Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 6284583, 9 pages
https://doi.org/10.1155/2020/6284583

Research Article

A Velocity-Combined Local Best Particle Swarm Optimization Algorithm for Nonlinear Equations

Zhigang Lian,[1] Songhua Wang ,[1,2] and Yangquan Chen[3]

[1]*Electrical Engineering Department, Shanghai DianJi University, Shanghai 201306, China*
[2]*School of Mathematics and Statistics, Baise University, Baise, Guangxi 533000, China*
[3]*School of Engineering, University of California, Merced, CA 95343, USA*

Correspondence should be addressed to Songhua Wang; wsh6600789@126.com

Received 19 June 2020; Accepted 4 August 2020; Published 25 August 2020

Academic Editor: Gonglin Yuan

Many people use traditional methods such as quasi-Newton method and Gauss–Newton-based BFGS to solve nonlinear equations. In this paper, we present an improved particle swarm optimization algorithm to solve nonlinear equations. The novel algorithm introduces the historical and local optimum information of particles to update a particle's velocity. Five sets of typical nonlinear equations are employed to test the quality and reliability of the novel algorithm search comparing with the PSO algorithm. Numerical results show that the proposed method is effective for the given test problems. The new algorithm can be used as a new tool to solve nonlinear equations, continuous function optimization, etc., and the combinatorial optimization problem. The global convergence of the given method is established.

1. Introduction

Many practical problems in engineering technology, information security, and so on can be solved by nonlinear equations [1, 2]. Because the complex of nonlinear equations, it is difficult to solve them, especially for high-dimensional nonlinear equations. Newton's method and its improved form are extensively used at present, but the Newton–Raphson method has limitations [3]. Their convergence and performance characteristics can be highly sensitive to the initial guess of the solution supplied to the methods. Also, it is difficult to select a good initial guess for most systems of nonlinear equations. Many researchers according to different nonlinear equations put forward various solutions. The Jacobian-free Newton–Krylova method is widely used in solving nonlinear equations arising in many applications; however, an effective precondition is required for each iteration, and determining such may be hard or expensive. Xu and Coleman [4] propose an efficient two-sided beclouding method to determine the lower triangular half of the sparse Jacobian matrix via automatic differentiation. With this lower triangular matrix, an effective preconditioner is constructed to accelerate the convergence of the Newton–Krylova method. Paper [5] introduces ANTIGONE, algorithms for continuous/integer global optimization of nonlinear equations, a general mixed-integer nonlinear global optimization framework. Yuan and Zhang [6, 7] presented a three-term Polak–Ribiere–Polyak conjugate gradient algorithm for large-scale nonlinear equations. Yan [8] introduced a new unified two-parameter wave model, connecting integrable local and nonlocal vector nonlinear Schrodinger equations. Fan and Lu [9] present a modified trust region algorithm for nonlinear equations with the trust region radii converging to zero. The trust region algorithm *s* preserves the global convergence as the traditional trust region algorithms. Moreover, it converges nearly q-cubically under the local error bound condition, which is weaker than the no singularity of the Jacobian at a solution. Paper [10] proposed and analyzed novel energy-preserving algorithms for solving the nonlinear Hamiltonian wave equation equipped with homogeneous Neumann boundary conditions. Paper [11] proposed a norm descent derivative-free algorithm for solving large-scale nonlinear symmetric equations without involving any information of the gradient or Jacobian matrix by using some approximate

substitutions. Yuan and Hu [12] proposed a new three-term conjugate gradient algorithm under the Yuan–Wei–Lu line search technique to solve large-scale unconstrained optimization problems and nonlinear equations. The numerical method is used to solve systems of equations in paper [13].

Evolutionary algorithms are also used to solve systems of equations, such as [14, 15]. This paper studies how to solve nonlinear equations using particle swarm optimization algorithm (PSO). The PSO algorithm was inspired by Shi and Eberhart [16] using social analogy of swarm behavior in populations of natural organisms, and it has been vastly developed. Jie Zhang et al. in paper [17] presented a hybrid clustering algorithm based on PSO using dynamic crossover. Chauhan et al. developed a novel inertia weight strategy for particle swarm optimization, in which he proposes three new nonlinear strategies for selecting inertia weight which plays a significant role in particle's foraging behavior [18]. Pan et al. [19] proposed an improved consensus protocol on the basis of the velocity and position equations of the canonical PSO algorithm, and transformed the dynamical PSO system into one new linear discrete-time system including random variables. The boundary of consensus region is given to better select the parameters in the PSO algorithm. The PSO has been successfully applied in many areas: function optimization, scheduling, fuzzy system control, and other areas. The core concept of PSO is changing the velocity and accelerating each particle toward its previous best position (pbest) and the global best position (gbest). Each particle modifies its current position and velocity according to the distance between its current position and pbest, and the distance between its current position and gbest. Supposing the search space is an N-dimension space, the swarm is made up of m particles. Each particle is represented as $x_i = (x_{i1}, x_{i2}, \ldots, x_{id})$. The velocity of the particle is denoted as $\text{vel}_i = (\text{vel}_{i1}, \text{vel}_{i2}, \ldots, \text{vel}_{id})$. The best previous position of the i-th particle is represented as $p_i = (p_{i1}, p_{i2}, \ldots, p_{id})$. The global best position of all particles is denoted as $p_{gi} = (p_{g1}, p_{g2}, \ldots, p_{gd})$. The velocity and the position are updated according to the following iterative equation:

$$\begin{aligned}\text{vel}_i(k+1) = {} & \text{vel}_i(k).w + c_1.r_1.\left(\text{pbest} - x_i(k)\right) \\ & + c_2.r_2.\left(\text{gbest} - x_i(k)\right),\end{aligned} \tag{1}$$

$$x_i(k+1) = x_i(k) + \text{vel}_i(k+1), \tag{2}$$

in which

(i) w is inertia weight

(ii) c_1 and c_2 are acceleration coefficient related to pbest and gbest

(iii) r_1 and r_2 are random number uniform distribution $U(0, 1)$

José García-Nieto and Enrique Alba used a parallel PSO for gene selection of high-dimensional microarray datasets [20]. Lin et al. [21] proposed a novel method for training the parameters of an adaptive network-based fuzzy inference system (ANFIS) that emphasizes the use of gradient descent (GD) methods.

