Powell	808	3915.583	14345	5096.96	12
	DE	5980	7641.429	11240	1285.347	28
	DE-Powell ($F = 0.6$)	1246	7295.162	22515	6325.193	37
	DE-Powell ($F = rand$)	1344	7158.927	22594	6309.731	**41**
Test 6	Powell	**197**	**288.5**	**448**	**52.87808**	**50**
	DE	420	1846	4040	579.2219	**50**
	DE-Powell ($F = 0.6$)	513	748.16	994	122.1423	**50**
	DE-Powell ($F = rand$)	473	624.28	840	83.53981	**50**
Test 7	Powell	-	-	-	-	0
	DE	1160	1650.714	2100	245.5821	28
	DE-Powell ($F = 0.6$)	1288	4180.022	9804	2203.875	45
	DE-Powell ($F = rand$)	828	4020.128	20585	4763.094	**47**

Table 2: Statistical results obtained after 100 runs

	min	std	mean	max	min	std	mean	max
	DE-Powell				LQPSO			
case1	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
case2	**0**	8.0229347E-15	3.1237356E-15	5.7392517E-14	3.2107537E-08	7.5730680E-02	2.5461613E-02	7.5262288E-01
case3	**0**	**0**	**0**	**0**	**0**	4.2459467E-33	1.4061158E-33	2.4651903E-32
case4	**0**	2.3966456E-13	3.8014094E-14	1.8645581E-12	**0**	2.9964896E-73	2.3914591E-74	3.7785053E-72
case5	5.2146631E-17	5.6952638E-17	7.2556415E-17	4.0913034E-16	**0**	**0**	**0**	**0**
case6	**1.1157657E-16**	**1.2095720E-08**	**2.7101581E-09**	**9.0895608E-08**	6.4752361E-16	5.9929806E-02	2.4977839E-02	0.25783943858
case7	**6.3689982E-08**	**4.0739966E-05**	**4.8095697E-06**	**4.0811123E-04**	6.2083149E-06	2.0622932E-04	6.4420291E-04	1.1372704E-03
case8	5.5648201E-17	2.4796804E-14	4.3438963E-15	2.3681532E-13	**2.6020852E-18**	**3.1857439E-15**	**8.4696820E-16**	**1.9212860E-14**
case9	**0**	**4.7468116E-13**	**2.5396348E-13**	**1.5007093E-12**	1.6298215E-10	102.025240148	168.705483779	551.460632344
	GRAV				ITHS			
case1	6.7103563E-04	5.1351410E-03	6.2169245E-03	1.7282111E-02	1.4157747E-08	1.6631394E-02	1.2369408E-02	1.1531520E-01
case2	2.8569775E-03	8.6720012E-04	4.6682424E-03	6.2627827E-03	8.8894196E-04	6.3627454E-02	8.8435649E-02	3.0142225E-01
case3	1.1483009E-10	7.1629484E-08	6.7357006E-08	3.0755886E-07	**0**	2.1482566E-03	4.6780060E-04	1.8213222E-02
case4	4.1591101E-08	2.9752350E-04	2.5507443E-04	8.3065113E-04	7.3957098E-05	7.2097385E-03	6.8963583E-03	5.4154614E-02
case5	2.3215168E-05	9.2757904E-05	1.6472379E-04	3.8186114E-04	8.6436797E-03	1.2961480E-02	2.8485105E-02	8.8232708E-02
case6	2.0045736E-04	1.4585813E-02	1.3579944E-02	8.5368317E-02	1.0355349E-01	1.8684471E-01	0.39870851563	1.2277557987
case7	5.3454898E-04	1.5998926E-03	2.6490305E-03	5.6091875E-03	3.7720668E-04	2.3264434E-02	2.0104330E-02	0.1620341415
case8	2.8720052E-04	8.8975401E-02	0.13831108452	0.29177988077	2.3452928E-02	7.1914483E-02	0.15654934659	0.4159781524
case9	7.47089935933	17.8057247481	37.1556293303	130.470752028	4.15504414694	338.294665653	312.651184712	5555.0798892

5.1 Case study 1

This example is Brown's almost linear system [3], is given by

$$\begin{cases} 2x_1+x_2+x_3+x_4+x_5=6, \\ x_1+2x_2+x_3+x_4+x_5=6, \\ x_1+x_2+2x_3+x_4+x_5=6, \\ x_1+x_2+x_3+2x_4+x_5=6, \\ x_1x_2x_3x_4x_5=1, \end{cases} \tag{13}$$

where $-2 \leq x_i \leq 2$, $i=1,\ldots,5$. There are two optimal solutions for this system as $x^*=(1,1,1,1,1)^T$ and $x^*=(0.916,0.916,0.916,0.916,1.418)^T$. Table 3 gives the optimum results by DE-Powell and compares the results with other metaheuristic algorithms. DE-Powell algorithms obtains the two optimum solutions #1 and #2, as shown in Table 3 and LQPSO obtained the same optimum solutions [28]. FPWOA and LQPSO get the optimum solution over 100 runs with standard division zero since DE-Powell and LQPSO outperform the other methods.

