ARTICLE IN PRESS
Computers and Mathematics with Applications ▮ (▮▮▮▮) ▮▮▮–▮▮▮
Contents lists available at ScienceDirect
Computers and Mathematics with Applications
journal homepage: www.elsevier.com/locate/camwa

Chaotic quantum behaved particle swarm optimization algorithm for solving nonlinear system of equations

Oguz Emrah Turgut [a,*], Mert Sinan Turgut [b], Mustafa Turhan Coban [a]

[a] *Department of Mechanical Engineering, Faculty of Engineering, Ege University, TR-35100 Bornova, Izmir, Turkey*
[b] *Department of Mechanical Engineering, Faculty of Engineering, Dokuz Eylul University, TR-35297 Buca, Izmir, Turkey*

ARTICLE INFO

Article history:
Received 4 October 2013
Received in revised form 2 June 2014
Accepted 16 June 2014
Available online xxxx

Keywords:
Chaotic maps
Metaheuristics
Nonlinear system of equations
Optimization methods
Quantum behaved particle swarm optimization
Root solvers

ABSTRACT

This study proposes a novel chaotic quantum behaved particle swarm optimization algorithm for solving nonlinear system of equations. Different chaotic maps are introduced to enhance the effectiveness and robustness of the algorithm. Several benchmark studies are carried out. Logistic map gives the best results and is utilized in solving nonlinear equation sets. Nine well known problems are solved with our algorithm and results are compared with Quantum Behaved Particle Swarm Optimization, Intelligent Tuned Harmony Search, Gravitational Search Algorithm and literature studies. Comparison results reveal that the proposed algorithm can cope with the highly non-linear problems and outperforms many algorithms which exist in the literature.

1. Introduction

Solving nonlinear sets of equations is a hard and tedious task as these kind of problems appear in many real-world applications. Economics [1], chemistry [2], physics [3–6] are some of the examples of scientific areas applied to solve these type of equations. Newton-type methods which are a conventional procedure for solving system of nonlinear equations are derivative based and depend on the sensitivity of the initial value. These are the main drawbacks of Newton's method since some of the derivatives do not come into existence and wrong initial guesses will lead to unexpected results. Bader [7] claimed that the Newton–Raphson method is incapable of solving a large scale system due to its high memory requirements. Bader [7] also proposed a tensor method utilizing Krylov subspace methods for solving nonlinear sets. Two different solution strategies called interval and continuation methods have been proposed for this task. Interval methods [8–18] are robust and tend to slow [19]. Continuation methods [20,21] are more effective for problems which has lower total degrees [19]. Plenty of research activities devoted to a successful solution on this problem have been made, however there is still room to improve existing studies.

New approaches on this issue arise from metaheuristic applications. Karr et al. [22] hybridized the Genetic algorithm and Newton's method for solving a nonlinear system of equations. The local search procedure is maintained by the Genetic algorithm which supplies initial values of Newton's method. Wang et al. [23] proposed a modified particle swarm equation with controller to improve the search dynamics of traditional PSO. Ouyang et al. [24] hybridized the Nelder–Mead algorithm with Particle Swarm Optimization to solve systems of nonlinear equations. Selection of good initial values of the Nelder–Mead

* Corresponding author.
E-mail address: oeturgut@hotmail.com (O.E. Turgut).

http://dx.doi.org/10.1016/j.camwa.2014.06.013
0898-1221/

algorithm is supplied by the particle swarm method which suffers from being trapped by local minima. Jia and He [25] combined Artificial Bee Colony with Particle Swarm Optimization to improve the effectiveness of the search mechanism. Yang et al. [26] hybridized the Hooke–Jeeves algorithm with the Glow-worm Swarm Optimization algorithm to speed up the local search procedure. Results showed that the modified algorithm has high convergence rate and accuracy for solving nonlinear equations. Hirsch et al. [27] proposed the Continuous Greedy Randomized Adaptive Search Procedure (C-GRASP) for solving nonlinear set of equations. Mo et al. [28] postulated Conjugate Direction Particle Swarm Optimization (CDPSO) which introduces the conjugate direction method into particle swarm optimization in order to enhance the ability of solving high dimensional optimization problems. Toutounian et al. [29] proposed a hybrid scheme which combines the Electromagnetic Metaheuristic method with the finite difference version of the Newton–GMRES method. Sacco and Henderson [30] introduced a novel search strategy that combines Luus–Jaakola random search with Fuzzy Clustering Means. Promising solution areas are obtained by this procedure and the Nelder–Mead algorithm is applied on these areas to reach an optimum solution. Moreover, the Genetic algorithm [31–35], Evolutionary algorithm [36,37], Firefly algorithm [38], Artificial Bee Colony algorithm [39], Invasive weed optimization [40], Imperialist competitive algorithm [41] and Particle Swarm Optimization [42,43] are also utilized for sorting out nonlinear systems of equations.

There is an increased interest on advances in applications of nonlinear dynamics, generally, on the use of chaos in optimization algorithms. Randomly generated chaotic sequences have been incorporated with majority of the metaheuristics to upgrade the probing potential of the mentioned algorithms such as Bee Colony algorithm [44], Bat algorithm [45], Harmony search [46], Krill Herd algorithm [47], Firefly algorithm [48], Imperialist competitive algorithm [49], Genetic algorithms [50], Simulated annealing [51], Ant colony optimization [52,53], Big Bang–Big Crunch algorithm [54] and Particle swarm optimization [55–60]. In order to solve this high dimensional optimization problem, we propose the Chaotic Quantum behaved Particle Swarm Optimization method. We used different chaotic maps to replace random parameters which exist in QPSO. In this way, different pseudorandom number sequences have been generated and search capacity of the algorithm has increased. To test the effectiveness of the proposed algorithms, we applied some benchmark functions including Colville, Schaffer, Griewank, Rastrigin, Dropwave and Rosenbrock functions. Chaotic map which gives the best results will be selected and utilized in solving nonlinear system of equations.

System of non linear equations can be described as

$$\begin{aligned} f_1(x_1, x_2, x_3, \ldots, x_n) &= 0.0 \\ f_2(x_1, x_2, x_3, \ldots, x_n) &= 0.0 \\ &\vdots \\ f_m(x_1, x_2, x_3, \ldots, x_n) &= 0.0, \end{aligned} \tag{1}$$

where $f_i, i = 1, 2, \ldots, m$ is a nonlinear equation system and $X = (x_1, x_2, \ldots, x_n)$ is the unknown solution vector. The problem is transformed into optimization given as

$$\text{Minimize } F(X) = \sqrt{\sum_{i=1}^{m} f_i^2}. \tag{2}$$

The paper is organized as follows. Section 2 describes Particle Swarm and Quantum behaved Particle Swarm Optimization methods, Section 3 introduces the chaotic maps which exist in the literature, Section 4 gives the CQPSO method proposed for solving the system of equations, in Section 5 numerical tests are studied and Section 6 gives the conclusion of this research.

2. Particle swarm optimization

2.1. Basics of particle swarm optimization

Particle Swarm Optimization [61] is a population based algorithm constructed on swarm behaviour of fish schooling and bird flocking to find an optimum solution of a problem. The algorithm has some similarities with Genetic algorithms and ant algorithms; however it is much simpler since it does not need crossover or mutation operator [62]. Each candidate solution named as "particle" flies around the solution space and lands on the optimal position. Particles in the swarm adjust their position by their own experience and experience of neighbouring particles (agents). Each agent has a memory which keeps track of its previous best position P_{best} with its respective fitness value. Agent with the best fitness value in the swarm is called Global best G_{best} which holds the optimum solution for this generation. Assume a swarm with N dimensional population with D dimensional particles. Position and velocity vectors of the ith particle is represented with $x_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$ and $v_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$ where $i = 1, 2, 3, \ldots, N$. Position and velocity update are maintained by

$$v_i^{k+1} = w^k v_i^k + c_i r_1 (P_{bi}^k - x_i^k) + c_2 r_2 (P_g^k - x_i^k) \tag{3}$$

$$x_i^{k+1} = x_i^k + v_i^{k+1}, \tag{4}$$

Please cite this article in press as: O.E. Turgut, et al., Chaotic quantum behaved particle swarm optimization algorithm for solving nonlinear system of equations, Computers and Mathematics with Applications (2014), http://dx.doi.org/10.1016/j.camwa.2014.06.013

where w is inertia weight described as

$$w^k = w_{\max} - \left(\frac{w_{\max} - w_{\min}}{k_{\max}}\right) k, \tag{5}$$

where $w_{\max} = 0.9$ and $w_{\min} = 0.4$ [63] and $k_{\max}$ is the maximum number of iterations (generation) and k is the current iteration (generation); $c_1 = 2.0$ and $c_2 = 2.0$ are the cognitive and social parameters; r_1 and r_2 are uniformly distributed random numbers generated between 0.0 and 1.0; P_{bi} stands for the best position found until generation k and P_g represents the best position in the swarm at generation k. The pseudocode for the traditional PSO algorithm is given as follows:

Step 1 Initialize each particle in the swarm with random values.
Step 2 Calculate the fitness value of each particle.
Step 3 Update velocity and position of the particle according to (3) and (4).
Step 4 Compare the fitness value of each particle with its respective P_{best}. If corresponding fitness value is better than previous P_{best}, then update P_{best}. If current P_{best} is better than G_{best} than update G_{best}.
Step 5 Repeat Step 3 and Step 4 until termination criterion is met.

2.2. Quantum behaved particle swarm optimization

Being trapped of local optima is the main drawback of the traditional PSO algorithm. Quantum behaved Particle Swarm Optimization (QPSO) was proposed by Sun et al. [64,65] to avoid these disadvantages. In QPSO, movements of the particles are completely different. Newtonian principles are invalid in quantum world as velocity and position update cannot be determined simultaneously. New state of the particle is determined by wave function $\psi(x, t)$ [66]. Also there exists $|\psi|^2$ which is the probability density function of the position of the particle [67]. Trajectory analyses in [68] showed that convergence of the PSO algorithm is provided if each particle in the swarm converges to its local attractor defined as

$$p_{i,j}^k = \left(\phi P_{i,j}^k + (1.0 - \phi) P_{g,j}^k\right) \tag{6}$$

where $i = 1, 2, 3, \ldots, N$ and $j = 1, 2, \ldots, D$; $\phi = (c_1 r_1^k)/(c_1 r_1^k + c_2 r_2^k)$ with corresponding uniformly distributed numbers of r_1 and r_2. c_1 and c_2 are cognitive and social parameters as described in the PSO section. With the concept of Monte Carlo, new position of the particle is determined by the following equations [69–71]

$$\begin{aligned} x_{i,j}^k &= p_{i,j} + \beta |M_{best,i} - x_{i,j}^k| \ln 1/u \quad \text{if } rnd \geq 0.5 \\ x_{i,j}^k &= p_{i,j} - \beta |M_{best,i} - x_{i,j}^k| \ln 1/u \quad \text{if } rnd \leq 0.5 \end{aligned} \tag{7}$$

where $x_{i,j}$ is the position of the jth dimension of the ith particle; β is a contraction–expansion coefficient [65] which can be tuned to accelerate the convergence of the algorithm; u and rnd are uniformly distributed random numbers between 0.0 and 1.0; $p_{i,j}$ is the local attractor defined in (6); M_{best} is the mean of the P_{best} position of particles in the swarm. M_{best} can be formulated as

$$\begin{aligned} M_{best}^k &= (M_{best,1}^k, M_{best,2}^k, \ldots, M_{best,j}^k, \ldots, M_{best,D}^k) \\ &= \left(\frac{1}{N}\sum_{i=1}^{N} P_{i,1}^k, \frac{1}{N}\sum_{i=1}^{N} P_{i,2}^k, \ldots, \frac{1}{N}\sum_{i=1}^{N} P_{i,j}^k, \ldots, \frac{1}{N}\sum_{i=1}^{N} P_{i,D}^k\right). \end{aligned} \tag{8}$$

Pseudocode of the QPSO algorithm is explained as follows [71–73]

Step 1 Initialize particles in the population with random position vectors.
Step 2 Evaluate the fitness of each particle.
Step 3 Calculate M_{best} vector according to (8).
Step 4 Compare the fitness of each particle with P_{best} value. If current fitness value is better than P_{best} then set current fitness value to P_{best} value and apply P_{best} as a current location in D-dimensional space.
Step 5 Compare P_{best} values with G_{best}. If P_{best} values are better then G_{best}, replace G_{best} with current P_{best}.
Step 6 Update the position of the particles according to (7).
Step 7 Repeat Step 2 to Step 6 until termination criteria is met.