Haddou and Maheu [22] developed some new smoothing techniques to solve general nonlinear complementarity problems. Litvinov et al. [23] carried out a survey on universal algorithms for solving the matrix Bellman equations over semirings and especially tropical and idempotent semirings. Chen et al. [24] developed a fast Fourier–Galerkin method for solving the nonlinear integral equation which is reformulated from a class of nonlinear boundary value problems. Wang and Zhang [25] presented a new family of two-step iterative methods to solve nonlinear equations. This paper presents the improved PSO algorithm, in order to optimize the complex nonlinear equations.

2. A Velocity-Combined Local Best Particle Swarm Optimization Algorithm for Equation

2.1. The Iterative Formula of VCLBPSO Algorithms. Although the PSO algorithm can converge very quickly towards the nearest optimal solution for many optimization problems, the PSO experiences difficulties in reaching the global optimal solution [26]. The fact is that the diversity of the swarm decreases when the swarm approaches the nearest optimal solution. Significant efforts have been devoted to improve the efficiency of the algorithm and enhance the diversity of the population. Gobbi et al. [27] researched on a kind of local approximation-based multiobjective optimization algorithm with applications.

In this article, we will work to increase the diversity of population information. In this new algorithm, when the current time run is completed, the best position of population particle will be used in the position update of next time run. Relatively, to the next round optimization, the current optimal particle is historical local best and differently from the next round global optimum. We call the improved PSO algorithm as the velocity-combined local best particle swarm optimization algorithm, VCLBPSO. The historical local information increases the diversity of reference information used by particles, which is expected to be useful to overcome the problem of premature convergence.

A swarm of m particles moves through a D-dimensional search space. A particle is defined by its current position $x_i = (x_{i1}, x_{i2}, \ldots, x_{id})$, $i = 1, 2, \ldots, m$, and velocity $v_{id}(k)$. Each particle remembers its better location information $p_i = (p_{i1}, p_{i2}, \ldots, p_{id})$, in which it has found the best solution of the optimization function f. Basing on the different combination between historical local best information and updated velocity, the VCLBPSO algorithm can be extended to four ways:

(1) The iterative formula of basic VCLBPSO is as follows:

$$\begin{aligned}v_{id}(k+1) = {} & w^* v_{id}(k) + c_1^* r_1^* \left(p_{id}(k) - x_{id}(k)\right) \\ & + c_2^* r_2^* \left(p_{gd}(k) - x_{id}(k)\right),\end{aligned} \tag{3}$$

$$v'_{id}(k+1) = \lambda^* v_{id}(k+1) + (1-\lambda)^* r_3^* \left(p_{ld}(t) - x_{id}(k)\right), \tag{4}$$

$$x_i(k+1) = x_i(k) + v'_{id}(k+1), \tag{5}$$

$$\overrightarrow{P}_L(t) = \overrightarrow{P}_L(t-1) \cup \overrightarrow{P}_L(t-2) \cup L \cup \overrightarrow{P}_L(1), \tag{6}$$

where equations (3) and (1) are exactly the same. w, c_1, c_2, r_1, r_2 and p_{id}, p_{gd} have the same meaning with that in equation (1). $p_{ld}(t)$ is the previous t-th time run searched local best solution for optimization problem, and $\overrightarrow{P}_L(t) = (p_{l1}(t), p_{l2}(t), \ldots, p_{lD}(t))$.$\lambda$is a constant coefficient that is used to balance the previous t-th time run searched local best and this run searched optimal information and to update velocity. Here, each particle updates its position mainly through a generation ago position and the velocity which updated with the previous t-th time run searched local best and this times run searched optimal information.

(2) The random combination way called velocity randomly combined local best particle swarm optimization algorithm, for short VRCLBPSO, and its iterative formula is as follows:

$$v_{id}(k+1) = w^* v_{id}(k) + c_1^* r_1^* \left(p_{id}(k) - x_{id}(k)\right) + c_2^* r_2^* \left(p_{gd}(k) - x_{id}(k)\right), \tag{7}$$

$$v'_{id}(k+1) = \lambda^* v_{id}(k+1) + (1-\lambda)^* r_3^* \cdot \left(p_{ld}(t) - x_{id}(k)\right), \tag{8}$$

$$x_i(k+1) = x_i(k) + v_{id}'(k+1), \tag{9}$$

$$\overrightarrow{P}_L(t) = \overrightarrow{P}_L(t-1) \cup \overrightarrow{P}_L(t-2) \cup L \cup \overrightarrow{P}_L(1), \tag{10}$$

where, w, c_1, c_2, r_1, r_2 and p_{id}, p_{gd}, $p_{ld}(t)$ have the same meaning with that in equation (3). In equation (7), $r_3 \in (0,1)$ is a random number.