5.2 Case study 2

This system of dimension 6 is given in [12,16]

$$\begin{cases} x_1+\frac{1}{4}x_2^4x_4x_6+0.75=0, \\ x_2+0.405\exp(1+x_1x_2)-1.405=0, \\ x_3-\frac{1}{2}x_4x_6+1.5=0, \\ x_4-0.605\exp(1-x_3^2)-0.395=0, \\ x_5-\frac{1}{2}x_2x_6+1.5=0, \\ x_6-x_1x_5=0. \end{cases} \tag{14}$$

where $-2 \leq x_i \leq 2$, $i=1,\ldots,6$. The optimal solution of this system is $x^*=(-1,1,-1,1,-1,1)^T$. Table 4 compares the optimum results that obtained by DE-Powell and others concerned methods. As shown in Table 4, DE-Powell surpass other algorithms, namely, LQPSO, QPSO, GRAV and ITHS. DE-Powell gets the optimum solution within 100 runs on all aspects of performance, while the other methods are not able to get the optimal solution over all 100 runs.

5.3 Case study 3

The system of four nonlinear equations is given in [7,23].

$$x_i - cos\left(2x_i - \sum_{j=1}^{4} x_j\right) = 0, \quad 1 \leq i \leq 4. \tag{15}$$

This system has been solved by our proposed algorithm DE-Powell and other algorithms, the best known solution of this case study is shown in Table 5. Table 5 compares the best results recorded in [23,28], all algorithms except GRAV obtained the optimal solution for this case.

5.4 Case study 4

Neurophysiology application example is proposed in [1, 4,?,30] and utilized to test the effectiveness and robustness of our algorithm. Problem consists of six nonlinear equations described as

$$\begin{cases} x_1^2+x_3^2=1, \\ x_2^2+x_4^2=1, \\ x_5x_3^3+x_6x_4^3=c_1, \\ x_5x_1^3+x_6x_2^3=c_2, \\ x_5x_1x_3^2+x_6x_2x_4^2=c_3, \\ x_5x_3x_1^2+x_6x_4x_2^2=c_4, \end{cases} \tag{16}$$

where $-10 \leq x_i \leq 10$, $i=1,\ldots,6$. The constants c_i is randomly chosen. In our experiments, $c_i=0, i=1,\ldots,4$ are considered. The best known solution with different solutions has been shown in Table 6. Table 6 compares the best known solution with different solutions in [1,28] and other algorithms with our proposed algorithm. DE-Powell, LQPSO, QPSO and imperialist competitive algorithm (ICA) [1] algorithms obtained four different optimum solutions.

5.5 Case study 5

10-dimensional interval arithmetic benchmark [4,5,18, 28], The problem is described as

$$\begin{cases} x_1-0.25428722-0.18324757x_4x_3x_9=0, \\ x_2-0.37842197-0.16275449x_1x_{10}x_6=0, \\ x_3-0.27162577-0.16955071x_1x_2x_{10}=0, \\ x_4-0.19807914-0.15585316x_7x_1x_6=0, \\ x_5-0.44166728-0.19950920x_7x_6x_3=0, \\ x_6-0.14654113-0.18922793x_8x_5x_{10}=0, \\ x_7-0.42937161-0.21180486x_2x_5x_8=0, \\ x_8-0.07056438-0.17081208x_1x_7x_6=0, \\ x_9-0.34504906-0.19612740x_{10}x_6x_8=0, \\ x_{10}-0.42651102-0.21466544x_4x_8x_1=0. \end{cases} \tag{17}$$

where $-2 \leq x_i \leq 2$. Table 7 gives the optimum results found in [4,28,18]. DE-Powell and Oliveira [18] algorithms get the same global best solution whereas LQPSO and QPSO has served optimum solution as mentioned in [28], which is the same solution from our proposed algorithm and Oliveira and Petraglia [18].

5.6 Case study 6

The inverse position problem for a six-revolute joint problem application [4,28] is taken as the benchmark problem. The problem can be described as

$$\begin{cases} x_i^2+x_{i+1}^2-1=0, \\ a_{1i}x_1x_3+a_{2i}x_1x_4+a_{3i}x_2x_3+a_{4i}x_2x_4+a_{5i}x_2x_7+ \\ a_{6i}x_5x_8+a_{7i}x_6x_7+a_{8i}x_6x_8+a_{9i}x_1+a_{10i}x_2+a_{11i}x_3+ \\ a_{12i}x_4+a_{13i}x_5+a_{14i}x_6+a_{15i}x_7+a_{16i}x_8+a_{17i}=0. \end{cases} \tag{18}$$

Table 3: Optimum results for case study 1

	DE-Powell		LQPSO		GRAV	ITHS	Jäger and Ratz [8]
	Sol #1	Sol #2	Sol #1	Sol #2			
x_1	1	0.91635458253	1	0.91635458253	0.91656602848	0.99999998393	0.91635458253
x_2	1	0.91635458253	1	0.91635458253	0.91626240767	0.99999998426	0.91635458253
x_3	1	0.91635458253	1	0.91635458253	0.91698124986	0.99999998465	0.91635458253
x_4	1	0.91635458253	1	0.91635458253	0.91614164659	0.99999998406	0.91635458253
x_5	1	1.41822708733	1	1.41822708733	1.41747835261	1.00000007618	1.41822708733
f_1	0	0	0	0	-4.2863100E-06	-2.9903315E-09	3.3750780E-14
f_2	0	0	0	0	-3.0790712E-04	-2.6576110E-09	2.3980817E-14
f_3	0	0	0	0	4.1093507E-04	-2.2731319E-09	2.3980817E-14
f_4	0	0	0	0	-4.2866821E-04	-2.8568605E-09	2.3980817E-14
f_5	0	0	0	0	5.3287740E-05	1.3080827E-08	2.1982416E-14
Ψ_{best}	**0**	**0**	**0**	**0**	6.7103563E-04	1.4157747E-08	5.7858280E-14