3. Chaotic maps

Chaos is a deterministic and random-like process exists in non linear and dynamical systems, depends on the initial conditions and includes infinite periodic motions [74]. Many researchers [75–81] combined chaotic systems with optimization algorithms to enhance the search efficiency and prevent from premature convergence to local minima. Due to the non-repetition of the chaos, search speed of the chaotic algorithms are generally faster than stochastic ones. In this section, one dimensional and non-invertible maps are used to build up chaotic sequences. Here we offer some well known chaotic maps found in the literature.

Please cite this article in press as: O.E. Turgut, et al., Chaotic quantum behaved particle swarm optimization algorithm for solving nonlinear system of equations, Computers and Mathematics with Applications (2014), http://dx.doi.org/10.1016/j.camwa.2014.06.013

3.1. Chebyshev map

Chebyshev map is represented as [82]

$$x_{t+1} = \cos\left(t \arccos(x_t)\right). \tag{9}$$

3.2. Circle map

Circle map is defined as the following representative equation [83]

$$x_{t+1} = x_t + b - (a - 2\pi)\sin(2\pi x_t) \bmod (1), \tag{10}$$

where $a = 0.5$ and $b = 0.2$.

3.3. Gauss/mouse map

The Gauss map consists of two sequential parts defined as [84]

$$x_{t+1} = \begin{cases} 0, & \text{if } x_t = 0. \\ 1/x_t, & \text{else } \bmod (1), \end{cases} \tag{11}$$

where $\frac{1}{x_t} \bmod (1) = \frac{1}{x_t} - \lfloor \frac{1}{x_t} \rfloor$.

3.4. Intermittency map

The intermittency map [85] is formed with two iterative equations and represented as

$$x_{t+1} = \begin{cases} \varepsilon + x_t + cx_t^n, & \text{if } 0 < x_t \leqslant P. \\ \dfrac{x_t - P}{1 - P}, & \text{elseif } P < x_t < 1, \end{cases} \tag{12}$$

where $c = \frac{1-\varepsilon-P}{P^2}$, $n = 2.0$, and ε is very close to zero.

3.5. Iterative map

The iterative chaotic map with infinite collapses [86] is defined with the following as

$$x_{t+1} = \sin\left(\frac{a\pi}{x_t}\right), \tag{13}$$

where $a \in (0, 1)$.

3.6. Liebovitch map

The proposed chaotic map [82] can be defined as

$$x_{t+1} = \begin{cases} \alpha x_t, & 0 < x_t \leq P_1 \\ \dfrac{P_2 - x_t}{P_2 - P_1}, & P_1 < x_t \leq P_2 \\ 1 - \beta(1 - x_t), & P_2 < x_t < 1, \end{cases} \tag{14}$$

where

$$\alpha = \frac{P_2\left(1 - (P_2 - P_1)\right)}{P_1} \tag{15a}$$

$$\beta = \frac{((P_2 - 1) - P_1(P_2 - P_1))}{P_2 - 1}. \tag{15b}$$

3.7. Logistic map

Logistic map [87] demonstrates how complex behaviour arises from a simple deterministic system without the need of any random sequence. It is based on a simple polynomial equation which describes the dynamics of biological population [88].

$$x_{t+1} = cx_t(1 - x_t), \tag{16}$$

where $x_0 \in (0, 1)$, $x_0 \notin \{0.0, 0.25, 0.50, 0.75, 1.0\}$ and when $c = 4.0$ a chaotic sequence is generated by the Logistic map.

3.8. Piecewise map

Piecewise map [86] can be formulated as follows

$$x_{t+1} = \begin{cases} x_t/P, & 0 < x_t < P \\ \dfrac{x_t - P}{0.5 - P}, & P \le x_t < 0.5 \\ \dfrac{(1 - P - x_t)}{0.5 - P}, & 0.5 < x_t < 1 - P \\ \dfrac{(1 - x_t)}{P}, & 1 - P < x_t < 1, \end{cases} \tag{17}$$

where $P \in (0, 0.5)$ and $x \in (0, 1)$.

3.9. Sine map

Sine map [89] can be described as

$$x_{t+1} = \frac{a}{4}\sin(\pi x_t), \tag{18}$$

where $0 < a \le 4$.

3.10. Singer map

One dimensional chaotic Singer map [90] is formulated as

$$x_{t+1} = \mu(7.86x_t - 23.31x_t^2 + 28.75x_t^3 - 13.302875x_t^4), \tag{19}$$

where $\mu \in (0.9, 1.08)$.

3.11. Sinusoidal map

Sinusoidal map [88] is generated as the following equation

$$x_{t+1} = ax_t^2 \sin(\pi x_t), \tag{20}$$

where $a = 2.3$.

3.12. Tent map

Tent map [91] is defined by the following iterative equation

$$x_{t+1} = \begin{cases} x_t/0.7, & x_t < 0.7 \\ \dfrac{10}{3}(1.0 - x_t), & x_t \ge 0.7. \end{cases} \tag{21}$$

4. Proposed chaotic quantum behaved particle swarm optimization

To improve the convergence ability of the QPSO algorithm, Coelho [72] proposed a novel chaotic quantum behaved particle swarm optimization approach based on Zaslavskii map [92]. In this work, uniformly distributed random parameters were modified and replaced with chaotic sequences. In another work, Sun et al. [93] introduced Gaussian distributed local attractor point. In this method, an originally defined local attractor point was selected as mean of the distribution and standard deviation of the distribution is chosen as the distance between the mean best and personal best particle. New local attractor point was formulated as

$$np_{i,j}^k = N(p_{i,j}^k, M_{best}^k - P_i^k). \tag{22}$$

In our study, we combine these two novel strategies and offer a new position update procedure described as in Algorithm 1, where the mutation rate is defined as in [94]

$$p_m^k = 0.4\left(1 - \frac{k}{k_{\max}}\right). \tag{23}$$

Algorithm 1 Proposed perturbation scheme

if $r > p_m^k$ **then**
$x_{i,j}^k = np_{i,j}^k + \beta|M_{best,j} - x_{i,j}^k|\ln(\frac{1}{u})$ *if rnd* ≥ 0.5
$x_{i,j}^k = np_{i,j}^k - \beta|M_{best,j} - x_{i,j}^k|\ln(\frac{1}{u})$ *if rnd* < 0.5
else
$x_{i,j}^k = np_{i,j}^k + \beta|M_{best,j} - x_{i,j}^k|\ln(\frac{1}{\varphi_{i,j}})$ *if rnd* ≥ 0.5
$x_{i,j}^k = np_{i,j}^k - \beta|M_{best,j} - x_{i,j}^k|\ln(\frac{1}{\varphi_{i,j}})$ *if rnd* < 0.5
end if

In (23) k is the current iteration and $k_{\max}$ is the maximum iteration number. β is a controllable contraction–expansion coefficient as proposed in [95]

$$\beta = \beta_0 + ((k_{\max} - k)(\beta_1 - \beta_0)/k_{\max}), \tag{24}$$

where β_0 and β_1 are set to 1.0 and 0.5, respectively. u, r and rnd are uniformly distributed random numbers between 0.0 and 1.0. $\varphi_{i,j}$ is the chaotic variable as proposed in [72]. We aim to improve the quality of the solution by using different chaotic maps introduced in Section 3. We compare the success rate of each chaotic map and a chaotic map which gives the best results is chosen to be utilized in our algorithm. Our proposed algorithm is benchmarked with well known test functions. These functions are listed in Table 1. The algorithm is run for 100 times for each chaotic map and maximum iteration is set to 3000. From our past experiences, when population size is between 20 and 30, the algorithm converges faster so we suggest $N = 25$ for population size. To evaluate performance of each chaotic map, a success rate is defined as

$$S_r = 100\frac{N_{successful}}{N_{all}}, \tag{25}$$

where N_{all} is the number of all trials (which is equal to 100 for this study) and $N_{successful}$ shows the number of solutions which are found to be successful. Successful run criteria is offered as

$$|X^* - X_{globalopt}| \leq (UB - LB) \times 10^{-5}. \tag{26}$$

X^* is the global best solution obtained by the proposed algorithm; $X_{globalopt}$ is the global optimum of the benchmark function; UB and LB represent upper and lower bounds of search space. Tables 2–7 gives the statistical results for different chaotic maps. Success rates of the chaotic maps are presented in Table 8. As seen in Table 8, however all chaotic maps fail to find global optimum of the Rosenbrock function, the best performance is shown by the Logistic map. Fig. 1 gives the sequence of 500 points generated in the Logistic map.

5. Numerical tests

Nine different benchmark problems for a nonlinear system of equations are utilized to test the effectiveness of the proposed algorithm. 100 consecutive algorithm runs are performed for each case study and best results are compared with obtained solutions of Quantum behaved Particle Swarm Optimization (QPSO) [64,65], Gravitational Search algorithm (GRAV) [96], Intelligent Tuned Harmony Search algorithm [97] and literature studies. For each algorithm, maximum number of iterations (generation) is set to 8000. Considered values for parameters used in ITHS are harmony memory size = 20 and harmony memory consideration rate = 0.95; in GRAV are population size = 20; in QPSO are cognitive factor (c_1) and social factor (c_2). Parameters used in our proposed algorithm (L-QPSO) are the same as used in the QPSO algorithm. Algorithm runs were performed in Java and executed on Intel Core with 2.50 GHz CPU and 6.0 GB RAM. The statistical results in terms of minimum value, standard deviation value, mean deviation value and maximum value for each algorithm for each case study are reported in Table 9. Table 9 shows the accuracy and robustness of the proposed algorithm since it outperforms other algorithms in most of cases with regard to the minimum objective function values.

5.1. Case study 1

The problem has been studied by many researchers [98–100] before. The non linear set is defined as

$$\begin{aligned} &2x_1 + x_2 + x_3 + x_4 + x_5 = 0.0 \\ &x_1 + 2x_2 + x_3 + x_4 + x_5 = 0.0 \\ &x_1 + x_2 + 2x_3 + x_4 + x_5 = 0.0 \\ &x_1 + x_2 + x_3 + 2x_4 + x_5 = 0.0 \\ &x_1x_2x_3x_4x_5 - 1.0 = 0.0, \end{aligned} \tag{27}$$

where $-2 \leq x_i \leq 2, i = 1, 2, \ldots, 5$. Table 10 gives the optimum results obtained by our proposed algorithm (L-QPSO) and compares the results with other metaheuristic algorithms and literature studies. L-QPSO surpasses other algorithms

Please cite this article in press as: O.E. Turgut, et al., Chaotic quantum behaved particle swarm optimization algorithm for solving nonlinear system of equations, Computers and Mathematics with Applications (2014), http://dx.doi.org/10.1016/j.camwa.2014.06.013

Table 1
Benchmark functions used to test the performance of the algorithms.

Function name	Definition	Bounds	Optimum
Colville	$f(x) = (x_1^2 - x_2)^2 + (x_1 - 1)^2 + (x_3 - 1)^2 + 90.0(x_3^2 - x_4)^2 + 10.1\left((x_3 - 1)^2 + (x_4 - 1)^2\right) + (19.8(1.0/x_2)(x_4 - 1.0))$	$x_i \in [-10, 10]^4$	0.0
Schaffer	$f(x) = 0.5 + \frac{\sin^2\left(\sqrt{\sum_{i=1}^N x_i^2}\right) - 0.5}{\left(1.0 + 0.001\sqrt{\sum_{i=1}^N x_i^2}\right)^2}$	$x_i \in [-100, 100]^{30}$	0.0
Griewank	$f(x) = \sum_{i=1}^N \frac{x_i^2}{4000} - \prod_{i=1}^N \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1.0$	$x_i \in [-600, 600]^{30}$	0.0
Rastrigin	$f(x) = 10N + \sum_{i=1}^N \left(x_i^2 - 10cos(2\pi x_i)\right)$	$x_i \in [-5.12, 5.12]^{30}$	0.0
Dropwave	$f(x) = -\frac{1 + \cos\left(12\sqrt{\sum_{i=1}^N x_i^2}\right)}{2 + 0.5\sum_{i=1}^N x_i^2}$	$x_i \in [-5.12, 5.12]^{40}$	−1.0
Rosenbrock	$f(x) = \sum_{i=1}^{N-1} \left(100(x_i^2 - x_{i+1}) + (x_i - 1)^2\right)$	$x_i \in [-2.048, 2.048]^{40}$	0.0

Table 2
Statistical results for the Colville function for different chaotic maps ($D = 4$).