(3) Entirely, velocity randomly combined the local best particle, for short EVRCLBPSO. Its iterative formula is as follows:

$$v_{id}(k+1) = w^* v_{id}(k) + c_1^* r_1^* \left(p_{id}(k) - x_{id}(k)\right) + c_2^* r_2^* \left(p_{gd}(k) - x_{id}(k)\right), \tag{11}$$

$$v'_{id}(k+1) = \lambda^* v_{id}(k+1) + (1-\lambda)^* r_3.^* \left(p_{ld}(t) - x_{id}(k)\right), \tag{12}$$

$$x_i(k+1) = x_i(k) + v'_{id}(k+1), \tag{13}$$

$$\overrightarrow{P}_L(t) = \overrightarrow{P}_L(t-1) \cup \overrightarrow{P}_L(t-2) \cup L \cup \overrightarrow{P}_L(1), \tag{14}$$

where, w, c_1, c_2, r_1, r_2 and p_{id}, p_{gd}, p_{ld} have the same meaning with that in equation (3). And $r_3 \in (0,1)$is an $m \times D$ matrix. If $r_3 \in (0,1)$ is $1 \times n$ dimension vector, and this algorithm for short NVRCLBPSO and its iterative formula is similarly to EVRCLBPSO.

2.2. The Method of Nonlinear Equations Converting to Function Optimization. To easily solve nonlinear equations, the mathematical expression of nonlinear equations transforming to function optimization is as follows.

Supposing a system of nonlinear equations is made up of n functions, involving n unknown variables, described as follows:

$$\begin{cases} f_1(x_1, x_2, \ldots, x_m) = 0, \\ f_2(x_1, x_2, \ldots, x_m) = 0, \\ \cdots, \\ f_n(x_1, x_2, \ldots, x_m) = 0. \end{cases} \tag{15}$$

where $f_i(X)$, $i = 1, 2, \ldots, m$ is a system of nonlinear functions, $X = (x_1, x_2, \ldots, x_n)^T$ is an unknown vector to it. In accordance with the algorithm principle, construct a fitness function is as follows:

$$\text{Min } F(X) = \sum_{i=1}^{n} \left| f_i \right|. \tag{16}$$

The problem for solving the system of nonlinear equations (15) is transformed into the optimization problem of minimum values of fitness function (16). According to mathematical principles, when $F(X)$ obtains the optimal value 0, $f_i(x_i)$ is also 0, and nonlinear equations are solved. When $F(X)$ is optimized by the improved PSO algorithm and obtaining the optimal value, $f_i(x_i)$ also is nearly to solve.

2.3. The Pseudocode of VCLBPSO Algorithm for Nonlinear Equations

(i) FUCTION VCLBPSOE ()
(ii) **FOR** optimization time from 1 to t (t usually is equation to (12))
(iii) Randomly engender original population and velocity
(iv) Compute the fitness of all particles using (16)
(v) Record the best fitness of all particles
(vi) **WHILE** the current number of iterations is less than the end iteration
(vii) Update the particle velocity using (3) of VCLBPSO algorithm
(viii) Modify the value of velocity using the update formula of VCLBPSO algorithm (4) or VRCLBPSO algorithm (8) and NVRCLBPSO algorithm (12)
(ix) Update the particle position using (5)
(x) Through comparing, get the optimal fitness of two generations, and record the better one as individual historical best

(xi) Record the best particle as this run global best in population

(xii) END

(xiii) Using (6), update the historical local best $\overrightarrow{P}_L(t)$. And if run time is the first, command the VCLBPSO algorithm is the same as basic PSO (1)-(2)

(xiv) END

(xv) Output the position of particle, and obtain the optimal fitness value

3. Experimental Approach

The application research of PSO in nonlinear equations has been investigated by some scholars, but it is limited. R. Brits et al. introduced the concept of shrinking particle neighborhood used in PSO to optimize systems of unconstrained equations [28]. Ouyang et al. proposed the hybrid particle swarm optimization algorithm to solve systems of nonlinear equations [29]. Mo et al. introduced the conjugate direction method into particle swarm optimization in order to improve PSO to solve systems of nonlinear equations [30]. They proposed methods that are mainly used to optimize the low-dimensional equations. Y. Mo and H. Liu used conjugate direction particle swarm optimization solving systems of nonlinear equations. In this paper, the improved PSO algorithm will mainly be used to optimize the mid-dimensional and high-dimensional nonlinear equations, besides low-dimensional.

Five nonlinear equations are used to test the effect of the improved PSO algorithm.

Equations 1. The theory of optimal solution is $(x_1, x_2) = (10, 1)$, or $(-7.5, 1)$:

$$\begin{cases} 0.05x_1^2 - 0.05x_2^2 - 0.125x_1 + 0.1x_2 - 3.8 = 0, \\ 0.1x_1x_2 - 0.1x_1 - 0.125x_2 + 0.125 = 0, \\ -100 \le x_i \le 100. \end{cases} \tag{17}$$

Equations 2. The theory of optimal solution is $(x_1, x_2, x_3, x_4, x_5, x_6) = (-1, 1, -1, 1-1, 1)$:

$$\begin{cases} x_1 + 0.25x_2^2x_4x_6 + 0.75 = 0, \\ x_2 + 0.405e^{1+x_1x_2} - 1.405 = 0, \\ x_3 - 0.5x_4x_6 + 1.5 = 0, \\ x_4 - 0.605e^{1-x_3^2} - 0.395 = 0, \\ x_5 - 0.5x_2x_6 + 1.5 = 0, \\ x_6 - x_1x_5 = 0, \\ -5 \le x_i \le 5. \end{cases} \tag{18}$$

Equations 3. The theory of optimal solution is $(x_1, x_2, x_3) = (4, 3, 1)$:

$$\begin{cases} x_1^{x_2} - x_2^{x_1} - 5x_1x_2x_3 - 85 = 0, \\ x_1^3 - x_2^{x_3} - x_3^{x_2} - 60 = 0, \\ x_1^{x_3} + x_3^{x_1} - x_2 - 2 = 0, \\ -2 \le x_1 \le 5, -1 \le x_2 \le 4, -1 \le x_3 \le 2. \end{cases} \tag{19}$$