Table 4: Optimum results for case study 2

	DE-Powell	LQPSO	QPSO	GRAV	ITHS
x_1	-1	-1.00000071857	-1.06006024401	-1.00000071857	-1.00469360194
x_2	1	1.00000045092	1.03789221093	1.00000045092	1.00269955201
x_3	-1	-0.99999920901	-0.96490791150	-0.99999920901	-0.99763045118
x_4	1	1.00000957563	1.04304617227	1.00000957563	1.00270464525
x_5	-1	-0.99999966616	-0.96783925322	-0.99999966616	-0.99763694114
x_6	1	1.00000047454	1.02588330513	1.00000047454	1.00262777389
f_1	0	9.0373000E-08	3.6034276E-04	1.4948000E-03	-6.3374000E-04
f_2	0	-2.2725000E-08	-7.3227699E-04	-1.8160000E-03	-2.8872000E-04
f_3	0	7.4938000E-08	7.0261200E-05	-1.0845000E-03	-3.0021000E-04
f_4	0	4.6304000E-10	-1.4208871E-04	-8.9244000E-04	-1.6590000E-04
f_5	0	-1.2889000E-07	-2.1739908E-04	5.3726000E-04	-3.0415000E-04
f_6	0	8.9808000E-08	-8.4609808E-05	-6.0686000E-04	3.0832000E-04
Ψ_{best}	**0**	3.2107000E-08	8.6349494E-04	2.8569000E-03	8.8894000E-04

Table 5: Optimum results for case study 3

	DE-Powell	LQPSO	QPSO	GRAV	ITHS	Sharma and Arora [23]
x_1	0.5149332646611	0.5149332646611	0.5149332646611	0.5149332646611	0.5149332646611	0.5149332646611
x_2	0.5149332646611	0.5149332646611	0.5149332646611	0.5149346009788	0.5149332646611	0.5149332646611
x_3	0.5149332646611	0.5149332646611	0.5149332646611	0.5149237165235	0.5149332646611	0.5149332646611
x_4	0.5149332646611	0.5149332646611	0.5149332646611	0.5149444327230	0.5149332646611	0.5149332646611
f_1	0	0	0	2.259407661E-06	0	0
f_2	0	0	0	-7.041394250E-08	0	0
f_3	0	0	0	7.706203400E-06	0	0
f_4	0	0	0	-7.094691013E-07	0	0
Ψ_{best}	**0**	**0**	**0**	1.148300900E-10	**0**	**0**

Table 6: Optimum results for case study 4

	DE-Powell	LQPSO	QPSO	GRAV	ITHS	ICA
x_1	-0.8419172629	0.4462091846	-0.7966166843	0.8353268473	0.7579922172	-0.0410960509
x_2	0.8419172629	-0.4462091846	0.7966166843	0.7828606939	0.7579956367	0.0410960509
x_3	-0.5396066367	0.8949286919	-0.6044847875	0.5497536704	0.6522901471	0.9991552005
x_4	0.5396066367	-0.8949286919	0.6044847875	-0.6221970203	0.6523056989	-0.9991552005
x_5	9.275556E-02	0.3667790583	-0.3435296497	-1.258964E-08	2.604670E-02	9.873355E-02
x_6	9.275556E-02	0.3667790583	-0.3435296497	1.436173E-08	-2.600994E-02	9.873355E-02
f_1	0	0	0	3.985034E-08	3.463733E-05	0
f_2	0	0	0	-1.780246E-09	6.011012E-05	0
f_3	0	0	0	-5.551106E-09	9.686086E-06	0
f_4	0	0	0	-4.474299E-10	1.585609E-05	0
f_5	0	0	0	1.174203E-09	1.141786E-05	0
f_6	0	0	0	-1.030592E-08	1.345653E-05	0
Ψ_{best}	**0**	**0**	**0**	4.159110E-08	7.395710E-05	**0**