	Best	Mean dev.	Median	Worst	Std. dev.
Bernoulli map	0.0	0.490236134	3.5659683E−23	3.676000135	1.24955949
Chebyshev map	0.0	1.470400054	1.9514631E−28	3.676000135	1.80008649
Circle map	2.292627005E−31	1.017091114	1.1692355E−31	3.676000135	1.64118588
Gauss map	3.989085941E−27	0.736982038	6.0104385E−17	3.676000135	1.46954001
Intermittency map	5.256469377E−18	0.105753149	9.8658053E−12	1.719880324	0.31989648
Iterative map	0.0	1.105487375	1.8885926E−13	3.676000135	1.68284624
Liebovitch map	0.0	0.919182294	1.8112336E−26	3.676000135	1.59164976
Logical map	0.0	6.9120900E−4	1.4074886E−19	0.010746897	0.00250243
Piecewise map	0.0	1.0675576817	9.5843839E−19	3.676000135	1.66836634
Sine map	9.502815428E−23	0.3670001377	1.1488826E−18	3.676000135	1.10280004
Singer map	1.472951221E−30	1.2085120997	1.4652627E−18	16.09334654	2.37072769
Sinusoidal map	1.919291170E−24	0.1265737089	8.0008974E−16	3.676000135	0.65924158
Tent map	0.0	0.3679444317	5.3261983E−21	3.676000135	1.10268670

Table 3
Statistical results for the Schaffer function for different chaotic maps ($D = 30$).

	Best	Mean dev.	Median	Worst	Std. dev.
Bernoulli map	0.0	0.0	0.0	0.0	0.0
Chebyshev map	0.0	9.429406E−5	0.0	0.01225822	0.00107097
Circle map	0.0	0.0	0.0	0.0	0.0
Gauss map	0.0	0.0	0.0	0.0	0.0
Intermittency map	0.0	0.0	0.0	0.0	0.0
Iterative map	0.0	0.0	0.0	0.0	0.0
Liebovitch map	0.0	0.0	0.0	0.0	0.0
Logical map	0.0	0.0	0.0	0.0	0.0
Piecewise map	0.0	0.0	0.0	0.0	0.0
Sine map	0.0	0.0	0.0	0.0	0.0
Singer map	0.0	1.9609755E−6	0.0	7.82234358E−5	1.0940729E−5
Sinusoidal map	0.0	0.0	0.0	0.0	0.0
Tent map	0.0	0.0	0.0	0.0	0.0

Table 4
Statistical results for the Griewank function for different chaotic maps ($D = 30$).

	Best	Mean dev.	Median	Worst	Std. dev.
Bernoulli map	0.0	0.0	0.0	0.0	0.0
Chebyshev map	0.0	0.0	0.0	0.0	0.0
Circle map	0.0	0.006047	0.0	0.07843011	0.0161414
Gauss map	0.0	0.0	0.0	0.0	0.0
Intermittency map	0.0	0.0	0.0	0.0	0.0
Iterative map	0.0	0.0	0.0	0.0	0.0
Liebovitch map	0.0	0.0	0.0	0.0	0.0
Logical map	0.0	0.0	0.0	0.0	0.0
Piecewise map	0.0	0.0	0.0	0.0	0.0
Sine map	0.0	0.0	0.0	0.0	0.0
Singer map	0.0	0.0363053	0.0	0.44072011	0.10933143
Sinusoidal map	0.0	0.0	0.0	0.0	0.0
Tent map	0.0	0.0	0.0	0.0	0.0

Please cite this article in press as: O.E. Turgut, et al., Chaotic quantum behaved particle swarm optimization algorithm for solving nonlinear system of equations, Computers and Mathematics with Applications (2014), http://dx.doi.org/10.1016/j.camwa.2014.06.013

Table 5
Statistical results for the Rastrigin function for different chaotic maps ($D = 30$).

	Best	Mean dev.	Median	Worst	Std. dev.
Bernoulli map	0.0	0.0	0.0	0.0	0.0
Chebyshev map	0.0	0.0	0.0	0.0	0.0
Circle map	0.0	0.0	0.0	0.0	0.0
Gauss map	0.0	0.0	0.0	0.0	0.0
Intermittency map	0.0	0.0	0.0	0.0	0.0
Iterative map	0.0	0.0	0.0	0.0	0.0
Liebovitch map	0.0	0.0	0.0	0.0	0.0
Logical map	0.0	0.0	0.0	0.0	0.0
Piecewise map	0.0	0.0	0.0	0.0	0.0
Sine map	0.0	0.0	0.0	0.0	0.0
Singer map	0.0	0.0020835	0.0	0.0720111	0.00940741
Sinusoidal map	0.0	0.0	0.0	0.0	0.0
Tent map	0.0	0.0	0.0	0.0	0.0

Table 6
Statistical results for the Dropwave function for different chaotic maps ($D = 40$).

	Best	Mean dev.	Median	Worst	Std. dev.
Bernoulli map	0.0	0.0	0.0	0.0	0.0
Chebyshev map	0.0	0.0	0.0	0.0	0.0
Circle map	0.0	0.0	0.0	0.0	0.0
Gauss map	0.0	0.0	0.0	0.0	0.0
Intermittency map	0.0	0.0	0.0	0.0	0.0
Iterative map	0.0	0.0	0.0	0.0	0.0
Liebovitch map	0.0	0.0	0.0	0.0	0.0
Logical map	0.0	0.0	0.0	0.0	0.0
Piecewise map	0.0	0.0	0.0	0.0	0.0
Sine map	0.0	0.0	0.0	0.0	0.0
Singer map	0.0	0.0020835	0.0	0.0720111	0.00940741
Sinusoidal map	0.0	0.0	0.0	0.0	0.0
Tent map	0.0	0.0	0.0	0.0	0.0

Table 7
Statistical results for the Rosenbrock function for different chaotic maps ($D = 40$).

	Best	Mean dev.	Median	Worst	Std. dev.
Bernoulli map	38.768427207	38.904376523	38.96440991	38.98012190	0.0924442267
Chebyshev map	38.767855546	38.930365080	38.97303293	38.98701457	0.0850291621
Circle map	38.770899007	38.913226681	38.96436054	38.98021885	0.0878102193
Gauss map	38.768982668	38.921835351	38.96196331	38.97591042	0.0792199891
Intermittency map	38.091997298	38.194193596	38.10219453	38.74926668	0.2236883141
Iterative map	38.771082584	38.919303291	38.96804147	38.98018380	0.0859967762
Liebovitch map	38.759391245	38.901099963	38.96902508	38.98249237	0.0967058727
Logical map	38.109041115	38.214326174	38.36118870	38.46459314	0.0563424485
Piecewise map	38.769807141	38.878652462	38.95164117	38.97854176	0.0982611122
Sine map	38.100597145	38.502613498	38.75277532	38.96222206	0.3169210387
Singer map	38.771109747	38.941826339	38.96287042	38.98582087	0.0516342961
Sinusoidal map	38.100514213	38.941826332	38.75578333	38.95904437	0.3124624025
Tent map	38.482940459	38.840641014	38.77110420	38.97624045	0.1093704136

Table 8
Success rates of each chaotic map.

	Collvile	Schaffer.	Griewank	Rastrigin	Dropwave	Rosenbrock
Bernoulli map	73	100	100	100	100	0
Chebyshev map	59	99	100	100	100	0
Circle map	66	100	82	100	100	0
Gauss map	79	100	100	100	100	0
Intermittency map	63	100	100	100	100	0
Iterative map	56	100	100	100	100	0
Liebovitch map	62	100	100	100	100	0
Logical map	89	100	100	100	100	0
Piecewise map	67	100	100	100	100	0
Sine map	87	100	100	100	100	0
Singer map	61	94	71	83	74	0
Sinusoidal map	83	100	100	100	100	0
Tent map	81	100	100	100	100	0

Please cite this article in press as: O.E. Turgut, et al., Chaotic quantum behaved particle swarm optimization algorithm for solving nonlinear system of equations, Computers and Mathematics with Applications (2014), http://dx.doi.org/10.1016/j.camwa.2014.06.013

Table 9
Statistical results obtained after 100 algorithm runs.

	Minimum	Std. dev.	Mean dev.	Maximum	Minimum	Std. dev.	Mean dev.	Maximum
	LQPSO				QPSO			
Case study 1	0.00000000000	0.00000000000	0.00000000000	0.00000000000	1.19023842E−4	0.30615475322	0.19685961215	1.7729344742
Case study 2	3.21075372E−8	0.07573068045	0.02546161341	0.75262288489	8.63494937E−4	0.22599042221	0.22843487031	1.3170466056
Case study 3	0.00000000000	4.2459467E−33	1.4061158E−33	2.4651903E−32	0.00000000000	0.00760588253	0.00104357789	0.0736498679
Case study 4	0.00000000000	2.9964896E−73	2.3914591E−74	3.7785053E−72	0.00000000000	0.00696221187	0.00276205661	0.0616852401
Case study 5	0.00000000000	0.00000000000	0.00000000000	0.00000000000	0.00000000000	0.14167213251	0.10273725469	0.5548308830
Case study 6	6.4752361E−16	0.05992980616	0.02497783859	0.25783943858	0.00195450943	2.06200649958	1.46452419391	12.472963781
Case study 7	6.20831492E−6	2.06229319E−4	6.44202906E−4	0.00113727036	9.62956081E−5	0.46025666089	0.31986584814	2.6750069648
Case study 8	2.6020852E−18	3.1857439E−15	8.4696820E−16	1.9212860E−14	1.6654248E−16	0.17567273405	0.20739653915	0.7353425881
Case study 9	1.6298215E−10	102.025240148	168.705483779	551.460632344	0.18607674831	176.052610218	228.515925917	767.75796004
	GRAV				ITHS			
Case study 1	6.71035625E−4	0.00513514102	0.00621692452	0.01728211133	1.41577466E−8	0.016631394416	0.01236940795	0.1153152004
Case study 2	0.00285697754	8.67200120E−4	0.00466824241	0.00626278271	8.88941958E−4	0.063627454049	0.08843564918	0.3014222498
Case study 3	1.1483009E−10	7.16294837E−8	6.73570062E−8	3.07558863E−7	0.00000000000	0.002148256574	4.67800603E−4	0.0182132216
Case study 4	4.15911012E−8	2.97523498E−4	2.55074433E−4	8.30651132E−4	7.39570981E−5	0.007209738538	0.00689635827	0.0541546139
Case study 5	2.32151678E−5	9.27579035E−5	1.64723787E−4	3.81861144E−4	0.00864367972	0.012961479837	0.02848510526	0.0882327076
Case study 6	2.00457355E−4	0.01458581301	0.01357994356	0.08536831725	0.10355348915	0.186844709848	0.39870851563	1.2277557987
Case study 7	5.34548982E−4	0.00159989262	0.00264903045	0.00560918745	3.77206677E−4	0.023264433757	0.02010433014	0.1620341415
Case study 8	2.87200515E−4	0.08897540132	0.13831108452	0.29177988077	0.02345292754	0.071914482939	0.15654934659	0.4159781524
Case study 9	7.47089935933	17.8057247481	37.1556293303	130.470752028	4.15504414694	338.2946656533	312.651184712	5555.0798892

Table 10
Optimum results for case study 1.