Equations 4. The discretized two-point boundary value problem is similar to the problem in [6]:

$$g(x) \triangleq Ax + \frac{1}{(n+1)^2}F(x) = 0, \tag{20}$$

when A is the $n \times n$ tridiagonal matrix given by

$$A = \begin{bmatrix} 8 & -1 & & & & \\ -1 & 8 & -1 & & & \\ & -1 & 8 & -1 & & \\ & \cdots & \cdots & \cdots & \cdots & \\ & & & \cdots & \cdots & -1 \\ & & & & -1 & 8 \end{bmatrix}. \tag{21}$$

And $F(x) = (F_1(x), F_2(x), \ldots, F_n(x))^T$ with $F_i(x) = \cos x_i - 1, \quad i = 1, 2, \ldots, n.$

Equations 5. Unconstrained optimization problem is as follows:

$$\min f(x), \quad x \in R^n. \tag{22}$$

With Engval function $f\colon R^n \longrightarrow R$ defined by

$$f(x) = \sum_{2}^{n}\left[\left(x_{i-1}^2 + x_i^2\right)^2 - 4^*x_{i-1} + 3\right], \qquad g(x) \triangleq \frac{1}{4}\nabla f(x), \tag{23}$$

where $g(x) = (g_1(x), g_2(x), \ldots, g_n(x))^T$ with

$$\begin{aligned} g_1(x) &= x_1\left(x_1^2 + x_2^2\right) - 1, \\ g_i(x) &= x_i\left(x_{i-1}^2 + 2^*x_i^2 + x_{i+1}^2\right) - 1, \\ g_n(x) &= x_n\left(x_{n-1}^2 + x_n^2\right), \\ i &= 2, 3, \ldots, n-1, \end{aligned} \tag{24}$$

4. Experimental Results and Discussion

To test the effect of the improved algorithm for nonlinear equations, the same parameters of different algorithms are set exactly the same. Each problem to be optimized will be run 32 times by different methods. In the following, we will compare the effectiveness of the improved algorithm and the ordinary PSO in the case of nonlinear equations with different dimensions.

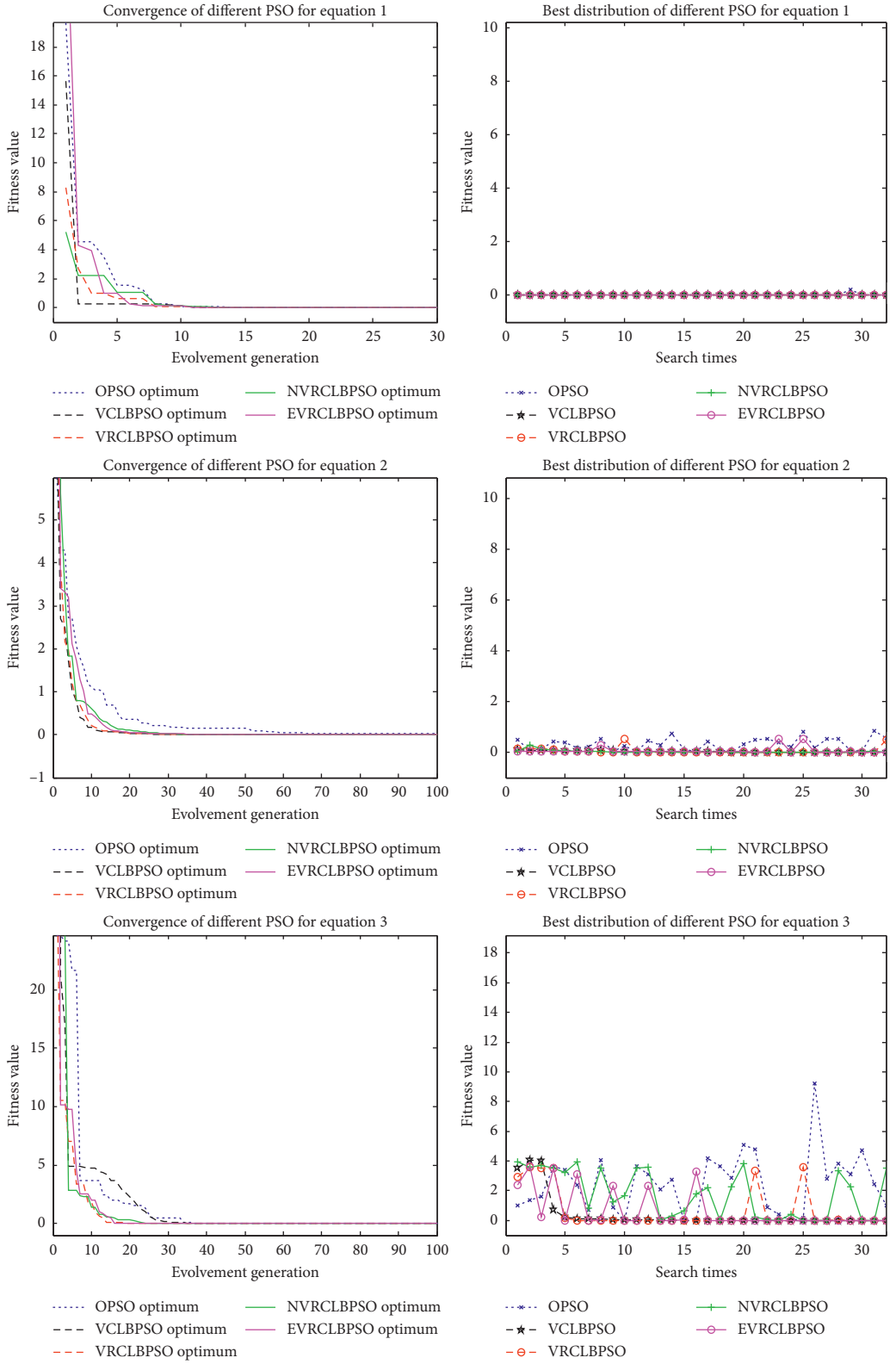
Table 1: Results of optimizing equ1–5 comparing VCLBPSO, VRCLBPSO, NVRCLBPSO, and EVRCLBPSO with PSO.