Table 7: Optimum results for case study 5

	DE-Powell	LQPSO	QPSO	GRAV	ITHS	Grosan and Abraham [4]	Oliveira and Petraglia [18]
x_1	0.257833393701	0.257833393701	0.257833393701	0.257839946927	0.254686410313	0.046490512	0.257833393701
x_2	0.381097154603	0.381097154603	0.381097154603	0.381079261668	0.378523004753	0.101356836	0.381097154603
x_3	0.278745017346	0.278745017346	0.278745017346	0.278737809173	0.276525468374	0.084057782	0.278745017346
x_4	0.200668964225	0.200668964225	0.200668964225	0.200676775829	0.201804033261	-0.138846031	0.200668964225
x_5	0.445251424841	0.445251424841	0.445251424841	0.445251560410	0.443869219215	0.494390574	0.445251424841
x_6	0.149183919969	0.149183919969	0.149183919969	0.149185582343	0.147985685016	-0.076068516	0.149183919969
x_7	0.432009698984	0.432009698984	0.432009698984	0.432006811493	0.432376554489	0.247581911	0.432009698984
x_8	0.073402777776	0.073402777776	0.073402777776	0.073403712785	0.069871690819	-0.017074816	0.073402777776
x_9	0.345966826876	0.345966826876	0.345966826876	0.345965056291	0.349297348759	0.000366754	0.345966826876
$x_1$0	0.427326275993	0.427326275993	0.427326275993	0.427333090362	0.432318039408	0.148111931	0.427326275993
f_1	-2.12503626E-17	0	0	6.52503549E-06	-3.17270311E-03	0.207795924	-2.12503626E-17
f_2	-1.64798730E-17	0	0	-1.80334006E-05	-2.55089378E-03	0.276979885	-1.64798730E-17
f_3	-2.42861287E-17	0	0	-7.16837993E-06	-2.16674725E-03	0.187686321	-2.42861287E-17
f_4	9.97465999E-18	0	0	7.73423073E-06	1.18507157E-03	0.336788711	9.97465999E-18
f_5	2.12503626E-17	0	0	2.12270381E-07	-1.32810307E-03	5.30391E-02	2.12503626E-17
f_6	1.30104261E-18	0	0	1.58576125E-06	-1.09258759E-03	0.222373054	1.30104261E-18
f_7	5.20417043E-18	0	0	-2.79803508E-06	5.18467284E-04	0.181608475	5.20417043E-18
f_8	-5.20417043E-18	0	0	8.50208862E-07	-3.47628514E-03	8.74896E-02	-5.20417043E-18
f_9	1.31188463E-17	0	0	-1.80713729E-06	3.37156538E-03	0.344720037	1.31188463E-17
f_{10}	-2.50450702E-17	0	0	6.75152562E-06	5.03611772E-03	0.278422749	-2.50450702E-17
Ψ_{best}	5.21466306E-17	**0**	**0**	2.32151680E-05	8.64367970E-03	0.746860040	5.21466306E-17

where $-10 \leq x_i \leq 10$, $i = 1,\ldots,4$, and the coefficients a_{ji} where $1 \leq i \leq 4$ and $1 \leq j \leq 17$ are given in Table 8. Table 9 shows the best of the different solutions obtained by our algorithm and other algorithms [28]. Moreover some of the non dominated solutions of Grosan and Abraham [4]. DE-Powell gets the best result for this case and outperforms other algorithms.

5.7 Case study 7

The combustion problem [4,18,28,29] occurred at a temperature of 3000 °C. It is described by system consisting of ten of nonlinear equations as follows.

$$\begin{cases} x_2 + 2x_6 + x_9 + 2x_{10} - 10^{-5} = 0, \\ x_3 + x_8 - 3\times 10^{-5} = 0, \\ x_1 + x_3 + 2x_5 + 2x_8 + x_9 + x_{10} - 5\times 10^{-5} = 0, \\ x_4 + 2x_7 - 10^{-5} = 0, \\ 0.5140437\times 10^{-7} x_5 - x_1^2 = 0, \\ 0.1006932\times 10^{-6} x_6 - 2x_2^2 = 0, \\ 0.7816278\times 10^{-15} x_7 - x_4^2 = 0, \\ 0.1496236\times 10^{-6} x_8 - x_1 x_3 = 0, \\ 0.6194411\times 10^{-7} x_9 - x_1 x_2 = 0, \\ 0.2089296\times 10^{-14} x_{10} - x_1 x_2^2 = 0. \end{cases} \quad (19)$$

Table 8: Parameters for case study 6

a_{ij}	1	2	3	4
1	-0.249150680	0.125016350	-0.635550070	1.48947730
2	1.609135400	-0.686607360	-0.115719920	0.23062341
3	0.279423430	-0.119228120	-0.666404480	1.32810730
4	1.434480160	-0.719940470	0.110362110	-0.25864503
5	0.000000000	-0.432419270	0.290702030	1.16517200
6	0.400263840	0.000000000	1.258776700	-0.26908494
7	-0.800527680	0.000000000	-0.629388360	0.53816987
8	0.000000000	-0.864838550	0.581404060	0.58258598
9	0.074052388	-0.037157270	0.195946620	-0.20816985
10	-0.083050031	0.035436896	-1.228034200	2.68683200
11	-0.386159610	0.085383482	0.000000000	-0.69910317
12	-0.755266030	0.000000000	-0.079034221	0.35744413
13	0.504201680	-0.039251967	0.026387877	1.24991170
14	-1.091628700	0.000000000	-0.057131430	1.46773600
15	0.000000000	-0.432419270	-1.162808100	1.16517200
16	0.049207290	0.000000000	1.258776700	1.07633970
17	0.049220729	0.013873010	2.162575000	-0.69686809