	LQPSO		QPSO	ITHS	GRAV	Jäger and Ratz [99]		
x_1	1.000000000000	0.91635458253338494	1.000144199216134	0.999999983928838	0.916566028481882	0.91635458253386	−0.5790430884942	1.000000000000
x_2	1.000000000000	0.91635458253338494	1.000120111355508	0.999999984261559	0.916262407670433	0.916354582533851	−0.5790430884941	1.000000000000
x_3	1.000000000000	0.91635458253338494	1.000131686228283	0.999999984646038	0.916981249858047	0.916354582533851	−0.5790430884941	1.000000000000
x_4	1.000000000000	0.91635458253338494	1.000129832805469	0.999999984062309	0.916141646585271	0.916354582533851	−0.5790430884941	1.000000000000
x_5	1.000000000000	1.41822708733075541	0.999369060482274	1.000000076182086	1.417478352612262	1.41822708733076	8.89521544247056	1.000000000000
f_1	0.000000000000	0.00000000000000000	0.000039089303800	−0.02990331537E−7	−0.00428631022E−3	0.33750779948E−13	−0.1403321903E−12	0.000000000000
f_2	0.000000000000	0.00000000000000000	0.000015001443173	−0.02657611024E−7	−0.30790712167E−3	0.23980817331E−13	−0.0399680288E−12	0.000000000000
f_3	0.000000000000	0.00000000000000000	0.000026576315949	−0.02273131903E−7	0.410935065933E−3	0.23980817331E−13	−0.0399680288E−12	0.000000000000
f_4	0.000000000000	0.00000000000000000	0.000024722893135	−0.02856860523E−7	−0.42866820683E−3	0.23980817331E−13	−0.0399680288E−12	0.000000000000
f_5	0.000000000000	0.00000000000000000	−0.00010533819589	0.130808268661E−7	0.053287740231E−3	0.21982418875E−13	0.0617284001E−12	0.000000000000

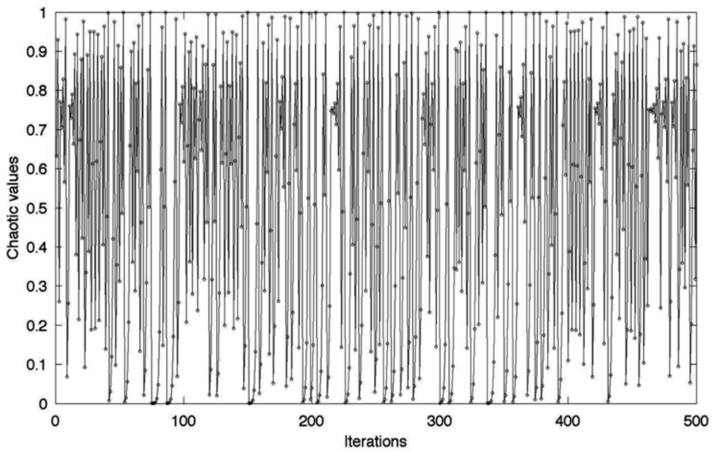

Fig. 1. Evolution of the Logistic map.

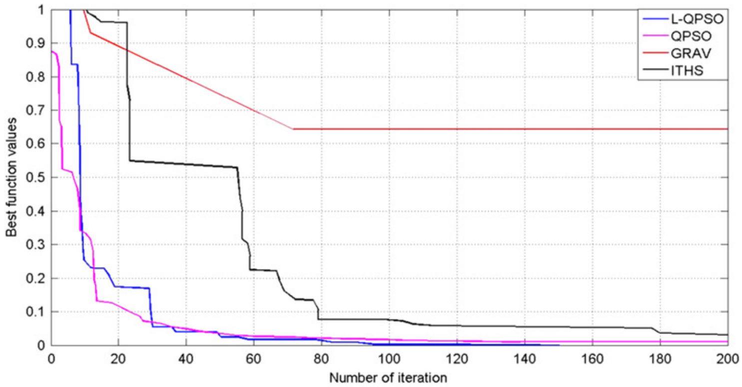

Fig. 2. Convergence history of the algorithms for case study 1.

since it has converged to the optimum solution after 147 iterations while others have not converged within 200 iterations as depicted in Fig. 2.

5.2. Case study 2

The non linear equation set which was studied by Krzyworzcka [101] and Jaberipour et al. [42] is solved by our proposed algorithm. Table 11 compares the optimum results and Fig. 3 serves the convergence history of the cost function. L-QPSO converges to the optimum solution within 1500 iterations whereas others cannot reach the optimum after 2000 iterations

$$\begin{aligned}
&x_1 + \frac{x_2^4 x_4 x_6}{4} + 0.75 = 0.0\\
&x_2 + 0.405\exp(1 + x_1 x_2) - 1.405 = 0.0\\
&x_3 - \frac{x_4 x_6}{2} + 1.5 = 0.0\\
&x_4 - 0.605\exp(1 - x_3^2) - 0.395 = 0.0\\
&x_5 - \frac{x_2 x_6}{2} + 1.5 = 0.0\\
&x_6 - x_1 x_5 = 0.0.
\end{aligned} \tag{28}$$

5.3. Case study 3

A system of four equation sets [102,103] is solved by our proposed algorithm and other algorithms. Best results and convergence history of this case study are shown in Table 12 and Fig. 4, respectively. All algorithms (except GRAV) obtained

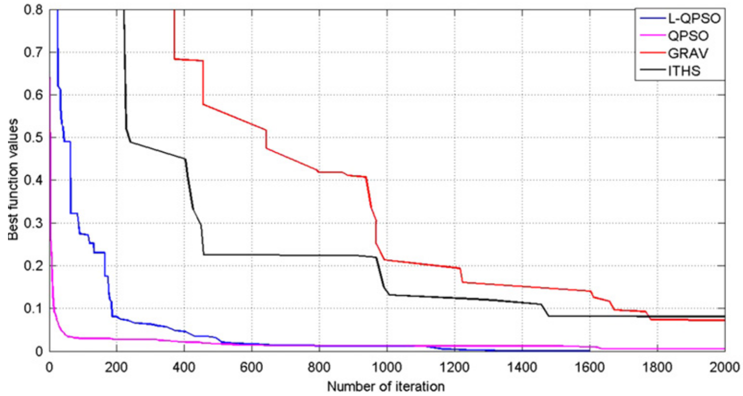

Fig. 3. Convergence history of the algorithms for case study 2.

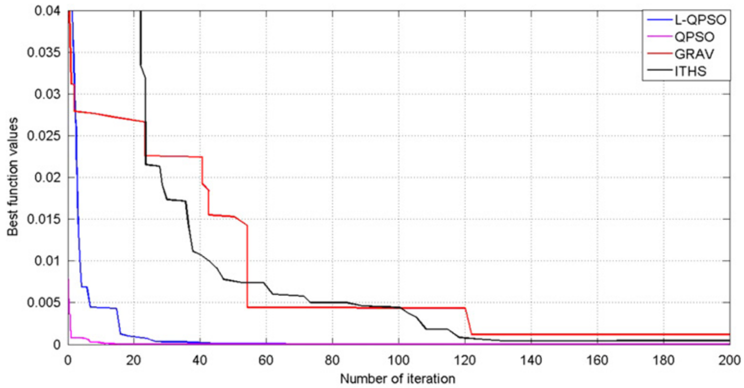

Fig. 4. Convergence history of the algorithms for case study 3.

the global best solution for this case, however, L-QPSO reaches optimum after 113 iterations and surpasses other algorithms in terms of convergence speed.

$$x_i - \cos\left(2x_i - \sum_{j=1}^{4} x_j\right) = 0, \quad 1 \le i \le 4. \tag{29}$$

5.4. Case study 4

Neurophysiology application [21,36,41] is utilized to test the effectiveness and robustness of our algorithm. Problem consists of six nonlinear equations described as

$$\begin{aligned} &x_1^2 + x_3^2 - 1.0 = 0.0 \\ &x_2^2 + x_4^2 - 1.0 = 0.0 \\ &x_5 x_3^3 + x_6 x_4^3 = c_1 \\ &x_5 x_1^3 + x_6 x_2^3 = c_2 \\ &x_5 x_1 x_3^2 + x_6 x_4^2 x_2 = c_3 \\ &x_5 x_1^2 x_3 + x_6 x_2^2 x_4 = c_4, \end{aligned} \tag{30}$$

where $-10 \le x_i \le 10$, $1 \le i \le 6$. In this equation set, c_i values are randomly chosen. These values are set to 0.0 in this study for the sake of simplicity. Table 13 compares the best results found in [36,41] and other algorithms with our algorithm. L-QPSO and QPSO algorithms obtained two different optimum solutions. Some of the solutions they acquired are reported in Table 13. Fig. 5 shows the convergence history of the proposed algorithm. Our proposed algorithm, L-QPSO, has reached the global best solution after 747 iterations.

Table 11
Best results for case study 2.

	LQPSO	QPSO	ITHS	GRAV
x_1	−1.0000007185712	−1.06006024401217	−1.00469360193614	−0.94769257629329
x_2	1.00000045091713	1.03789221092740	1.002699552010811	0.95991034374004
x_3	−0.9999992090087	−0.96490791149742	−0.99763045117792	−1.03187245704251
x_4	1.00000957562795	1.04304617226693	1.002704645251826	0.96116900972767
x_5	−0.99999966616156	−0.96783925322234	−0.99763694114471	−1.03086507223002
x_6	1.000000474541353	1.02588330512720	1.00262777388871	0.97633630789931
f_1	9.037302239889E−8	0.36034275521E−3	−0.63374448056E−3	0.00149483833089
f_2	−2.27255014806E−8	−0.73227698937E−3	−0.28872794332E−3	−0.0018160678696
f_3	7.493891107651E−8	0.07026119979E−3	−0.30021434625E−3	−0.0010845581545
f_4	4.63049154575E−10	−0.14208871243E−3	−0.16590037166E−3	−0.00089244886256
f_5	−1.28890915052E−7	−0.21739907838E−3	−0.30415100062E−3	0.00053726730918
f_6	8.980881838205E−8	−0.08460980828E−3	0.30832206541E−3	−0.00060686822131

Table 12
Best results for case study 3.

	LQPSO	QPSO	ITHS	GRAV	Sharma and Arora [103]
x_1	0.5149332646611294	0.514933264661129	0.514933264661129	0.514931340034248	0.51493326466112941
x_2	0.5149332646611294	0.514933264661129	0.514933264661129	0.514934600978757	0.51493326466112941
x_3	0.5149332646611294	0.514933264661130	0.514933264661129	0.514923716523520	0.51493326466112941
x_4	0.5149332646611294	0.514933264661129	0.514933264661129	0.514944432722957	0.51493326466112941
f_1	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.225940766129E−5	0.0000000000000000
f_2	0.0000000000000000	0.0000000000000000	0.0000000000000000	−0.00704139425E−5	0.0000000000000000
f_3	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.770620340042E−5	0.0000000000000000
f_4	0.0000000000000000	0.0000000000000000	0.0000000000000000	−0.70946910128E−5	0.0000000000000000

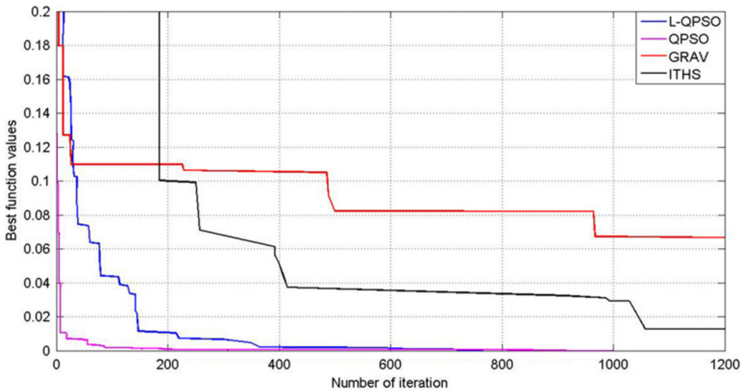

Fig. 5. Convergence history of the algorithms for case study 4.

5.5. Case study 5

Interval arithmetic problem [12,36,104–106] is considered as the benchmark problem. The problem is modelled as

$$\begin{aligned}
&x_1 - 0.25428722 - 0.18324757x_4x_3x_9 = 0.0\\
&x_2 - 0.37842197 - 0.16275449x_1x_{10}x_6 = 0.0\\
&x_3 - 0.27162577 - 0.16955071x_1x_2x_{10} = 0.0\\
&x_4 - 0.19807914 - 0.15585316x_7x_1x_6 = 0.0\\
&x_5 - 0.44166728 - 0.19950920x_7x_6x_3 = 0.0\\
&x_6 - 0.14654113 - 0.18922793x_8x_5x_{10} = 0.0\\
&x_7 - 0.42937161 - 0.21180476x_2x_5x_8 = 0.0\\
&x_8 - 0.07056438 - 0.17081208x_1x_7x_6 = 0.0\\
&x_9 - 0.34504906 - 0.19612740x_{10}x_6x_8 = 0.0\\
&x_{10} - 0.42651102 - 0.21466544x_4x_8x_1 = 0.0.
\end{aligned} \tag{31}$$

Table 13
Best results for case study 4.