equ1, $Dim = 2$; $Ps = 50$; $Genn = 30$						
Result	Theory	PSO	VCLBPSO	VRCLBPSO	NVRCLBPSO	EVRCLBPSO
The best	0	$2.4942e-5$	**$3.5688\ e-7$**	$1.9969e-6$	$1.1267e-6$	$2.2815e-6$
Average	0	0.0074	$7.5187e-4$	$9.6079e-4$	**$2.2438\ e-4$**	0.001
Std	0	0.0406	0.0039	0.005	**$9.6179\ e-4$**	0.0055
Solution	(10, 1), (−7.5, 1)	(−7.5, 1)	(−7.5, 1)	(10, 1)	(10, 1)	(10, 1)
equ2, $Dim = 6$; $Ps = 50$; $Genn = 100$						
Result	Theory	PSO	VCLBPSO	VRCLBPSO	NVRCLBPSO	EVRCLBPSO
The best	0	0.0183	0.0048	**$6.31\ e-5$**	0.0015	0.0035
Average	0	0.3254	0.0337	0.0512	**0.0292**	0.0618
Std	0	0.2374	**0.0265**	0.1261	0.0548	0.1325
Solution	(−1, 1−1, 1−1, 1)	(−0.9586, 0.971, −1.0349, 0.9584, −1.0236, 0.9812)	(−0.9901, 0.9918, −1.0077, 0.9902, −1.006, 0.996)	(−1.0001, 1.0001, −0.9998, 1.0003, −0.9999, 1.0001)	(−9975, 0.9976, −1.0014, 0.9979, −1.0016, 0.9992)	(−9933, 0.9948, 1.0086, 0.9896, 1.0041, 0.9972)
equ3, $Dim = 3$; $Ps = 50$; $Genn = 100$						
Result	Theory	PSO	VCLBPSO	VRCLBPSO	NVRCLBPSO	EVRCLBPSO
The best	0	$6.2134e-5$	$3.1343e-7$	**$2.5109\ e-9$**	$2.9418e-4$	$7.5664e-7$
Average	0	2.4947	**0.4107**	0.6465	1.7852	0.6519
Std	0	1.9988	**1.137**	1.3551	1.5887	1.2548
Solution	(4, 31)	(4, 3, 1)	(4, 3, 1)	(4, 3, 1)	(4, 3, 1)	(4, 3, 1)
equ4, $Dim = 10$; $Ps = 50$; $Genn = 100$						
Result	Theory	PSO	VCLBPSO	VRCLBPSO	NVRCLBPSO	EVRCLBPSO
The best	0	$6.0214e-4$	*$9.7507e-9$*	$6.4342e-8$	$7.398e-7$	**$2.5128\ e-9$**
Average	0	168.6747	$3.9341e-4$	$5.9038e-4$	**$2.4065\ e-4$**	0.0012
Std	0	95.7371	0.0018	0.0023	**$8.4349\ e-4$**	0.006
equ4, $Dim = 100$; $Ps = 150$; $Genn = 6000$						
Result	Theory	PSO	VCLBPSO	VRCLBPSO	NVRCLBPSO	EVRCLBPSO
The best	0	158.0426	28.9963	2.2039	17.3192	**0.4519**
Average	0	$1.8912e+3$	298.0509	609.2726	144.6553	**131.3043**
Std	0	808.4996	93.358	86.7113	40.3754	**15.2786**
equ5, $Dim = 10$; $Ps = 50$; $Genn = 100$						
Result	Theory	PSO	VCLBPSO	VRCLBPSO	NVRCLBPSO	EVRCLBPSO
The best	0	0.0139	$3.8629e-5$	$1.1267e-6$	$4.3148e-6$	**$8.342\ e-8$**
Average	0	$1.4829e+3$	0.0079	0.0013	**0.0011**	0.0057
Std	0	$2.793e+3$	0.0277	**0.0039**	*0.004*	0.0271
equ5, $Dim = 150$; $Ps = 150$; $Genn = 6000$						
Result	Theory	PSO	VCLBPSO	VRCLBPSO	NVRCLBPSO	EVRCLBPSO
The best	0	34.801	1.0661	1.0943	0.8923	**0.2137**
Average	0	$5.3188e+3$	1.8718	1.6356	1.0518	**0.7122**
Std	0	$2.5912e+4$	0.62	0.3316	**0.1087**	0.5780