Table 9: Optimum results for case study 6

	DE-Powell	LQPSO	QPSO	GRAV	ITHS	Grosan and Abraham [4]
x_1	0.997945582200	0.590050131899	0.772231269943	-0.418714637585	-0.481187883256	-0.0625820337
x_2	-0.064067269078	0.807366609320	0.635092244415	0.908150863302	0.855827688882	0.7777446281
x_3	-0.997945582200	-0.590050131890	0.772345476746	-0.418496945674	0.489354396993	-0.0503725828
x_4	-0.064067269078	0.807366609320	0.635464640411	0.908273850895	0.856058114077	0.3805368959
x_5	0.997945582200	-0.590050131890	-0.771984058288	0.418286659161	0.529021825853	-0.5592587603
x_6	0.598972482326	0.942931949016	0.501630602443	-0.267011571006	-0.413193459556	-0.6988338865
x_7	0.447874788025	0.266895066252	0.016544744589	-0.589809505947	-0.583759143370	0.3963927675
x_8	-0.815231997842	-1.224247772170	-1.199869634114	-0.998579648273	-0.817013237701	0.0861763643
f_1	0	2.22044604E-16	-3.16706807E-04	5.99382446E-05	-3.60171880E-02	0.3911967824
f_2	0	0	-1.40305634E-04	-1.22315945E-04	-2.80912411E-02	0.3925758963
f_3	0	0	3.32844662E-04	1.01081758E-04	-2.76967795E-02	0.8526542737
f_4	0	0	-2.25304537E-04	-7.48825476E-05	1.26995869E-02	0.5424213097
f_5	6.93889390E-18	-3.40005800E-16	8.67394191E-04	6.52303303E-05	-1.74787136E-02	0.7742116224
f_6	8.67361738E-18	-6.07153210E-17	1.36536886E-03	3.21280003E-05	-6.43899846E-02	0.1537105718
f_7	0	0	7.99271797E-04	-1.70272692E-06	-8.18048271E-03	0.9116019977
f_8	1.11022302E-16	1.11022302E-16	5.32099950E-04	2.26792051E-05	-5.64808155E-02	0.1519175234
Ψ_{best}	**1.11576573E-16**	4.25346978E-16	1.95450943E-03	2.00457360E-04	0.103593498941	1.6749682555

where $-20 \leq x_i \leq 20$. Table 10 presents the best result with different solutions obtained by DE-Powell and other algorithms. Our algorithm finds a lot of non dominated solutions for this problem, and obtains the best solution comparing with the other algorithms. Also, DE-Powell obtains the minimum worst solution and minimum standard division among the other algorithms. Oliveira and Petraglia [18] reported six different solutions using the fuzzy adaptive simulated annealing algorithm, which are the best results of this study and added into Table 10 for the sake of comparison.

5.8 Case study 8

Consider one of benchmark problems [1,31] that consists of eight non linear equations and is defined by

$$\begin{cases} 4.731 \times 10^{-3} x_1 x_3 - 0.3578 x_2 x_3 - 0.1238 x_1 + x_7 - 1.637 \times 10^{-3} x_2 - 0.9338 x_4 - 0.3571 = 0, \\ 0.2238 x_1 x_3 + 0.7623 x_2 x_3 + 0.2638 x_1 - x_7 - 0.07745 x_2 - 0.6734 x_4 - 0.6022 = 0, \\ x_6 x_8 + 0.3578 x_1 + 4.731 \times 10^{-3} x_2 = 0, \\ -0.7623 x_1 + 0.2238 x_2 + 0.3461 = 0, \\ x_1^2 + x_2^2 - 1 = 0, \\ x_3^2 + x_4^2 - 1 = 0, \\ x_5^2 + x_6^2 - 1 = 0, \\ x_7^2 + x_8^2 - 1 = 0. \end{cases} \quad (20)$$

where $-1 \leq x_i \leq 1$ and $i = 1, \ldots, 8$. Table 11 presents the known solutions of case study 8 [28] that solved by

Table 10: Optimum results for case study 7

	DE-Powell	LQPSO	QPSO	GRAV	ITHS	Oliveira and Petraglia [18]
x_1	-1.50083955E-07	-5.92864500E-08	-4.88278463E-07	-1.29788338E-03	7.24587841E-03	6.05151660E-07
x_2	8.95047570E-04	-6.94279000E-05	6.47373030E-03	9.18114481E-03	1.01801769E-02	-4.11586963E-03
x_3	-14.68357214583	-0.298022727000	0.988680886100	0.351976543448	-2.90517380E-03	-0.565259901992
x_4	-1.06127505E-04	-8.85260400E-05	6.88493552E-03	1.34497861E-02	-2.32291582E-03	5.84693914E-03
x_5	0.813792089857	-0.412726852200	0.249330591600	0.381060037378	-2.57099289E-03	-0.130586435447
x_6	15.93998306073	-5.47120683E-02	-4.75443378E-03	0.328340515467	-1.28513542E-04	3.61353748E-02
x_7	5.80636695E-05	4.92534440E-05	-3.43749623E-03	-6.72201015E-03	1.07527110E-03	-2.91846910E-03
x_8	14.68360214891	0.298052730400	-0.988651263890	-0.35194359193	2.82171705E-03	0.565289928388
x_9	-0.741481204001	0.945338532100	0.976976825720	-0.15159591940	1.70605627E-04	-0.540050513545
x_10	-15.56968498160	-0.417917503000	-0.486965844210	-0.25714404312	-4.93715825E-03	0.235952824633
f_1	1.83623390E-09	-3.71842543E-08	7.57933610E-12	-3.18298966E-05	2.09438980E-04	1.56804700E-08
f_2	3.08443468E-09	3.40314122E-09	-3.77787894E-07	2.95151945E-06	-1.13456744E-04	2.63960000E-08
f_3	-3.97291811E-09	-1.52430012E-08	3.47636752E-08	1.21588444E-04	2.56003193E-05	1.29259847E-10
f_4	-1.66204936E-10	-1.91525933E-08	-5.69542591E-08	-4.23416696E-07	-1.82373624E-04	9.45794000E-10
f_5	4.18324472E-08	-2.12159327E-08	1.28164436E-08	-1.66491312E-06	5.25028861E-05	-6.71307965E-09
f_6	2.82759772E-09	-1.51495430E-08	-8.38188468E-05	-1.68553778E-04	2.07272018E-04	-3.38771270E-05
f_7	-1.12630473E-08	-7.83685922E-09	-4.74023371E-05	-1.80896747E-04	5.39593790E-06	-3.41866973E-05
f_8	-6.75516492E-09	2.19672601E-08	3.34826022E-07	4.56771848E-04	2.10509583E-05	4.26648682E-07
f_9	-4.57960610E-08	5.85528822E-08	6.36789430E-08	1.19066648E-05	-7.37643137E-05	3.09622231E-08
f_{10}	8.77040994E-14	-5.07159522E-16	2.04623346E-11	1.09403029E-07	7.50933872E-08	-1.02510078E-11
Ψ_{best}	**6.36899822E-08**	6.20831492E-06	9.62956081E-05	5.34548982E-04	3.77206671E-04	6.85713100E-05