	LQPSO	QPSO	ITHS	GRAV	Grosan and Abraham [36]	Abdollahi et al. [41]
x_1	0.446209184554328	−0.796616684320047	0.757992217157792	0.835326847252122	0.045943625	−0.041096050919063
x_2	−0.446209184554328	0.796616684320047	0.757995636725586	0.782860693935276	−0.1626952821	0.041096050919063
x_3	0.894928691918726	−0.604484787453692	0.652290147139058	0.549753670392631	−0.9215324786	0.999155200456294
x_4	−0.894928691918726	0.604484787453692	0.652305698905455	−0.622197020332733	0.9841530788	−0.999155200456294
x_5	0.366779058332292	−0.343529649687506	0.026046699540825	−0.000000012589636	−0.6789794019	0.098733550533454
x_6	0.366779058332292	−0.343529649687506	−0.026009939089497	0.000000014361731	−0.9070329917	0.098733550533454
f_1	0.0000000000000000	0.0000000000000000	0.3463732647923E−4	0.398503403642E−7	0.1489636110	0.0000000000000000
f_2	0.0000000000000000	0.0000000000000000	0.6011011956097E−4	−0.01780246392E−7	0.0049729625	0.0000000000000000
f_3	0.0000000000000000	0.0000000000000000	0.0968608629041E−4	−0.05551106317E−7	0.3332320690	0.0000000000000000
f_4	0.0000000000000000	0.0000000000000000	0.1585609322857E−4	−0.00447429924E−7	0.0038536711	0.0000000000000000
f_5	0.0000000000000000	0.0000000000000000	0.1141785526275E−4	0.011742031683E−7	0.1183698936	0.0000000000000000
f_6	0.0000000000000000	0.0000000000000000	0.1345652755927E−4	−0.10305918922E−7	0.0224932754	0.0000000000000000

Please cite this article in press as: O.E. Turgut, et al., Chaotic quantum behaved particle swarm optimization algorithm for solving nonlinear system of equations, Computers and Mathematics with Applications (2014), http://dx.doi.org/10.1016/j.camwa.2014.06.013

Table 14
Best results for case study 5.

	LQPSO	QPSO	ITHS	GRAV	Grosan and Abraham [36]	Oliveira and Petraglia [106]
x_1	0.25783339370050357	0.25783339370050357	0.254686410312621	0.257839946926554	0.0464905115	0.2578333937005036
x_2	0.38109715460280674	0.38109715460280674	0.378523004753339	0.381079261668136	0.1013568357	0.3810971546028067
x_3	0.27874501734644036	0.27874501734644036	0.276525468374490	0.278737809172705	0.0840577820	0.2787450173464404
x_4	0.20066896422534358	0.20066896422534358	0.201804033260634	0.200676775829179	−0.1388460309	0.2006689642253436
x_5	0.44525142484104163	0.44525142484104163	0.443869219215441	0.445251560409610	0.4943905739	0.4452514248410416
x_6	0.14918391996935457	0.14918391996935457	0.147985685015705	0.149185582343140	−0.0760685163	0.1491839199693546
x_7	0.43200969898372027	0.43200969898372027	0.432376554488803	0.432006811493179	0.2475819110	0.4320096989837203
x_8	0.07340277777624865	0.07340277777624865	0.069871690818600	0.073403712784558	−0.0170748156	0.07340277777624865
x_9	0.34596682687555431	0.34596682687555431	0.349297348759015	0.345965056291278	0.0003667535	0.3459668268755543
x_{10}	0.42732627599329052	0.42732627599329052	0.432318039408281	0.427333090362260	0.1481119311	0.4273262759932905
f_1	0.00000000000000000	0.00000000000000000	−0.003172703109397	0.065250354895E−4	0.2077959240	3.295974604355E−17
f_2	0.00000000000000000	0.00000000000000000	−0.002550893777805	−0.180334006456E−4	0.2769798846	−7.285838599102E−17
f_3	0.00000000000000000	0.00000000000000000	−0.002166747254159	−0.071683799276E−4	0.1876863212	3.122502256758E−17
f_4	0.00000000000000000	0.00000000000000000	0.001185071568625	0.077342307299E−4	0.3367887114	3.599551212651E−17
f_5	0.00000000000000000	0.00000000000000000	−0.001328103066942	0.002122703812E−4	0.0530391321	−3.64291929955E−17
f_6	0.00000000000000000	0.00000000000000000	−0.001092587590357	0.015857612497E−4	0.2223730535	2.949029909160E−17
f_7	0.00000000000000000	0.00000000000000000	0.000518467284375	−0.027980350751E−4	0.1816084752	6.071532165918E−17
f_8	0.00000000000000000	0.00000000000000000	−0.003476285138268	0.008502088621E−4	0.0874896386	−0.607153216591E−17
f_9	0.00000000000000000	0.00000000000000000	0.003371565377805	−0.018071372921E−4	0.3447200366	1.301042606982E−17
f_{10}	0.00000000000000000	0.00000000000000000	0.005036117718070	0.067515256242E−4	0.2784227489	−2.526191061891E−17

Table 15
Best results for case study 6.

	LQPSO	QPSO	ITHS	GRAV	Grosan and Abraham [36]	
x_1	0.590050131899	0.772231269942513	−0.481187883256480	−0.418714637584636	−0.0625820337	−0.1564353525
x_2	0.807366609320	0.635092244414582	0.855827688882007	0.908150863302484	0.7777446281	0.4507122320
x_3	−0.59005013189	0.772345476746086	0.489354396992578	−0.418496945673775	−0.0503725828	0.4622139796
x_4	0.807366609320	0.635464640410570	0.856058114076982	0.908273850895166	0.3805368959	−0.8818348503
x_5	−0.59005013189	−0.771984058288123	0.529021825853160	0.418286659161462	−0.5592587603	−0.6522824284
x_6	0.942931949016	0.501630602442857	−0.413193459556378	−0.267011571005521	−0.6988338865	0.4082826235
x_7	0.266895066252	0.016544744588547	−0.583759143369605	−0.589809505946614	0.3963927675	0.4718261386
x_8	−1.22424777217	−1.199869634113995	−0.817013237700635	−0.998579648272644	0.0861763643	−0.5070478474
f_1	2.22044604E−16	−0.000316706807422	−0.036017187950030	0.0599382446380E−3	0.3911967824	0.7723864643
f_2	0.000000000000	−0.000140305634309	−0.028091241086912	−0.122315944674E−3	0.3925758963	0.5832167209
f_3	0.000000000000	0.000332844662273	−0.027696779466991	0.1010817582113E−3	0.8526542737	0.0087255337
f_4	0.000000000000	−0.000225304536866	0.012699586906051	−0.074882547608E−3	0.5424213097	0.2031050697
f_5	−3.4000580E−16	0.000867394190697	−0.017478713588524	0.0652303303051E−3	0.7742116224	0.6056929403
f_6	−6.0715321E−17	0.001365368860350	−0.064389984564789	0.0321280002644E−3	0.1537105718	0.3663682493
f_7	0.000000000000	0.000799271797368	−0.008180482707589	−0.001702726915E−3	0.9116019977	0.3532359802
f_8	1.11022302E−16	0.000532099949980	−0.056480815532868	0.0226792051423E−3	0.1519175234	0.4646334692

Table 14 gives the optimum results found by mentioned algorithms and literature studies. L-QPSO and QPSO algorithms found the same global best solution whereas L-QPSO has served better convergence performance than other algorithms since it has reached to optimum after 487 iterations while others did not converge even after 800 iterations as shown in Fig. 6.

5.6. Case study 6

The inverse position problem for a six-revolute joint problem application [19,30,36] is taken as the benchmark problem. The problem can be described as

$$x_i^2 + x_{i+1}^2 - 1.0 = 0.0$$
$$a_{1i}x_1x_3 + a_{2i}x_1x_4 + a_{3i}x_2x_3 + a_{4i}x_2x_4 + a_{5i}x_2x_7 + a_{6i}x_5x_8 + a_{7i}x_6x_7 + a_{8i}x_6x_8 + a_{9i}x_1 + a_{10i}x_2 + a_{11i}x_3 + a_{12i}x_4 + a_{13i}x_5 + a_{15i}x_7 + a_{16i}x_8 + a_{17i} = 0.0, \quad (32)$$

where $1 \le i \le 4$. 10 independent non dominated solutions are obtained from our calculations. Table 15 shows best of the independent solutions obtained by L-QPSO, other algorithms and some of the non dominated solutions of Grosan and Abraham [36]. The coefficients utilized in Eq. (32), a_{ji}, where $1 \le j \le 4$ and $1 \le i \le 17$ are given in Table 16. Fig. 7 shows the convergence history of the aforementioned algorithms. As it is portrayed in Fig. 7, L-QPSO converges to the optimum solution after 167 iterations.

Table 16
Parameters for case study 6.

$a_{i,j}$	1	2	3	4
1	−0.24915068	0.125016350	−0.63555007	1.48947730
2	1.609135400	−0.686607360	−0.11571992	0.23062341
3	0.27942343	−0.119228120	−0.66640448	1.32810730
4	1.43480160	−0.719940470	0.11036211	−0.25864503
5	0.00000000	−0.432419270	0.29070203	1.16517200
6	0.40026384	0.000000000	1.2587767	−0.26908494
7	−0.80052768	0.000000000	−0.62938836	0.53816987
8	0.000000000	−0.864838550	0.58140406	0.58258598
9	0.074052388	−0.037157270	0.19594662	−0.20816985
10	−0.083050031	0.035436896	−1.2280342	2.68683200
11	−0.38615961	0.085383482	0.000000000	−0.69910317
12	−0.75526603	0.00000000	−0.079034221	0.35744413
13	0.50420168	−0.039251967	0.026387877	1.24991170
14	−1.0916287	0.00000000	−0.05713143	1.46773600
15	0.00000000	−0.432419270	−1.1628081	1.16517200
16	0.04920729	0.0000000000	1.2587767	1.07633970
17	0.04920729	0.013873010	2.162575	−0.69686809

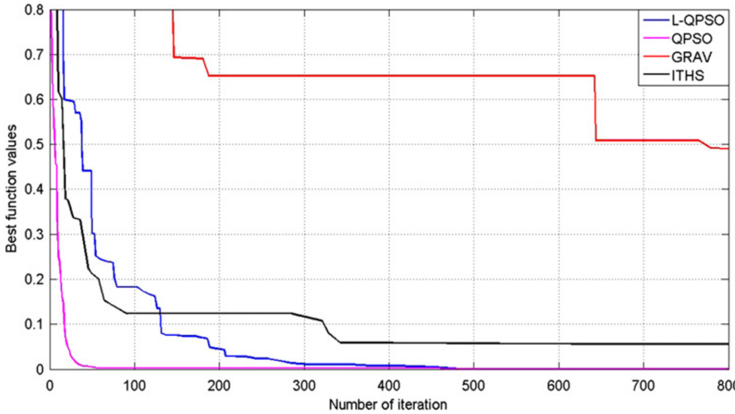

Fig. 6. Convergence history of the algorithms for case study 5.