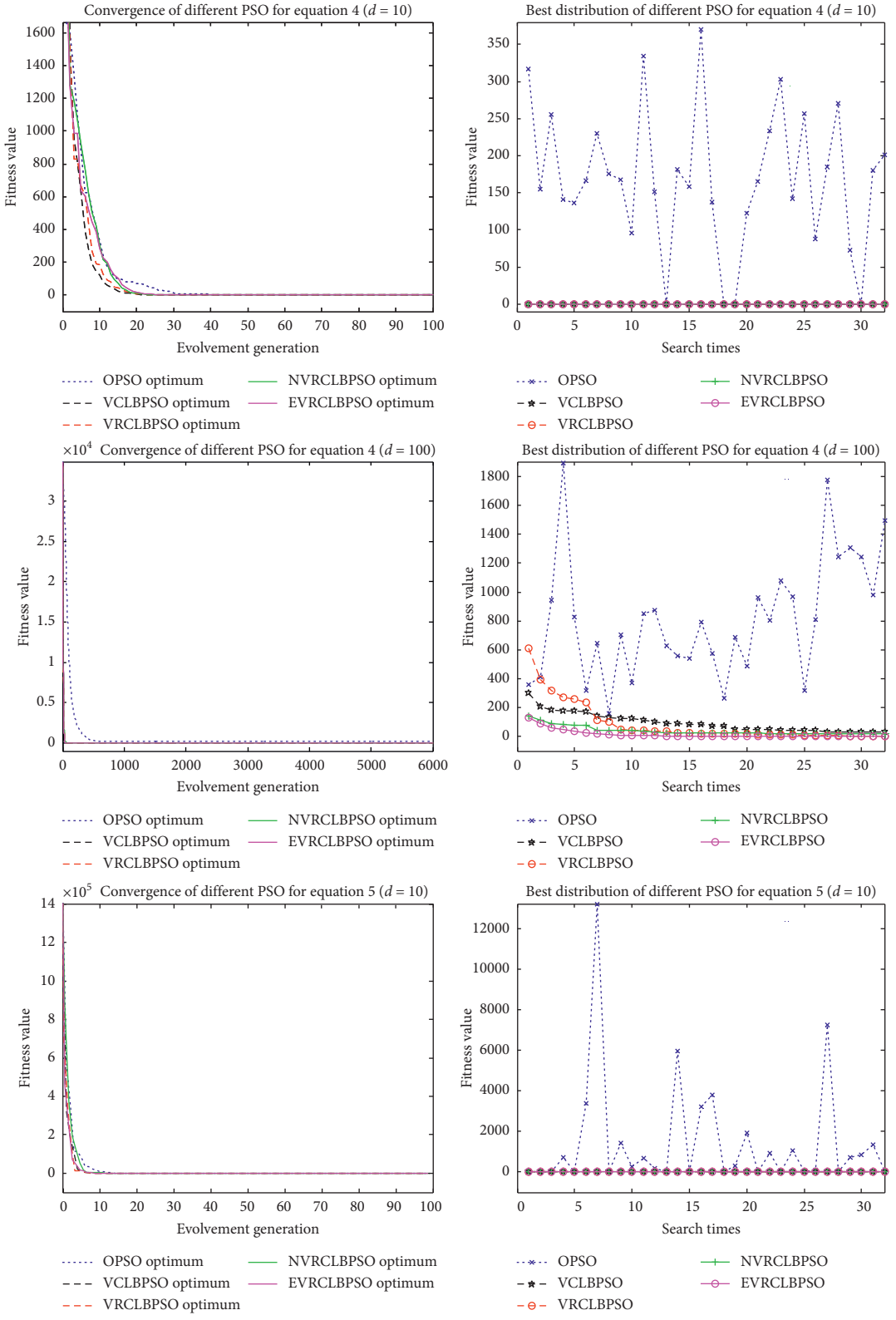
The experimental results are listed in Table 1. Relevant parameters are listed in the same table. In Table 1, *Dim* denotes problem's dimension; the *Ps* and *Genn* indicate population size and algorithm terminate generation, respectively.

Table 1 confirms that the improved algorithms obtained better results in the case of different dimensions. Table 1 shows that introduction history local best information can increase a particle's flight reference information diversity. This keeps the particles search to avoid falling into local optimal solution too early and the accuracy of convergence is improved. Improved PSO can obtain the small minimum, average, and standard deviation. In general, the VRCLBPSO, NVRCLBPSO, and NVRCLBPSO for low-dimensional problem are efficient, and the NVRCLBPSO search is more stable. The EVRCLBPSO is the best for high-dimensional problems. Their convergence and the best distribution of 32 run time figures are shown in Appendix A, in which the d is the dimension of problems.

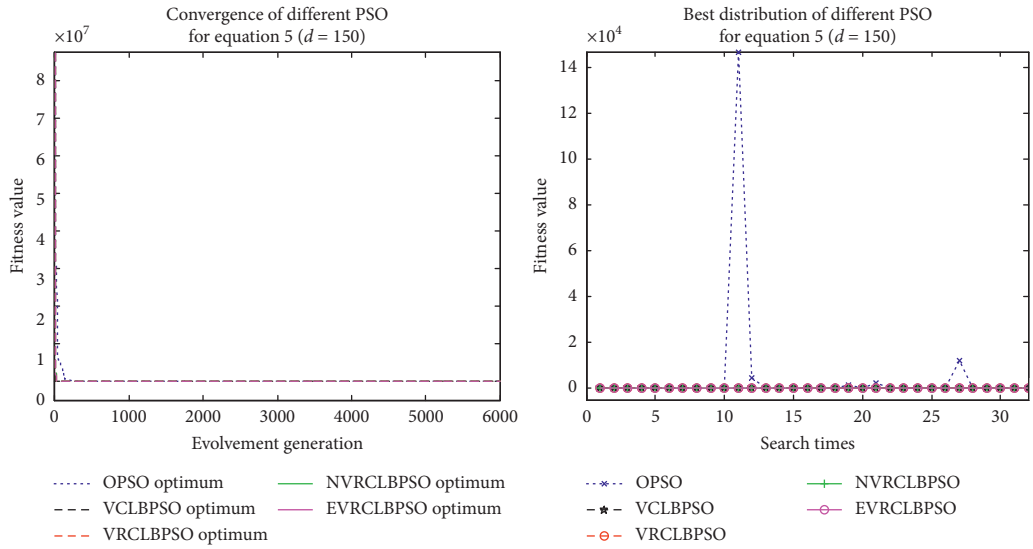
From Table 1 and Figure 1 of Appendix A, we discover that the convergence rate of the improved method for each problem is clearly faster than ordinary PSO. Also, for each problem with higher dimension, the EVRCLBPSO algorithm has the better convergence speed and more optimal search ability. When optimizing high-dimensional nonlinear equation problems, PSO easily falls into local optimum. While this paper presented a method successfully jumped out of the local optimal solution. Numerical results show that the proposed method is effective for some nonlinear equation problems. And the global convergence of the given method is established.