imperialist competitive algorithm (ICA) [1] and filled function method for solving a nonlinear system [31]. LQPSO gives the best solution for this case comparing with other algorithms and gets the minimum worst solution and minimum standard division among those algorithms.

5.9 Case study 9

This case study is considered as a benchmark problem [1, 6, 16, 28]. The problem is defined by

$$\begin{cases} A = bh - (b-2t)(h-2t), \\ I_y = \frac{bh^3}{12} - \frac{(b-2t)(h-2t)^3}{12}, \\ I_n = \frac{2t(h-t)^2(b-t)^2}{h+b-2t}. \end{cases} \quad (21)$$

where b is the width of the section, h is the height of the section and t is the thickness of the section. $A = 165, I_y = 9369$ and $I_n = 6835$, $0 \le x_i \le 25$. There are multiple solutions for this nonlinear system. ICA [1] and particle swarm optimization algorithm (PPSO) [6] attain the optimum solutions of this problem utilizing Imperialist Competitive Algorithm and Particle Swarm Optimization method, respectively. Table 12 shows the optimum results obtained from these studies, DE-Powell and other algorithms. DE-Powell outperforms the other algorithms by obtaining the optimal solution for this case. Moreover DE-Powell has the minimum standard division and worst solution. The proposed method obtains the optimal solution for this system as follows: x^*={12.25651959934869594803785730619, 22.894938623626284623924220795743, 2.7898179195381542783138684171718}.

6 Conclusions and future work

In this paper, a novel hybrid algorithm, based on DE and Powell conjugate direction method in order to solve systems of nonlinear equations. The system of nonlinear equations is transformed into an optimization problem. Our proposed algorithm, DE-Powell, has superiority over Powell Conjugate Direction (CD) and Differential Evolution (DE), separately, it overcomes the inaccuracy of Powell conjugate direction method and DE for solving systems of nonlinear equations. In order to check the effectiveness of the proposed algorithm, nine systems of nonlinear equations with different dimensions are employed. Based on the presented results, it is clear that the DE-Powell produced better solutions and outperforms than other algorithms in the literature such as Chaotic Quantum Particle Swarm Optimization (LQPSO) [28], Intelligent Tuned Harmony Search algorithm (ITHS) [32], imperialist competitive algorithm (ICA) [1], Quantum behaved Particle Swarm Optimization (QPSO) [25], [26], multiobjective approach for nonlinear systems [4], fuzzy adaptive simulated annealing [18], weighted-Newton method [23], combined method based