5.7. Case study 7

This case [19,20,36,106] is interested in the combustion problem which occurred at a temperature of 3000 °C. The non linear set consists of ten equations given as

$$\begin{aligned}
&x_2 + 2x_6 + x_9 + 2x_{10} - 10^{-5} = 0.0\\
&x_3 + x_8 - 3 \times 10^{-5} = 0.0\\
&x_1 + x_3 + 2x_5 + 2x_8 + x_9 + x_{10} - 5 \times 10^{-5} = 0.0\\
&x_4 + 2x_7 - 10^{-5} = 0.0\\
&0.5140437 \times 10^{-7} x_5 - x_1^2 = 0.0\\
&0.1006932 \times 10^{-6} x_6 - x_1^2 = 0.0\\
&0.7816278 \times 10^{-15} x_7 - x_4^2 = 0.0\\
&0.1496236 \times 10^{-6} x_8 - x_1 x_3 = 0.0\\
&0.6194411 \times 10^{-7} x_9 - x_1 x_2 = 0.0\\
&0.2089296 \times 10^{-14} x_{10} - x_1 x_2^2 = 0.0.
\end{aligned} \tag{33}$$

Nine different solutions are obtained by our proposed algorithm for this case study. Table 17 exposes the best result obtained by our algorithm and other algorithms. Grosan and Abraham [25] found eight non dominated solutions for this problem. Oliveira and Petraglia [106] acquired six different solutions using the fuzzy adaptive simulated annealing algorithm. Best results of these studies are also added into Table 17. Fig. 8 shows convergence history of the case study 7. L-QPSO surpasses other algorithms in terms of convergence speed as it has reached the optimum after 1478 iterations.

Please cite this article in press as: O.E. Turgut, et al., Chaotic quantum behaved particle swarm optimization algorithm for solving nonlinear system of equations, Computers and Mathematics with Applications (2014), http://dx.doi.org/10.1016/j.camwa.2014.06.013

Table 17
Optimum results for case study 7.

	LQPSO	QPSO	ITHS	GRAV	Oliveira and Petraglia [106]	Grosan and Abraham [36]	
x_1	−5.9286450E−8	−4.88278463E−7	0.007245878409306	−0.00129788338220	6.0515165980073E−7	−0.0552429896	−0.338378558
x_2	−6.9427900E−5	6.47373030E−3	0.010180176931163	0.009181144808811	−4.115869626410E−3	−0.0023377533	0.0185669333
x_3	−0.298022727	0.98868088610	−0.00290517379654	0.351976543447875	−5.652599019917E−1	0.0455880930	0.0534924988
x_4	−8.8526040E−5	0.00688493552	−0.00232291581807	0.013449786136748	5.846939142550E−3	−0.1287029472	0.0392783417
x_5	−0.4127268522	0.24933059160	−0.00257099288961	0.381060037378405	−1.305864354471E−1	0.0539771728	0.0183882247
x_6	−0.0547120683	−0.00475443378	−0.0001285135424	0.328340515466930	3.613537479299E−2	−0.0151036079	0.0005246892
x_7	4.92534440E−5	−0.00343749623	0.001075271097203	−0.00672201015185	−2.918469098378E−3	0.1063159019	−0.1024269629
x_8	0.2980527304	−0.98865126389	0.002821717052936	−0.35194359192842	5.6528992838770E−1	0.0386267592	0.0500461848
x_9	0.9453385321	0.976976825720	0.000170605626642	−0.15159591940474	−5.400505135447E−1	−0.1144905135	−0.1013361102
x_{10}	−0.417917503	−0.48696584421	−0.00493715824672	−0.25714404311724	2.359528246328E−1	0.0872294353	0.0404252678
f_1	−3.718425434E−8	7.5793361E−12	2.09438979507E−4	−3.18298965591E−5	1.5680469988E−8	0.0274133878	0.0008794626
f_2	3.403141215E−9	−3.77787894E−7	−1.13456743608E−4	2.951519453150E−6	2.63960000269E−8	0.0841848522	0.1035086837
f_3	−1.524300123E−8	3.476367522E−8	2.560031932362E−5	1.215884436503E−4	1.292598473E−10	0.1482418893	0.0955626197
f_4	−1.915259325E−8	−5.69542591E−8	−1.82373623661E−4	−4.23416695693E−7	9.457940001E−10	0.0839188567	0.2441423777
f_5	−2.121593269E−8	1.281644356E−8	5.250288608271E−5	−1.66491312265E−6	−6.7130796532E−9	0.0030517851	0.0011449995
f_6	−1.514954299E−8	−8.38188468E−5	2.072720176124E−4	−1.68553778343E−4	−3.3877126976E−5	0.0000109317	0.0006894619
f_7	−7.836859219E−9	−4.74023371E−5	5.395937897848E−6	−1.80896747124E−4	−3.4186697336E−5	0.0165644486	0.0015427967
f_8	2.196726011E−8	3.348260222E−7	2.105095828321E−5	4.567718475998E−4	4.266486820E−7	0.0025184283	0.0018100789
f_9	5.855288223E−8	6.367894304E−8	−7.37643136604E−5	1.190666480262E−5	3.096222308E−8	0.0001291515	0.0006282589
f_{10}	−5.071595221E−16	2.04623346E−11	7.509338718631E−8	1.094030285101E−7	−1.02510078E−11	0.0000003019	0.0000116649

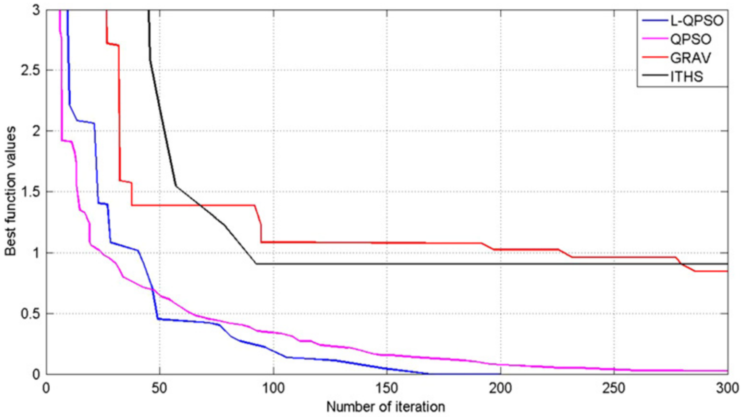

Fig. 7. Convergence history of the algorithms for case study 6.

5.8. Case study 8

Benchmark problem [41,98,107] consists of eight non linear equations described as

$$\begin{aligned}
&4.731\times10^{-3}x_1x_3 - 0.3578x_2x_3 - 0.1238x_1 + x_7 - 1.637\times10^{-3}x_2 - 0.9338x_4 - 0.3571 = 0.0\\
&0.2238x_1x_3 + 0.7623x_2x_3 + 0.2638x_1 - x_7 - 0.007745x_2 - 0.6734x_4 - 0.6022 = 0.0\\
&x_6x_8 + 0.3578x_1 + 4.731\times10^{-3}x_2 = 0.0\\
&-0.7623x_1 + 0.2238x_2 + 0.3461 = 0.0\\
&x_1^2 + x_2^2 - 1.0 = 0.0\\
&x_3^2 + x_4^2 - 1.0 = 0.0\\
&x_5^2 + x_6^2 - 1.0 = 0.0\\
&x_7^2 + x_8^2 - 1.0 = 0.0,
\end{aligned} \tag{34}$$

where $-1 \le x_i \le 1,\ i = 1, 2, \ldots, 8$. There are 16 known solutions for this problem as reported in [98]. Table 18 records the best solutions obtained from L-QPSO, QPSO, GRAV, ITHS algorithms and other researchers. Fig. 9 gives the convergence history of the case study 8. L-QPSO has reached the optimum solution after 219 iterations and surpasses other algorithms concerning the convergence speed.

5.9. Case study 9

Sizing of a thin wall rectangle girder section [28,41,42,108,109] is considered as the benchmark problem. The problem can be defined as

$$\begin{aligned}
f_1(x) &= bh - (b-2t)(h-2t) = 165\\
f_2(x) &= \frac{bh^3}{12} - \frac{(b-2t)(h-2t)}{12} = 9369\\
f_3(x) &= \frac{2t(h-t)^2(b-t)^2}{h+b-2t} = 6835,
\end{aligned} \tag{35}$$

where b is the section width, h is the section height and t is the section thickness. The nonlinear system has multiple solutions found by several researchers. Luo et al. [108] found six different non-dominated solutions with hybrid optimization of chaos optimization and the quasi-Newton method for this problem while Jaberipour et al. [42] found two different solutions with their proposed Particle Swarm Optimization method. Abdollahi et al. [41] and Mo et al. [28] attained optimum solutions of this problem utilizing Imperialist Competitive Algorithm and Conjugate Particle Swarm Optimization, respectively. Table 19 illustrates the optimum results obtained from these studies, our algorithm and other algorithms. Fig. 10 shows the convergence history of this problem. As seen in Fig. 10, L-QPSO has converged to optimum within 1503 iterations.

6. Conclusion

We offered a novel chaotic quantum behaved particle swarm optimization for solving nonlinear systems of equations. The system of nonlinear set of equations was transformed into an optimization problem and twelve chaotic maps have been

Table 18
Optimum results for case study 8.

	LQPSO	QPSO	ITHS	GRAV	Floudas et al. [98]	Abdollahi et al. [41]	Wang et al. [107]
x_1	0.16443166585432	0.164431665854	0.163641391023647	0.671708013238769	0.1644	0.164431665854327	0.67154465
x_2	−0.9863884768509	−0.98638847685	−0.98627002980711	0.740773292936966	−0.9864	−0.986388476850967	0.74097111
x_3	0.94762379640808	0.947623796408	0.95152343463466	−0.253002500191632	−0.9471	0.718452601027603	0.95189459
x_4	−0.3193886981111	−0.319388698111	−0.30706727299400	−0.967496424017105	−0.3210	−0.695575919707312	−0.30643725
x_5	−0.9984274739370	−0.998427473937	−0.9968607179168	0.957898268905606	−0.9982	0.997964383970433	0.96381470
x_6	−0.0560587128612	−0.056058712861	0.03838334619587	0.287126008079644	−0.0594	0.063773727557003	−0.26657405
x_7	−0.2575850995046	−0.257585099504	−0.25765415743645	−0.528417236794134	0.4110	−0.527809105283546	0.40463693
x_8	0.96625561654937	0.9662556165493	−0.9646484644758	−0.848969584145896	0.9116	−0.849363025083964	0.91447470
f_1	0.00000000000000	0.00000000000000	−0.01014171386634	−1.8519833783159E−4	−8.8531E−5	2.775557561562891E−016	−0.00000375
f_2	0.00000000000000	0.00000000000000	−0.01139457294335	−9.1791861881463E−5	3.5894E−5	−1.11022302462515E−016	0.00001537
f_3	2.6020852139E−18	1.7347234759E−18	0.016858410227952	8.0477908869916E−5	6.6216E−6	1.734723475976807E−018	0.00000899
f_4	0.00000000000000	1.6653345369E−16	0.00062893495184	−1.5795553262093E−4	2.1560E−5	1.665334536937735E−016	0.00001084
f_5	0.00000000000000	0.00000000000000	−0.00049292344812	−6.3273422149401E−5	1.2320E−5	0.00000000000000000000	0.00001039
f_6	0.00000000000000	0.00000000000000	−0.00031284319707	5.95955891025122E−5	3.9410E−5	0.00000000000000000000	0.00000709
f_7	0.00000000000000	0.00000000000000	−0.00479542780918	1.04380881087442E−5	−6.8400E−5	0.00000000000000000000	0.00000049
f_8	0.00000000000000	0.00000000000000	−0.00306767514011	−2.5586905399671E−5	−6.4440E−5	0.00000000000000000000	−0.00000498

Table 19
Optimum results for case study 9.

	b	h	t	$f_1(x)$	$f_2(x)$	$f_3(x)$
LQPSO	12.25651961037963	22.894938922662796	2.7898177367173895	165	9369	6835
QPSO	12.25667460849311	22.903035265500499	2.7849857955969321	165.1859	9369.0052	6834.9887
ITHS	8.915790281648581	23.291452604339380	12.885353223417773	165.8741	9366.4924	6831.8042
GRAV	12.26024390788173	22.775634683577675	2.857554308400358	167.5713	9362.2016	6836.7276
Abdollahi et al. [41]	8.943088778747601	23.271481879207862	12.912774291361677	165	9369	6835
Jaberipour et al. [42]	43.155566052654329	10.128950202278199	12.944048457756352	709.2412	9369	6835
	−7.6029951984634	−24.541982377674	−11.576715672202	208.1851	9369	6835
Mo et al. [28]	8.943089	23.271482	12.912774	165	9369	6835
Luo et al. [108]	12.5655	22.8949	2.7898	166.7229	9369	6835
	−12.5655	−22.8949	−2.7898	166.7229	9369	6835
	8.943089	23.271482	12.912774	165	9369	6835
	−8.943089	−23.271482	−12.912774	165	9369	6835
	−2.3637	35.7564	3.0151	165	9369	6835
	2.3637	−35.7564	−3.0151	165	9369	6835

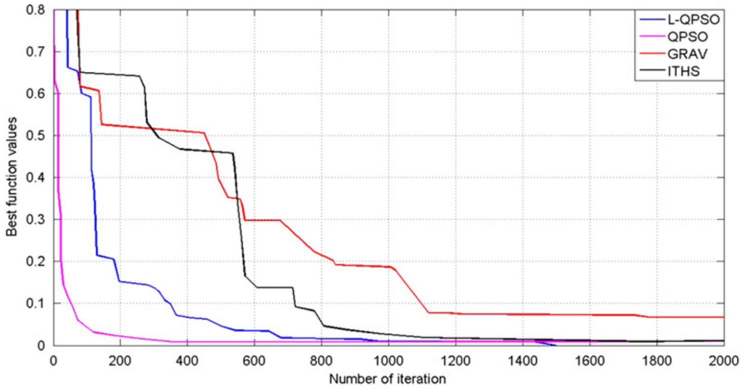

Fig. 8. Convergence history of the algorithms for case study 7.