Convergence of different PSO for equation 1
Fitness value
Evolvement generation
Best distribution of different PSO for equation 1
Fitness value
Search times
OPSO optimum
VCLBPSO optimum
VRCLBPSO optimum
NVRCLBPSO optimum
EVRCLBPSO optimum
OPSO
VCLBPSO
VRCLBPSO
NVRCLBPSO
EVRCLBPSO
Convergence of different PSO for equation 2
Fitness value
Evolvement generation
Best distribution of different PSO for equation 2
Fitness value
Search times
OPSO optimum
VCLBPSO optimum
VRCLBPSO optimum
NVRCLBPSO optimum
EVRCLBPSO optimum
OPSO
VCLBPSO
VRCLBPSO
NVRCLBPSO
EVRCLBPSO
Convergence of different PSO for equation 3
Fitness value
Evolvement generation
Best distribution of different PSO for equation 3
Fitness value
Search times
OPSO optimum
VCLBPSO optimum
VRCLBPSO optimum
NVRCLBPSO optimum
EVRCLBPSO optimum
OPSO
VCLBPSO
VRCLBPSO
NVRCLBPSO
EVRCLBPSO

(a)


Convergence of different PSO for equation 4 ($d = 10$)
Fitness value
Evolvement generation
Best distribution of different PSO for equation 4 ($d = 10$)
Fitness value
Search times
OPSO optimum
VCLBPSO optimum
VRCLBPSO optimum
NVRCLBPSO optimum
EVRCLBPSO optimum
OPSO
VCLBPSO
VRCLBPSO
NVRCLBPSO
EVRCLBPSO
$\times 10^4$ Convergence of different PSO for equation 4 ($d = 100$)
Fitness value
Evolvement generation
Best distribution of different PSO for equation 4 ($d = 100$)
Fitness value
Search times
OPSO optimum
VCLBPSO optimum
VRCLBPSO optimum
NVRCLBPSO optimum
EVRCLBPSO optimum
OPSO
VCLBPSO
VRCLBPSO
NVRCLBPSO
EVRCLBPSO
$\times 10^5$ Convergence of different PSO for equation 5 ($d = 10$)
Fitness value
Evolvement generation
Best distribution of different PSO for equation 5 ($d = 10$)
Fitness value
Search times
OPSO optimum
VCLBPSO optimum
VRCLBPSO optimum
NVRCLBPSO optimum
EVRCLBPSO optimum
OPSO
VCLBPSO
VRCLBPSO
NVRCLBPSO
EVRCLBPSO

(b)



(c)

FIGURE. 1: The comparison figures of VCLBPSO, VRCLBPSO, NVRCBLPSO, EVRCLBPSO, and PSO algorithm for 5 nonlinear equations

5. Conclusions and Perspectives

So far, the research of PSO applied to optimization equations is very limited, especially for high-dimensional nonlinear equations. We propose an improved PSO algorithm for solving nonlinear equations in this paper. Experimental results comparing with the basic PSO algorithm show that the improved method has achieved better effect in both of convergence speed and accuracy. Particularly for high-dimensional nonlinear equations, the effect is very obvious. The application scenarios of this improved algorithm are relatively widespread, and it can be applied to solve unconstrained optimization problems, constrained optimization problems, equation problems, engineering practice problems, etc.

Nonlinear equations are an important question. Therefore, its optimization is very meaningful and valuable for many practical problems. It will be a valuable direction to apply the PSO algorithm and its variant to optimizing nonlinear equations, especially high-dimensional.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors thank Dr. Tomas Oppenheim and Dr. Zhuo Li, School of Engineering, University of California, for their comprehensive revision and some discussion of value. This work was supported by the Shanghai Natural Science Fund (14ZR1417300) and the Guangxi Natural Science Fund (2020GXNSFAA159069).

References

[1] H.-J. Peng, Q. Gao, H.-W. Zhang, Z.-G. Wu, and W.-X. Zhong, "Parametric variational solution of linear-quadratic optimal control problems with control inequality constraints," *Applied Mathematics and Mechanics*, vol. 35, no. 9, pp. 1079–1098, 2014.

[2] N. H. Tuan, L. D. Thang, D. D. Trong, and V. A. Khoa, "Approximation of mild solutions of the linear and nonlinear elliptic equations," *Inverse Problems in Science and Engineering*, vol. 23, no. 7, pp. 1237–1266, 2015.

[3] N. Bianchini, S. Fanelli, and M. Gori, "Optimal algorithms for well-conditioned nonlinear systems of equations," *IEEE Transactions on Computers*, vol. 50, no. 7, pp. 689–698, 2001.

[4] W. Xu and T. F. Coleman, "Solving nonlinear equations with the Newton-Krylov method based on automatic differentiation," *Optimization Methods and Software*, vol. 29, no. 1, pp. 88–101, 2014.

[5] R. Misener and C. A. Floudas, "ANTIGONE: algorithms for coNTinuous/integer global optimization of nonlinear equations," *Journal of Global Optimization*, vol. 59, no. 2-3, pp. 503–526, 2014.