Table 11: Optimum results for case study 8

	DE-Powell	LQPSO	QPSO	GRAV	ITHS	ICA	Wang et al. [31]
x_1	0.67155426182	0.16443166585	0.16443166585	0.67170801324	0.16364139102	0.16443166585	0.67154465
x_2	0.74095537884	-0.98638847685	-0.98638847685	0.74077329294	-0.98627002981	-0.98638847685	0.74097111
x_3	-0.65159061100	0.94762379641	0.94762379641	-0.25300250019	0.95152343463	0.71845260103	0.95189459
x_4	-0.75857081124	-0.31938869811	-0.31938869811	-0.96749642402	-0.30706727299	-0.69557591971	-0.30643725
x_5	0.96254501886	-0.99842747394	-0.99842747394	0.95789826891	-0.99686071792	0.99796438397	0.96381470
x_6	0.27112190370	-0.05605871286	-0.05605871286	0.28712600808	0.03838334620	0.06377372756	-0.26657405
x_7	-0.43757756375	-0.25758509950	-0.25758509950	-0.52841723679	-0.25765415744	-0.52780910528	0.40463693
x_8	-0.89918066911	0.96625561655	0.96625561655	-0.84896958415	0.96464846448	-0.84936302508	0.91447470
f_1	0	0	0	-1.8519834E-04	-1.0141714E-02	2.7755576E-16	-3.750E-06
f_2	0	0	0	-9.1791862E-05	-0.69875127031	-1.1102230E-16	1.537E-05
f_3	-3.9031278E-18	2.6020852E-18	1.7347235E-18	8.0477909E-05	9.0911282E-02	1.7347235E-18	8.990E-06
f_4	-5.5511151E-17	0	1.6653345E-16	-1.5795553E-04	6.2893495E-04	1.6653345E-16	1.084E-05
f_5	0	0	0	-6.3273422E-05	-4.9292345E-04	0	1.039E-05
f_6	0	0	0	5.9595589E-05	-3.1284320E-04	0	7.090E-06
f_7	0	0	0	1.0438088E-05	-4.7954278E-03	0	4.900E-07
f_8	0	0	0	-2.5586905E-05	-3.0676751E-03	0	-4.980E-06
Ψ_{best}	5.5648201E-17	**2.6020852E-18**	1.6654248E-16	2.8720052E-04	0.70473696973	3.4219811E-16	6.318E-10

Table 12: Optimum results for case study 9

	DE-Powell	LQPSO	QPSO	GRAV	ITHS	ICA	PPSO [6]
b	12.2565195993	12.2565196104	12.2566746085	12.2602439079	8.91579028165	8.94308877875	43.155566055
h	22.8949386236	22.8949389227	22.9030352655	22.7756346836	23.2914526043	23.2714818792	10.128950202
t	2.78981791954	2.78981773672	2.78498579560	2.85755430840	12.8853532234	12.9127742914	12.944048458
f_1	165	164.999	165.1859	167.57130	165.874091108	165	709.2412
f_2	9369	9368.99	9369.01	9362.2016	9366.49248249	9369	9369
f_3	6835	6835.00	6834.9887	6836.7276	6831.80426651	6835	6835
Ψ_{best}	**0**	1.6298215E-10	1.8607675E-01	7.47089935933	4.15504414695	N/A	N/A

on Grobner bases [8], and filled function method[31], Gravitational Search algorithm (GRAV) [22].

This work motivates us to various projects as future works, for example, large scale nonlinear systems are under investigation by adjusting our proposed algorithm and can we apply our proposed algorithm on nonlinear systems that arise from the optimality conditions of various optimization problems?

Acknowledgments

The research of the 2nd author is supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

[1] Abdollahi, M., Isazadeh, A., Abdollahi, D., 2013. Imperialist competitive algorithm for solving systems of nonlinear equations. Comput. Math. Appl. 65 (12), 1894–1908.

[2] Differential evolution codes. Www.icsi.berkeley.edu/ storn/code.html.

[3] Floudas,C.A. and Pardalos, P.M., Adjiman, C.S., Esposito, W. R., Gumus, Z. H., Harding, S. T. , Klepeis, J. L., Meyer, C.A., & Schweiger, C., 2013. Handbook of test problems in local and global optimization, Vol. 33. Springer Science & Business Media.

[4] Grosan, C., & Abraham, A., 2008. A new approach for solving nonlinear equations systems. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans 38 (3), 698–714.

[5] Hong, H., & Stahl, V., 1994. Safe starting regions by fixed points and tightening. Computing 53(3-4), 323–335.

[6] Jaberipour, M., Khorram, E., Karimi, B., 2011. Particle swarm algorithm for solving systems of nonlinear equations. Comput. Math. Appl. 62 (2), 566–576.

[7] Jäger, C. and Ratz, D., 1995. A combined method for enclosing all solutions of nonlinear systems of polynomial equations. Reliable Computing 1(1):41–64.

[8] Jäger, C., & Ratz, D., 1995. A combined method for enclosing all solutions of nonlinear systems of polynomial equations. Reliable Computing 1 (1) 41–64.

[9] Kelley, C. T. 2003. Solving Nonlinear Equations with Newton Method, SIAM, Philadelphia.

[10] Kiusalaas, J., 2010. Numerical methods in engineering with MATLAB. Cambridge University Press, 2010.

[11] Koupaei, J. A., and Hosseini, S. M. M., 2015. A new hybrid algorithm based on chaotic maps for solving systems of