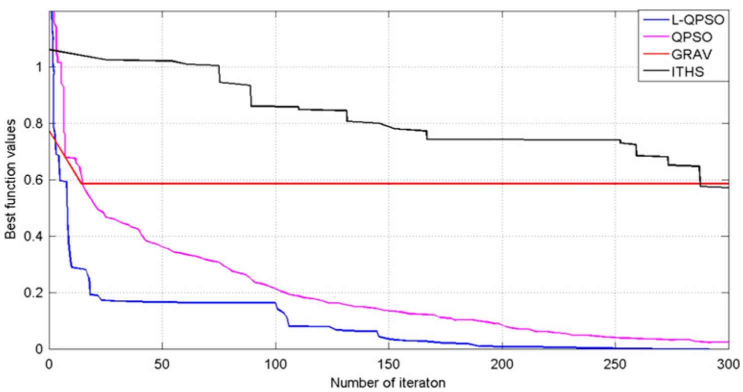

Fig. 9. Convergence history of the algorithms for case study 8.

investigated to improve the effectiveness of the algorithm. After numerical tests, the algorithm that used the Logistic map gave the best results and utilized in our algorithm. Proposed algorithm is easy to implement and has capability of solving all existing problems. By applying the Logistic map on our algorithm, we solved nine different nonlinear sets and compared our results with the Gravitational Search Algorithm, Intelligent Tuned Harmony Search algorithm and literature approaches. In all case studies, our solver found nearly all possible solutions with a minimum error and outperformed several literature studies and mentioned optimization methods. For a future work, we plan to apply this algorithm on reliability–redundancy problems.

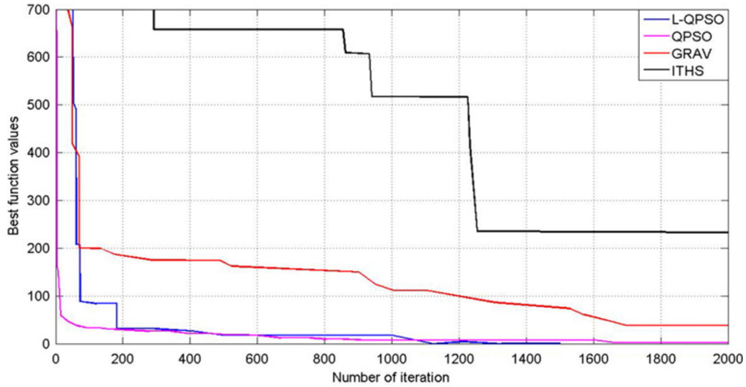

Fig. 10. Convergence history of the algorithms for case study 9.

References

[1] K. Argyros, On the solution of undetermined systems of nonlinear equations in Euclidean spaces, Pure Math. Appl. 4 (1993) 199–209.
[2] A. Holstad, Numerical solution of nonlinear equations in chemical speciation calculations, Comput. Geosci. 3 (1999) 229–257.
[3] K. Kowalski, K. Jankowski, Towards complete solutions to systems of nonlinear equations of many-electron theories, Phys. Rev. Lett. 81 (1999) 1195–1198.
[4] B.M. Barbashov, V.V. Nesterenko, A.M. Chervyakov, General solutions of nonlinear equations in the geometric theory of the relativistic string, Comm. Math. Phys. 84 (1982) 471–481.
[5] S.R. Bickham, S.A. Kiselev, A.J. Sievers, Stationary and moving intrinsic localized modes in one-dimensional monatomic lattices with cubic and quartic anharmonicity, Phys. Rev. B 47 (1993) 14206–14211.
[6] G. Yuan, X. Lu, A new backtracking inexact BFGS method for symmetric nonlinear equations, Comput. Math. Appl. 55 (2008) 116–129.
[7] B.W. Bader, Tensor–Krylov methods for solving large-scale systems of nonlinear equations, SIAM J. Numer. Anal. 43 (3) (2005) 1321–1347.
[8] R. Hammer, M. Hocks, M. Kulisch, D. Ratz, Numerical toolbox for verified computing, in: Basic Numerical Problems, Theory, Algorithms, and PASCAL-XSC Programs, Springer-Verlag, Heidelberg, 1993.
[9] E. Hansen, Global Optimization Using Interval Analysis, Marcel Dekker, New York, 1992.
[10] E.R. Hansen, R.I. Greenberg, An interval newton method, Appl. Math. Comput. 12 (1983) 89–98.
[11] E.R. Hansen, S. Sengupta, Bounding solutions of systems of equations using interval analysis, BIT 21 (1981) 203–211.
[12] H. Hong, V. Stahl, Safe starting regions by fixed points and tightening, Computing 53 (1994) 323–335.
[13] R.B. Kearfott, Pre conditioners for the interval Gauss–Seidel method, SIAM J. Numer. Anal. 27 (1990) 804–822.
[14] R.B. Kearfott, A review of preconditioners for the interval Gauss–Seidel method, Interval Comput. 1 (1991) 59–85.
[15] R. Krawczyk, Newton-algorithmen zur Bestimmung von Nullstellen mit Fehlerschranken, Computing 4 (1969) 187–201.
[16] R.E. Moore, Interval Analysis, Prentice-Hall, Englewood Cliffs, NJ, 1966.
[17] A. Neumaier, Interval Methods for Systems of Equations, in: Prentice-Hall Internat. Ser. Comput. Sci., Cambridge University Press, Cambridge, 1990.
[18] S.M. Rump, Verification methods for dense and sparse systems of equations, in: J. Herzberger (Ed.), Topics in Validated Computations, Elsevier, New York, 1988, pp. 217–231.
[19] P. Van Hentenryck, D. McAllester, D. Kapur, Solving polynomial systems using a branch and prune approach, SIAM J. Numer. Anal. 34 (2) (1997) 797–827.
[20] A.P. Morgan, Solving Polynomial Systems Using Continuation for Scientific and Engineering Problems, Prentice-Hall, Englewood Cliffs, NJ, 1987.
[21] J. Verschelde, P. Verlinden, R. Cools, Homotopies exploiting Newton polytopes for solving sparse polynomial systems, SIAM J. Numer. Anal. 31 (1994) 915–930.
[22] C.L. Karr, B. Weck, L.M. Freeman, Solutions to systems of nonlinear equations via a genetic algorithm, Eng. Appl. Artif. Intell. 11 (1998) 369–375.
[23] Q. Wang, J. Zeng, J. Zie, Modified particle swarm optimization for solving systems of equations, in: Third International Conference on Intelligent Computing ICIC 2007, Qingdao, China, pp. 661–667.
[24] A. Ouyang, Y. Zhou, Q. Luo, Hybrid particle swarm optimization algorithm for solving systems of nonlinear equations, in: IEEE International Conference on Granular Computing, 2009, GRC '09, 2009, pp. 460–465.
[25] R. Jia, D. He, Hybrid Artificial Bee Colony Algorithm for Solving Nonlinear System of Equations, in: 2012 Eighth International Conference on Computational Intelligence and Security, CIS, 2012, pp. 56–60.
[26] Y. Yang, Y. Zhou, Q. Gong, Hybrid artificial glow-worm swarm optimization algorithm for solving system of nonlinear equations, J. Comput. Inf. Syst. 6 (10) (2010) 3431–3438.
[27] M.J. Hirsch, P.M. Pardalos, M.G.C. Resende, Solving systems of nonlinear equations with continuous GRASP, Nonlinear Anal. RWA 10 (2009) 2000–2006.
[28] Y. Mo, H. Liu, Q. Wang, Conjugate direction particle swarm optimization solving systems of nonlinear equations, Comput. Math. Appl. 57 (2009) 1877–1882.
[29] F. Toutounian, J. Saberi-Nadjafi, S.H. Taheri, A hybrid of the Newton-GMRES and electromagnetic meta-heuristic methods for solving systems of nonlinear equations, J. Math. Model. Algorithms Oper. Res. 8 (2009) 425–443.
[30] W.F. Sacco, N. Henderson, Finding all solutions of nonlinear systems using a hybrid metaheuristic with fuzzy clustering means, Appl. Soft Comput. 11 (2011) 5424–5432.
[31] P.S. Mhetre, Genetic algorithm for linear and nonlinear equation, Internat. J. Adv. Eng. Technol. 3 (2) (2012) 114–118.
[32] H. Ren, L. Wu, W. Bi, K. Argyros, Solving nonlinear equations systems via an effective genetic algorithm with symmetric and harmonious individuals, Appl. Math. Comput. 219 (2013) 10967–10973.
[33] A. Pourrajabian, R. Ebrahimi, M. Mirzaei, M. Shams, Applying genetic algorithms for solving nonlinear algebraic equations, Appl. Math. Comput. 219 (2013) 11483–11494.
[34] I.M.M. El-Emary, M.M.A. El-Kareem, Towards using genetic algorithm for solving nonlinear equation systems, World Appl. Sci. J. 5 (3) (2008) 282–289.
[35] A. Rovira, M. Valdes, J. Casanova, A new methodology to solve non-linear equation system using genetic algorithms. application to combined cycle gas turbine simulation, Internat. J. Numer. Methods Engrg. 63 (2005) 1424–1435.
[36] C. Grosan, A. Abraham, A new approach for solving nonlinear equations systems, IEEE Trans. Syst. Man Cybern. A 38 (3) (2008) Senior Member, IEEE.
[37] C. Grosan, A. Abraham, A. Gelbukh, Evolutionary method for nonlinear systems of equations, in: Advances in Artificial Intelligence, MICAI 2006, in: Lecture Notes in Computer Science, vol. 4293, 2006, pp. 283–293.
[38] C. Zhaxi, Y. Li, A novel firefly algorithm of solving nonlinear equation group, Appl. Mech. Mater. 389 (2013) 918–923.