[6] G. Yuan and M. Zhang, "A three-terms Polak-Ribière-Polyak conjugate gradient algorithm for large-scale nonlinear equations," *Journal of Computational and Applied Mathematics*, vol. 286, pp. 186–195, 2015.

[7] G. Yuan, T. Li, and W. Hu, "A conjugate gradient algorithm for large-scale nonlinear equations and image restoration problems," *Applied Numerical Mathematics*, vol. 147, pp. 129–141, 2020.

[8] Z. Yan, "Integrable PT-symmetric local and nonlocal vector nonlinear Schrödinger equations: a unified two-parameter model," *Applied Mathematics Letters*, vol. 47, pp. 61–68, 2015.

[9] J. Fan and N. Lu, "On the modified trust region algorithm for nonlinear equations," *Optimization Methods and Software*, vol. 30, no. 3, pp. 478–491, 2015.

[10] C. Liu, X. Wu, and W. Shi, "New energy-preserving algorithms for nonlinear Hamiltonian wave equation equipped with Neumann boundary conditions," *Applied Mathematics and Computation*, vol. 339, pp. 588–606, 2018.

[11] J. K. Liu and Y. M. Feng, "A norm descent derivative-free algorithm for solving large-scale nonlinear symmetric equations," *Journal of Computational and Applied Mathematics*, vol. 344, pp. 89–99, 2018.

[12] G. Yuan and W. Hu, "A conjugate gradient algorithm for large-scale unconstrained optimization problems and nonlinear equations," *Journal of Inequalities and Applications*, vol. 2018, no. 1, pp. 1–19, 2018.

[13] K. Anada, T. Ishiwata, and T. Ushijima, "A numerical method of estimating blow-up rates for nonlinear evolution equations by using rescaling algorithm," *Japan Journal of Industrial and Applied Mathematics*, vol. 35, no. 1, pp. 33–47, 2018.

[14] A. Ghodousian and A. Babalhavaeji, "An efficient genetic algorithm for solving nonlinear optimization problems defined with fuzzy relational equations and max-Lukasiewicz composition," *Applied Soft Computing*, vol. 69, pp. 475–492, 2018.

[15] A. Ullah, S. A. Malik, and K. S. Alimgeer, "Evolutionary algorithm based heuristic scheme for nonlinear heat transfer equations," *PLoS One*, vol. 13, no. 1, pp. 1–18, 2018.

[16] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization," in *Proceedings of the 1999 Congress on Evolutionary Computation (CEC 99)*, pp. 1945–1950, Washington, DC, USA, July 1999.

[17] J. Zhang, Y. Wang, and J. Feng, "A hybrid clustering algorithm based on PSO with dynamic crossover," *Soft Computing*, vol. 18, no. 5, pp. 961–979, 2014.

[18] P. Chauhan, K. Deep, and M. Pant, "Novel inertia weight strategies for particle swarm optimization," *Memetic Computing*, vol. 5, no. 3, pp. 229–251, 2013.

[19] F. Pan, Q. Zhang, J. Liu, W. Li, and Q. Gao, "Consensus analysis for a class of stochastic PSO algorithm," *Applied Soft Computing*, vol. 23, pp. 567–578, 2014.

[20] J. García-Nieto and E. Alba, "Parallel multi-swarm optimizer for gene selection in DNA microarrays," *Applied Intelligence*, vol. 37, no. 2, pp. 255–266, 2012.

[21] X. Lin, J. Sun, V. Palade, W. Fang, X. Wu, and W. Xu, "Training ANFIS parameters with a quantum-behaved particle swarm optimization algorithm," *Lecture Notes in Computer Science*, vol. 7331, pp. 148–155, 2012.

[22] M. Haddou and P. Maheux, "Smoothing methods for nonlinear complementarity problems," *Journal of Optimization Theory and Applications*, vol. 160, no. 3, pp. 711–729, 2014.

[23] G. L. Litvinov, A. Y. Rodionov, S. N. Sergeev, and A. N. Sobolevski, "Universal algorithms for solving the matrix Bellman equations over semirings," *Soft Computing*, vol. 17, no. 10, pp. 1767–1785, 2013.

[24] X. Chen, R. Wang, and Y. Xu, "Fast Fourier-Galerkin methods for nonlinear boundary integral equations," *Journal of Scientific Computing*, vol. 56, no. 3, pp. 494–514, 2013.

[25] X. Wang and T. Zhang, "A new family of Newton-type iterative methods with and without memory for solving nonlinear equations," *Calcolo*, vol. 51, no. 1, pp. 1–15, 2014.

[26] P. Chou, "High-dimension optimization problems using specified particle swarm optimization," *Lecture Notes in Computer Science*, vol. 7331, pp. 164–172, 2012.

[27] M. Gobbi, P. Guarneri, L. Scala, and L. Scotti, "A local approximation based multi-objective optimization algorithm with applications," *Optimization and Engineering*, vol. 15, no. 3, pp. 619–641, 2014.

[28] R. Brits, A. P. Englbrech, and F. Den Bergh, "Solving systems of unconstrained equations using particle swarm optimization," *IEEE International Conference on Systems, Man and Cybernetics*, vol. 3, 2002.

[29] A. Ouyang, Y. Zhou, and Q. Luo, "Hybrid particle swarm optimization algorithm for solving systems of nonlinear equations," in *Proceedings of the 2009 IEEE International Conference on Granular Computing*, pp. 17–19, Nanchang, China, August 2009.

[30] Y. Mo, H. Liu, and Q. Wang, "Conjugate direction particle swarm optimization solving systems of nonlinear equations," *Computers & Mathematics with Applications*, vol. 57, no. 11-12, pp. 1877–1882, 2009.