nonlinear equations. Chaos, Solitons & Fractals 81, 233–245.
[12] Krzyworzcka, S., 1996. Extension of the Lanczos and CGS methods to systems of nonlinear equations. Journal of computational and applied mathematics 69 (1), 181–190.
[13] Liu, H., Zhou, Y., & Li, Y. (2011) A Quasi-Newton Population Migration Algorithm for Solving Systems of Nonlinear Equations. Journal of Computers 6, 36–42. http://dx.doi.org/10.4304/jcp.6.1.36-42.
[14] Luo, Y.-Z., Tang, G.-J., &Zhou, L.-N., 2008. Hybrid approach for solving systems of nonlinear equations using chaos optimization and quasi-Newton method. Applied Soft Computing 8 (2), 1068–1073.
[15] Mallipeddi, R., Suganthan, P. N., Pan, Q. K., Tasgetiren, M. F., 2011. Differential evolution algorithm with ensemble of parameters and mutation strategies. Applied Soft Computing 11 (2), 1679–1696.
[16] Mo, Y., Liu, H., & Wang, Q., 2009. Conjugate direction particle swarm optimization solving systems of nonlinear equations. Computers & Mathematics with Applications 57 (11), 1877–1882.
[17] Ouyang, A., Zhou, Y., & Luo, Q., 2009. Hybrid particle swarm optimization algorithm for solving systems of nonlinear equations. Granular Computing, 2009, GRC'09. IEEE International Conference on. IEEE.
[18] Oliveira, H. A., & Petraglia, A., 2013. Solving nonlinear systems of functional equations with fuzzy adaptive simulated annealing. Applied Soft Computing 13 (11), 4349–4357.
[19] Pana, Q.-K., Suganthanb, P. N., Wang, L., Gao, L., & Mallipeddib, R., 2011. A differential evolution algorithm with self-adapting strategy and control parameters. Computers & Operations Research 38 (1), 394–408.
[20] Powell M.J. D. 1964, An efficient method for finding the minimum of a function of several variables without calculating derivatives, Computer J. 7 (2), 155–162.
[21] Price, K., Storn, R. M., & Lampinen, J. A., 2006. Differential evolution: a practical approach to global optimization. Springer Science & Business Media.
[22] Rashedi, E., Nezamabadi-pour, H., & Saryazdi, S., 2009. GSA: a gravitational search algorithm, Inform. Science, 179, 2232–2248.
[23] Sharma, J. R., & Arora, H., 2013. On efficient weighted-Newton methods for solving systems of nonlinear equations. Appl. Math. Comput. 222 , 497–506.
[24] Storn, R., & Price, K., 1997. Differential evolution a simple and efficient heuristic for global optimization over continuous spaces. Journal of global optimization 11 (4), 341–359.
[25] Sun, J., Feng, B., & Xu, W. B., 2004. Particle swarm optimization with particles having quantum behavior. IEEE Proceedings of Congress. Evolutionary Computation, pp. 325-331.
[26] Sun, J., Feng, B., & Xu, W. B., 2005. Adaptive parameter control for quantum behaved particle swarm optimization on individual level, in: Proceedings of the 2005 IEEE International Conference on Systems, Man and Cybernetics, Piscataway, NJ, pp. 3049-3054.
[27] Talbi, E., 2009. Metaheuristics: from design to implementation, Vol. 74. John Wiley & Sons.
[28] Turgut, O. E., Turgut, M. S., & Coban, M. T., 2014. Chaotic quantum behaved particle swarm optimization algorithm for solving nonlinear system of equations. Computers & Mathematics with Applications 68 (4), 508–530.
[29] Van Hentenryck, P., McAllester, D., & Kapur D., 1997. Solving polynomial systems using a branch and prune approach. SIAM Journal on Numerical Analysis 34 (2), 797–827.
[30] Verscheide, J., Verlinden, P., & R. Cools, R., 1994. Homotopies exploiting Newton polytopes for solving sparse polynomial systems. SIAM J. Numer. Anal. 31, 915–930.
[31] Wang, C., Luo, R., Wu, K., & Han, B. (2011). A new filled function method for an unconstrained nonlinear equation. Journal of Computational and Applied Mathematics 235(6), 1689–1699.
[32] Yadav P., Kumar, R., Panda, S. K., & Chang, C. S. 2012. An intelligent tuned harmony search algorithm for optimisation. Information Sciences 196, 47–72.
[33] Yang, X.-S., 2010. Nature-inspired metaheuristic algorithms. Luniver press.
[34] Yan, Y., Zhou, Y. and Gong, Q., 2010. Hybrid artificial glowworm swarm optimization algorithm for solving system of nonlinear equations. Journal of Computational Information Systems 6 (10), 3431–3438.

Abdelmonem M. Ibrahim is an assistant professor of computer science at Mathematical department, Faculty of Science, Al-Azhar University, Assiut Branch, Assiut, Egypt. Currently, he is a visiting assistant professor at Electrical and Computer Engineering, The University of British Columbia (Canada). Received the PhD degree in ?Computer Science? at Al-Azhar University, Assiut Branch, Assiut, Egypt. His main research interests are: Data mining, Artificial intelligence, Metaheuristics, signal processing and applications, optimization theory."

Mohamed A. Tawhid got his PhD in Applied Mathematics from the University of Maryland Baltimore County, Maryland, USA. From 2000 to 2002, he was a Postdoctoral Fellow at the Faculty of Management, McGill University, Montreal, Quebec, Canada. Currently, he is a full professor at Thompson Rivers University. His research interests include nonlinear/stochastic/heuristic optimization, operations research, modelling and simulation, data analysis, and wireless sensor network. He has published in journals such as Computational Optimization and Applications, J. Optimization and Engineering, Journal of Optimization Theory and

Applications, European Journal of Operational Research, Journal of Industrial and Management Optimization, Journal Applied Mathematics and Computation, etc. Mohamed Tawhid published more than 50 referred papers and edited 5 special issues in J. Optimization and Engineering (Springer), J. Abstract and Applied Analysis, J. Advanced Modeling and Optimization, and International Journal of Distributed Sensor Networks. Also, he has served on editorial board several journals. Also, he has worked on several industrial projects in BC, Canada.