Please cite this article in press as: O.E. Turgut, et al., Chaotic quantum behaved particle swarm optimization algorithm for solving nonlinear system of equations, Computers and Mathematics with Applications (2014), http://dx.doi.org/10.1016/j.camwa.2014.06.013

[39] J. Zhang, Artificial bee colony algorithm for solving nonlinear equationand system, Comput. Eng. Appl. 48 (22) (2012) 45–47.
[40] E. Pourjarafi, H. Mojallahi, Solving nonlinear equations systems with a new approach based on invasive weed optimization algorithm and clustering, Swarm Evol. Comput. 4 (2012) 33–43.
[41] M. Abdollahi, A. Isazadeh, D. Abdollahi, Imperialist competitive algorithm for solving systems of nonlinear equations, Comput. Math. Appl. 65 (2013) 1894–1908.
[42] M. Jaberipour, E. Khorram, B. Karimi, Particle swarm algorithm for solving systems of nonlinear equations, Comput. Math. Appl. 62 (2011) 566–576.
[43] Z. Shuang, The application of particle swarm optimization to solving nonlinear equations, in: 2010 International Conference on Computational Aspects of Social Networks (CASoN), 26–28 Sept. 2010, pp. 733,736.
[44] B. Alatas, Chaotic bee colony algorithms for global numerical optimization, Expert Syst. Appl. 37 (2010) 5682–5687.
[45] A.H. Gandomi, X.S. Yang, Chaotic bat algorithm, J. Computat. Sci. 5 (2014) 224–232.
[46] B. Alatas, Chaotic harmony search algorithms, Appl. Math. Comput. 216 (2010) 2687–2699.
[47] G.G. Wang, L. Guo, A.H. Gandomi, G.S. Hao, H. Wang, Chaotic Krill Herd algorithm, Inform. Sci. 274 (2014) 17–34.
[48] A.H. Gandomi, X.S. Yang, S. Talathari, A.H. Alavi, Firefly algorithm with chaos, Commun. Nonlinear Sci. Numer. Simul. 18 (2013) 89–98.
[49] S. Talathari, B.F. Azar, R. Sheikholeslami, A.H. Gandomi, Imperialist competitive algorithm combined with chaos for global optimization, Commun. Nonlinear Sci. Numer. Simul. 17 (2012) 1312–1319.
[50] G. Gharooni-fard, F. Moein-darbari, H. Deldari, A. Morvaridi, Scheduling of scientific workflows using a chaos-genetic algorithm, Procedia Comput. Sci. 1 (2010) 1445–1454.
[51] J. Mingjun, T. Huanwen, Application of chaos in simulated annealing, Chaos Solitons Fractals 21 (2004) 933–941.
[52] L.d.S. Coelho, V.C. Mariana, Use of chaotic sequences in a biologically inspired algorithm for engineering design optimization, Expert Syst. Appl. 34 (2008) 1905–1913.
[53] W. Gong, S. Wang, Chaos ant colony optimization and application, in: 4th International Conference on Internet Computing for Science and Engineering, 2009, pp. 301–303.
[54] B. Alatas, Uniform Big Bang- Chaotic Big Crunch optimization, Commun. Nonlinear Sci. Numer. Simul. 16 (2011) 3696–3703.
[55] A.H. Gandomi, G.J. Yun, X.S. Yang, S. Talathari, Chaos-enhanced accelerated particle swarm optimization, Commun. Nonlinear Sci. Numer. Simul. 18 (2013) 327–340.
[56] J. Chuanwen, E. Bompard, A hybrid method of chaotic particle swarm optimization and linear interior for reactive power optimisation, Math. Comput. Simul. 68 (2005) 57–65.
[57] J. Chuanwen, E. Bompard, A self-adaptive chaotic particle swarm algorithm for short term hydroelectric system scheduling in deregulated environment, Energy Convers. Manage. 46 (2005) 2689–2696.
[58] T. Xiang, X. Liao, K. Wong, An improved particle swarm optimization algorithm combined with piecewise linear chaotic map, Appl. Math. Comput. 190 (2007) 1637–1645.
[59] H.J. Meng, P. Zheng, R.Y. Wu, X.J. Hao, Z. Xie, A hybrid particle swarm algorithm with embedded chaotic search, in: Conference on Cybernetics and Intelligent Systems, Signapore, 2004, pp. 367–371.
[60] B. Alatas, E. Akin, A.B. Ozer, Chaos embedded particle swarm optimization algorithms, Chaos Solitons Fractals 40 (2009) 1715–1734.
[61] J. Kennedy, R.C. Eberhart, Particle swarm optimization. in: Proceedings of the IEEE Conference on Neural Networks, Perth, Australia: 1995, pp. 1942–1948.
[62] X.S. Yang, Engineering Optimization: An Introduction with Metaheuristic Applications, John Wiley and Sons Inc., Hoboken, New Jersey, 2010.
[63] Y. Shi, R.C. Eberhart, A modified particle swarm optimizer, in: Proceedings of the IEEE International Conference on Evolutionary Computation, IEEE Press, Piscataway, NJ, 1998, pp. 69–73.
[64] J. Sun, B. Feng, W.B. Xu, Particle swarm optimization with particles having quantum behavior. in: IEEE Proceedings of Congress on, Evolutionary Computation, 2004, pp. 325–331.
[65] J. Sun, B. Feng, W.B. Xu, 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.
[66] W. Schweizer, Numerical Quantum Dynamics, Hingham, MA, USA, 2001.
[67] J. Liu, W. Xu, J. Sun, Quantum-behaved particle swarm optimization with mutation operator, in: Proceedings of 17th International Conference on Tools with Artificial Intelligence, Hong Kong, China, 2005.
[68] M. Clerc, J. Kennedy, The particle swarm: explosion, stability, and convergence in a multi-dimensional complex space, IEEE Trans. Evol. Comput. 6 (1) (2002) 58–73.
[69] Y. Cai, J. Sun, J. Wang, Y. Ding, N. Tian, X. Liao, et al., Optimizing the codon usage of synthetic gene with QPSO algorithm, J. Theoret. Biol. 254 (1) (2008) 123–127.
[70] M. Xi, J. Sun, W. Xu, An improved quantum-behaved particle swarm optimization algorithm with weighted mean best position, Appl. Math. Comput. 205 (2) (2008) 751–759.
[71] L.S. Coelho, Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems, Expert Syst. Appl. 37 (2010) 1676–1683.
[72] L.S. Coelho, A quantum particle swarm optimizer with chaotic mutation operator, Chaos Solitons Fractals 37 (5) (2008) 1409–1418.
[73] L.S. Coelho, V.C. Mariani, Particle swarm approach based on quantum mechanics and harmonic oscillator potential well for economic load dispatch with valve-point effects, Energy Convers. Manage. 49 (11) (2008) 3080–3085.
[74] B. Liu, L. Wang, Y.H. Jin, F. Tang, D.X. Huang, Directing orbits of chaotic systems by particle swarm optimization, Chaos Solitons Fractals 29 (2) (2006) 454–461.
[75] V.S.H. Rao, N. Yadaiah, Parameter identification of dynamical systems, Chaos Solitons Fractals 23 (4) (2005) 1137–1151.
[76] J.J. Yan, M.L. Hung, A novel stability criterion for interval time-delay chaotic systems via the evolutionary programming approach, Chaos Solitons Fractals 29 (5) (2006) 1079–1084.
[77] W.D. Chang, Parameter identification of Rosslers chaotic system by an evolutionary algorithm, Chaos Solitons Fractals 29 (5) (2006) 1047–1053.
[78] B. Liu, L. Wang, Y.H. Jin, F. Tang, D.X. Huang, Improved particle swam optimization combined with chaos, Chaos Solitons Fractals 25 (5) (2005) 1261–1271.
[79] T. Fujita, T. Watanabe, K. Yasuda, R. Yokoyama, Global optimization method using intermittency chaos, in: Proceedings of the 36th conference on decision and control San Diego, CA, USA, 1997, pp. 1508–1509.
[80] B. Li, W. Jiang, Optimizing complex functions by chaos search, Cybern. Syst. 29 (4) (1998) 409–419.
[81] R. Caponetto, L. Fortuna, S. Fazzino, M.G. Xibilia, Chaotic sequences to improve the performance of evolutionary algorithms, IEEE Trans. Evol. Comput. 7 (3) (2008) 289–304.
[82] M.S. Tavazoei, M. Haeri, Comparison of different one-dimensional maps as chaotic search pattern in chaos optimization algorithms, Appl. Math. Comput. 187 (10) (2007) 76–85.
[83] R.C. Hilborn, Chaos and Nonlinear Dynamics: An Introduction for Scientists and Engineers, second ed., Oxford Univ. Press, New York, 2004.
[84] D. He, C. He, L. Jiang, H. Zhu, G. Hu, Chaotic characteristic of a one-dimensional iterative map with infinite collapses, IEEE Trans. Circuits Syst. 48 (7) (2001) 900–906.
[85] A. Erramilli, R.P. Singh, P. Pruthi, Modeling Packet Traffic With Chaotic Maps, Royal Institute of Technology, Stockholm-Kista, Sweden, 1994.
[86] R.M. May, Simple mathematical models with very complicated dynamics, Nature 261 (4) (1976) 59–67.
[87] J.S. Arora, O.A. Elwakeil, A.I. Chahande, C.C. Hsieh, Global optimization methods for engineering application: a review, Struct. Optim. 9 (1995) 137–159.
[88] Y. Li, S. Deng, D. Xiao, A novel Hash algorithm construction based on chaotic neural network, Neural Comput. Appl. 20 (2011) 133–141.
[89] R.L. Devaney, An Introduction to Chaotic Dynamical Systems, Addison-Wesley, 1987.

Please cite this article in press as: O.E. Turgut, et al., Chaotic quantum behaved particle swarm optimization algorithm for solving nonlinear system of equations, Computers and Mathematics with Applications (2014), http://dx.doi.org/10.1016/j.camwa.2014.06.013

[90] H. Peitgen, H. Jurgens, D. Saupe, Chaos and Fractals, Springer-Verlag, Berlin, Germany, 1992.
[91] E. Ott, Chaos in Dynamical Systems, Cambridge University Press, UK, Cambridge, 2002.
[92] G.M. Zaslavskii, The simplest case of a strange attractor, Phys. Lett. A 69 (1978) 145–157.
[93] J. Sun, W. Fang, V. Palade, X. Wu, W. Xu, Quarntum behaved particle swarm optimization with Gaussian distributed local attractor point, Appl. Math. Comput. 218 (2011) 3763–3775.
[94] X. Yang, P. Shi, W. Shen, K. Jang, S. Pang, Multi-objective Quantum-behaved Particle Swarm Optimization with Entropy Based Density Assesment and Chaotic Mutation Operator, J. Computat. Inform. Syst. 9 (10) (2013) 3873–3881.
[95] J. Sun, W. Fang, X.J. Wu, V. Palade, W.B. Xu, Quantum-behaved particle swarm optimization: analysis of the individual particle's behaviour and parameter selection, Evolutionary Computation http://dx.doi.org/10.1162/EVCO.a.00049.
[96] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, GSA: a gravitational search algorithm, Inform. Sci. 179 (2009) 2232–2248.
[97] P. Yadav, R. Kumar, S.K. Panda, C.S. Chang, An intelligent tuned harmony search algorithm for optimisation, Inform. Sci. 196 (2012) 47–72.
[98] C.A. Floudas, P.M. Pardalos, C.S. Adjiman, W.R. Esposito, Z.H. Gumus, S.T. Harding, J.L. Klepeis, C.A. Meyer, C.A Schweiger, Handbook of Test Problems in Local and Global Optimization, Kluwer Academic Publishers, Dordrecht, Netherlands, 1999.
[99] C. Jäger, D. Ratz, A combined method for enclosing all solutions of nonlinear systems of polynomial equations, Reliab. Comput. (1995) 41–64.
[100] A. Morgan, V. Shapiro, Box-bisection for solving second-degree systems and the problem of clustering, ACM Trans. Math. Software 13 (1987) 152–167.
[101] S. Krzyworzcka, Extension of the Lanczos and CGS methods to systems of nonlinear equations, J. Comput. Appl. Math. 69 (1996) 181–190.
[102] M. Grau-Snchez, Grau, M. Noguera, Frozen divided difference scheme for solving systems of nonlinear equations, J. Comput. Appl. Math. 235 (2011) 1739–1743.
[103] J.R. Sharma, H. Arora, On efficient weighted-Newton methods for solving systems of nonlinear equations, Appl. Math. Comput. 222 (2013) 497–506.
[104] R.E. Moore, Methods and Applications of Interval Analysis, SIAM, Philadelphia, PA, 1979.
[105] L. Ingber, Adaptive simulated annealing (ASA): lessons learned, Control Cybernet. 25 (1) (1996) 33–54.
[106] H.A. Oliveira, A. Petraglia, Solving nonlinear systems of functional equations with fuzzy adaptive simulated annealing, Appl. Soft Comput. 13 (2013) 4349–4357.
[107] C. Wang, R. Luo, K. Wu, B. Han, A new filled function method for an unconstrained nonlinear equation, J. Comput. Appl. Math. 235 (2011) 1689–1699.
[108] Y.Z. Luo, G.J. Tang, L.N. Zhou, Hybrid approach for solving systems of nonlinear equations using chaos optimization and quasi-Newton method, Appl. Soft Comput. 8 (2008) 1068–1073.
[109] Y.Z. Luo, D.C. Yuan, G.J. Tang, Hybrid genetic algorithm for solving systems of nonlinear equations, Chin. J. Comput. Mech. 22 (1) (2005) 109–114.